START HERE

Machine Learning, Foundations

The classical ground under modern AI, for someone who can read code.

First edition, September 2026

Addaly

Series editor: Dr Mustafa Mun

Draft, open for academic review

START HERE

Machine Learning, Foundations

The classical ground under modern AI, for someone who can read code.

Series editor: Dr Mustafa Mun

Addaly

First edition, September 2026 · Draft, open for academic review

Machine Learning, Foundations

First edition, September 2026, build 62a26075.

Written by AI models to a written standard, with Dr Mustafa Mun as series editor. Exam keys checked blind by a second model: 3,665 of 3,666 agreed. Academic review invited.

The manuscript was drafted with the assistance of a large language model and was edited and published under his name. That is stated plainly here rather than left to be discovered, because a reader is entitled to know how a book they are trusting was made.

How to cite: *Mun, M. (2026). Machine Learning, Foundations. Addaly. addaly.com/learn/machine-learning-foundations.*

This book is the printed form of the course of the same name at addaly.com/learn/machine-learning-foundations. The website is the living version and is corrected first. Where the two differ, the website is right.

Free to read, free to download, free to print, and free to teach from. There is no paid edition and there never will be.

Every diagram in it is drawn from data by code, not generated as a picture, so the numbers on the page are the numbers in the argument. The answers to every check, test and examination question are at the back.

9 modules · 92 lessons · 30 figures · 9 case studies · 71,137 words · about 13 hours

Brief contents

1

MODULE 1

What learning is, and what it is not

Take a vague business question, state it as a dataset with a defined row, a defined target and a defined moment of prediction, name the loss and the baseline it must beat, and justify in one paragraph whether machine learning is the right tool at all.

Lesson 1 · 8 min

What a machine actually learns

THE ONE THING TO KEEP

Learning means adjusting the parