AI Research Labs

Machine Learning Foundations · Practice test 2 · 41 questions

Statistical Learning and Model Evaluation

How to use this test. Answer all 41 questions without notes, then mark yourself with the answer key at the back. Each question has exactly one best answer. Score 75% or more and you've mastered the material; below that, reread the lessons for the questions you missed.

2.1 Loss Functions and Risk

Q1 calculate A prediction is 8 and the true value is 11. What are the squared loss and the absolute loss?

A 9 and 3
B 6 and 3
C 9 and −3
D 3 and 9

Q2 calculate The true label is $y = 0$ and the model predicts $p = 0.9$ for class 1. What is the cross-entropy loss?

A $-\ln(0.1) \approx 2.303$
B $-\ln(0.9) \approx 0.105$
C $-\ln(0.5) \approx 0.693$
D 0.9

Q3 apply You're predicting house prices. A handful of mansions sell for 20 times the typical price, and you don't want them to dominate the fit. Which loss is the better choice?

A Absolute loss, because it grows only linearly with the size of an error
B Squared loss, because it handles large values better than absolute loss
C Cross-entropy, because it penalises confident mistakes the most heavily
D 0–1 loss, because it only counts whether each prediction is exactly right

Q4 interpret For a case with true label 1, model A predicts $p = 0.8$ and model B predicts $p = 0.2$. How do their cross-entropy losses compare?

A B's loss (1.609) is about 7 times A's (0.223)
B B's loss is 4 times A's, since $0.8/0.2 = 4$
C A's loss is larger, since it is more confident
D They're equal, because only misclassified cases count

Q5 concept A logistic regression is fitted by minimising cross-entropy, then judged by its expected cost at a chosen threshold. Is that inconsistent?

- A Yes: cost-based evaluation only makes sense for models that were trained with hinge loss
- B No: cross-entropy and expected cost are just two names for the same quantity
- C Yes: a model should always be evaluated with the same loss it was trained on
- D No: the loss must be easy to optimise, while the metric should reflect the real task

2.2 Empirical Risk Minimisation (ERM)

Q6 concept What does empirical risk minimisation (ERM) minimise?

- A The test-set error, compared across every candidate model in turn
- B The number of parameters needed to reach a target training accuracy
- C The expected loss over the whole population, within a chosen family of models
- D The average loss on the training sample, within a chosen family of models

Q7 concept Ordinary least squares is ERM with...

- A squared loss over the family of linear functions
- B absolute loss over the family of linear functions
- C cross-entropy loss over the family of linear functions
- D squared loss over the family of all possible functions

Q8 concept In $\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{R}_n(f)$, what is $\hat{f}$?

- A The function in $\mathcal{F}$ with the fewest parameters
- B The average of all the functions in the family $\mathcal{F}$
- C The lowest value of empirical risk reached by any function in $\mathcal{F}$
- D The function in $\mathcal{F}$ that achieves the lowest empirical risk

Q9 apply With $n = 10$ training points, you run ERM with squared loss over all polynomials of degree up to 9. What do you get?

- A A flat line at the mean of y, because the extra terms cancel each other out
- B The model with the lowest possible population risk among all the polynomials up to degree 9
- C A straight line, because ERM always prefers the simplest member of the family
- D A curve through every training point, with zero training loss but likely poor predictions

Q10 concept How does ridge regression fit into the ERM picture?

- A ERM with cross-entropy loss, applied to a continuous target
- B ERM applied to a smaller random subset of the training data, chosen afresh each time
- C ERM with absolute loss instead of squared loss, and no penalty
- D ERM with squared loss plus a penalty on the size of the coefficients

2.3 Underfitting and Overfitting

Q11 `interpret` Training error is 2% and validation error is 28%. What's the most likely diagnosis?

- A A good fit: the training error is impressively low
- B Overfitting: it fits quirks of the training data
- C Leakage: validation data has reached the model
- D Underfitting: the validation error is too high

Q12 `interpret` Training error is 35% and validation error is 36%. What's the diagnosis, and what might help?

- A Leakage; re-split the data and train again
- B Overfitting; add regularisation to shrink the gap between them
- C A good fit, because the two errors are so close
- D Underfitting; try a more flexible model or better features

Q13 `concept` Why can't training error be used to choose between a degree-1 and a degree-10 polynomial?

- A Training error is too noisy to compare across models
- B A degree-10 polynomial always has a higher training error
- C Training error can't be computed for polynomial models
- D Training error always favours the more flexible model

Q14 `diagnose` A model overfits. Which of these changes would make it WORSE?

- A Dropping weak, noisy features from the model
- B Collecting more training data
- C Adding a regularisation penalty
- D Adding higher-order polynomial terms

Q15 `interpret` As model complexity increases from very simple to very flexible, what typically happens?

- A Training error keeps falling; test error falls, then rises
- B Test error keeps falling; training error falls, then rises
- C Training error rises; test error keeps falling
- D Training and test error both keep falling together

2.4 Generalisation and the Generalisation Gap

Q16 `concept` Which change tends to shrink the generalisation gap?

- A Removing the regularisation penalty
- B Giving the model more free parameters
- C Collecting more representative training data
- D Adding random noise to the training labels

Q17 diagnose Validation accuracy is 96%, but the final test accuracy is 79%. What should the team do?

A Merge the test set into training and report the new training accuracy
B Investigate why, improve using train and validation data, then test on fresh data
C Report 96% as the expected accuracy, since this particular test set may just be unlucky
D Keep tuning the model until the score on this test set reaches 96%

Q18 interpret Training error is 30% and validation error is 31%. The gap is tiny. Is this a good model?

A No: a good model should have a negative gap, with lower validation error
B Yes: a small gap shows that the model generalises well
C Yes, as long as the training set was reasonably large
D Not necessarily: both errors are high, suggesting underfitting

Q19 interpret On a learning curve, as the training-set size grows, what typically happens?

A Training error falls while validation error rises, so the gap widens
B Both errors stay roughly constant once there are more than 100 training examples
C Training and validation error both fall steadily towards zero
D Training error rises a little, validation error falls, and the gap narrows

Q20 diagnose A model validated well at launch, but its accuracy drops steadily over the following months. What is the most likely cause?

A The model's training error was too low at launch time
B The regularisation strength λ was set too high
C The incoming data have drifted away from the training distribution
D The model was trained and tuned with too few cross-validation folds at launch

2.5 The Bias–Variance Trade-off

Q21 concept Which component of expected squared prediction error can no model reduce?

A The squared bias
B The training error
C The irreducible noise σ^2
D The variance of the fitted $\hat{f}$

Q22 concept As model complexity increases, what generally happens to bias and variance?

A Both bias and variance rise
B Bias falls and variance rises
C Bias rises and variance falls
D Both bias and variance fall

Q23 `interpret` A linear model is fitted to 50 different training sets drawn from the same source. The fitted lines are almost identical, but all miss the true curve the same way. How would you describe it?

- A High bias, low variance
- B Low bias, low variance
- C High bias, high variance
- D Low bias, high variance

Q24 `apply` You double the training data for a very flexible model. What do you expect?

- A Variance falls; bias and noise stay roughly the same
- B All three components fall by about half
- C Noise falls; bias and variance stay roughly the same
- D Bias falls; variance and noise stay roughly the same

Q25 `concept` In the bias–variance decomposition, what does "variance" refer to?

- A How spread out the residuals are on the training set after fitting
- B How spread out the target values y are in the data
- C How much the features vary between observations
- D How much the fitted prediction changes across training sets

2.6 Training, Validation and Test Sets

Q26 `apply` You need to choose the polynomial degree for a regression. Which data should you use to make that choice?

- A All of the data combined, to use every observation
- B The validation set, or cross-validation on the training data
- C The training set, since that's where the model is fitted
- D The test set, since it gives the most honest score

Q27 `calculate` You have 1,000 observations and use a 60:20:20 train/validation/test split. How many observations are in the test set?

- A 400
- B 200
- C 100
- D 600

Q28 `diagnose` A team tries 40 model configurations, checks each on the test set, and reports the best test score as its expected performance. What's wrong?

- A Nothing, because none of the models was trained on the test data
- B They tried too few configurations to be confident of finding a genuinely good model
- C Picking the best of 40 on the test set makes the reported score optimistic
- D The test set should be larger than the training set for this to work

Q29 concept Only 2% of transactions are fraudulent. Why use a stratified split?

A To remove duplicate transactions before splitting
B To make the training set as balanced as the test set
C To keep the fraud rate similar in every subset
D To put all fraud cases in the training set

Q30 apply You've chosen a model and its hyperparameters using the validation set. What is a common next step before reporting performance?

A Refit it on training + test, then report the validation score
B Refit the chosen model on training + validation, then test once
C Re-run the selection on the test set to confirm the choice
D Tune the hyperparameters again on the test set to finalise

2.7 Cross-validation

Q31 calculate The validation errors from 5-fold cross-validation are 4, 6, 5, 7 and 3. What is the CV estimate?

A 7
B 4.5
C 5
D 25

Q32 calculate You compare 3 candidate models with 10-fold cross-validation. How many model fits does that take, not counting the final refit?

A 30
B 13
C 10
D 3

Q33 diagnose Why is ordinary random K-fold CV inappropriate for forecasting next month's demand from monthly data?

A K-fold CV needs at least 1,000 observations to produce stable estimates
B Monthly demand is a count, and K-fold CV only works for continuous, unbounded targets
C Random folds let the model train on later months and validate on earlier ones
D Each month would end up in the validation fold more than once overall

Q34 interpret Model A has mean CV RMSE 2.22 with fold errors between 2.1 and 2.4. Model B has mean 2.20 with fold errors between 1.5 and 3.2. Which conclusion is most reasonable?

A B is clearly better, because it has the lower mean CV error
B They're similar on average, and A's stability may make it preferable
C The CV results are invalid, because the fold errors differ
D A is overfitting, because its fold errors are so close together

Q35 diagnose A team standardises features using the mean and SD of the whole training set, then runs 5-fold CV on it. Is there a leak?

A No: only feature selection, never scaling, can leak inside cross-validation
B Yes: held-out folds shaped the scaling; standardise inside each fold
C Yes: they should have standardised using the test set instead
D No: standardising doesn't learn anything from the data

2.8 Data Leakage and Good ML Practices

Q36 diagnose A team fills missing blood-test values with the median of the whole dataset, then splits it 80/20. Which leak is this?

A Target leakage
B Group (related-record) leakage
C Preprocessing before splitting
D Time leakage

Q37 diagnose To predict whether a customer will cancel next month, a model uses "number of calls to the cancellation hotline this month". Which leak is this?

A Test-set reuse for selection
B Target leakage
C Feature selection before CV
D Preprocessing before splitting

Q38 calculate Training statistics for house size are $\mu = 1{,}400$ and $\sigma = 400$ sq ft. What is the standardised value z of a 1,800 sq ft house in the test set?

A $+0.8$
B -0.4
C $+1.0$
D $+0.6$

Q39 diagnose A team standardises using training data only, picks the best of 12 models by 5-fold CV on the training set, and evaluates the winner once on the test set. Verdict?

A Leak: 12 models is too many to compare
B Leak: CV and testing used the same model
C No leak: this is the correct workflow
D Leak: preprocessing came before CV

Q40 diagnose A dermatology dataset has three photos per patient. The photos are split at random into training and test sets. What's the fix?

A Use more CV folds so that each photo is tested exactly once
B Split by patient, so each patient's photos stay together
C Remove two of each patient's photos before training
D Standardise each photo before splitting at random

Q41 diagnose From 20,000 candidate genes, a team keeps the 50 most correlated with the label using all rows, then reports 5-fold CV accuracy. What's wrong?

A. Validation labels helped choose the genes; select inside each fold

B. Five folds are too few; with ten folds there would be no problem

C. It only leaks if fewer than 50 genes are truly related to the label

D. Nothing: CV protects against this because each row is held out once

Answer key

Each answer comes with the reasoning, including why the most tempting wrong option is wrong.

Q1 **A** · lesson 2.1

The error is $11 - 8 = 3$. Squared loss $3^2 = 9$; absolute loss $|3| = 3$. Losses are never negative, and "6" doubles the error instead of squaring it.

Q2 **A** · lesson 2.1

For $y = 0$ the loss is $-\ln(1 - p) = -\ln 0.1 = 2.303$: the model gave only 0.1 to what actually happened. 0.105 is what you'd get if the true label were 1.

Q3 **A** · lesson 2.1

Squared loss squares big errors, so a few mansions can drag the whole fit towards them. Absolute loss weights an error of 20 as 20, not 400. 0–1 and cross-entropy are for classification, not prices.

Q4 **A** · lesson 2.1

Loss is $-\ln p$ for the true class: $-\ln 0.8 = 0.223$, $-\ln 0.2 = 1.609$. The log makes the penalty grow sharply as the probability given to the truth falls; it isn't a simple ratio of probabilities. Every case contributes, not just misclassified ones.

Q5 **D** · lesson 2.1

Fitting needs a smooth, well-behaved loss; evaluation should measure what matters in practice (costs, AUC, sensitivity). They're different quantities, and using different ones for the two jobs is normal.

Q6 **D** · lesson 2.2

ERM can only use what it has: the training sample. It picks the member of the family $\mathcal{F}$ with the lowest average training loss. Population risk is the goal but can't be computed, and the test set must not be used for fitting.

Q7 **A** · lesson 2.2

OLS minimises the residual sum of squares among straight lines (or hyperplanes). Over *all* functions, squared loss could be driven to zero by interpolating the data, which is not what OLS does.

Q8 **D** · lesson 2.2

"arg min" returns the *argument* that minimises (the function itself), not the minimum value. $\min \hat{R}_n$ would be the number; $\arg\min$ is the model.

Q9 **D** · lesson 2.2

A degree-9 polynomial has 10 coefficients, enough to pass exactly through 10 points. ERM happily picks it: training loss is zero. It nearly always overfits, which is why ERM must be paired with held-out evaluation or regularisation.

Q10 **D** · lesson 2.2

Ridge adds $\lambda \sum \beta_j^2$ to the RSS, so it's regularised ERM: still minimising training loss, but with a price on complexity. Absolute loss without a penalty would be least absolute deviations, a different method.

Q11 **B** · lesson 2.3

A big gap between very low training error and much higher validation error is the overfitting signature. Underfitting would show high error on *both*. Leakage tends to make validation look too *good*, not bad.

Q12 **D** · lesson 2.3

Both errors are high and similar, so the model can't capture the pattern even on data it was fitted to. Regularisation would make it simpler still. A small gap is only good news when both errors are low.

Q13 **D** · lesson 2.3

A degree-10 polynomial can always fit the training data at least as well as a straight line (it includes the line as a special case), so training error systematically rewards complexity. Compare on validation data or by cross-validation.

Q14 **D** · lesson 2.3

Overfitting means too much flexibility for the data available. More polynomial terms add flexibility. The other three reduce variance: more data, a penalty on complexity, or fewer features.

Q15 **A** · lesson 2.3

This is the U-shaped curve. Training error only goes down as flexibility grows. Error on new data improves while the model captures real structure, then worsens as it starts fitting noise. The bottom of the U is the sweet spot.

Q16 **C** · lesson 2.4

With more representative data, training performance reflects real performance more closely and the model has less room to memorise. More parameters and less regularisation widen the gap; noisier labels give more noise to fit.

Q17 **B** · lesson 2.4

Possible causes include overfitting to the validation set, distribution shift or a preprocessing mismatch. Tuning against the same test set would leak it, and reporting 96% ignores the evidence. A fresh, untouched test set gives the honest final number.

Q18 **D** · lesson 2.4

A small gap only says training performance is representative of new data. Here both are poor, so the model is too simple. Good generalisation means low error on new data, not just a small gap.

Q19 **D** · lesson 2.4

With more data it's harder to memorise everything (training error creeps up) and the fit captures real structure better (validation error falls). Neither reaches zero: noise sets a floor.

Q20 **C** · lesson 2.4

Models assume future data look like training data. When customer behaviour, prices or sensors change, that assumption fails gradually: distribution shift. Monitoring and retraining on recent data address it; CV settings at launch don't.

Q21 **C** · lesson 2.5

Noise is randomness in Y that the predictors can't explain. Bias falls with more flexible models and variance with more data or regularisation, but σ^2 is a floor under every model.

Q22 **B** · lesson 2.5

Flexible models can follow the true shape more closely (less bias) but react more to the particular sample they were trained on (more variance). That tension is the trade-off.

Q23 **A** · lesson 2.5

Stable across samples means low variance; consistently wrong in the same direction means high bias. It's like darts tightly grouped but away from the bullseye.

Q24 **A** · lesson 2.5

More data makes the fitted function change less between samples, which reduces variance. It doesn't change what the model family can represent (bias) or the randomness in Y (noise).

Q25 **D** · lesson 2.5

It is the variance of $\hat{f}(x_0)$ over repeated training samples, a property of the fitting procedure. The spread of the data itself is a different quantity, and confusing the two is a common slip.

Q26 **B** · lesson 2.6

Degree is a hyperparameter, so choosing it is a model-selection decision. Training error always prefers higher degrees, and using the test set would make the final score optimistic.

Q27 **B** · lesson 2.6

20% of 1,000 is 200. Training gets 600 and validation 200; 400 would be validation and test combined.

Q28 **C** · lesson 2.6

No model saw test rows, but the *choice* did. Among 40 tries, the winner partly won by luck on those particular rows, so its score overstates what to expect on new data. Select on validation, then test once.

Q29 **C** · lesson 2.6

With a rare class, a random split can leave a validation or test set with very few positives, making its scores unreliable. Stratifying preserves the class proportions in each subset; it doesn't rebalance them.

Q30 **B** · lesson 2.6

Once all decisions are made, using more data for the final fit usually helps, and the untouched test set gives one honest estimate. Any selection or tuning on the test set turns it into validation data.

Q31 **C** · lesson 2.7

The CV estimate is the mean of the fold errors: $25/5 = 5$. 25 is the sum; 7 is the worst fold.

Q32 **A** · lesson 2.7

Each candidate is fitted once per fold: $3 \times 10 = 30$. That cost is the price of a more stable comparison than a single split.

Q33 **C** · lesson 2.7

That's time leakage: the model learns from the future it's meant to predict. Use forward-chaining folds that always validate on the period after the training window. Every observation still appears in exactly one validation fold in ordinary K-fold.

Q34 **B** · lesson 2.7

A 0.02 difference in means is small next to B's huge spread, which signals instability (high variance). Consistent fold errors are a good sign, not overfitting. Look at the spread as well as the mean.

Q35 **B** · lesson 2.7

The mean and SD are statistics learned from data, including the rows that later act as validation folds. Each round should compute them from its own training folds. Using the test set would be a worse leak.

Q36 **C** · lesson 2.8

The median was computed using the test rows, so test information entered the pipeline. Split first, compute the median on training data only, and use that value for the test set too.

Q37 **B** · lesson 2.8

Calling the cancellation line is a consequence of already deciding to cancel. It makes the model look superb in testing but arrives too late to be useful. Use only information available when the prediction is made.

Q38 **C** · lesson 2.8

$z = (1800 - 1400)/400 = +1.0$. Always use the training statistics for test data. -0.4 is what you'd get by wrongly using test-set statistics ($\mu = 2000$, $\sigma = 500$); $+0.6$ uses the full-data statistics.

Q39 **C** · lesson 2.8

Scaling is learned from training data, selection uses only training data, and the test set is used once at the end. Comparing many models by CV is fine; the problem is only selecting on the test set. (Strictly, scaling should also be refitted inside each CV fold.)

Q40 **B** · lesson 2.8

Photos of the same patient share features, so the model can recognise the person rather than the condition, inflating the test score. A group split keeps every patient on one side. Deleting photos wastes data and isn't needed.

Q41 **A** · lesson 2.8

With 20,000 genes, many correlate with the label by chance. Choosing them using every row's label means the validation folds influenced the features, so CV accuracy is inflated. CV only protects steps that happen inside the folds.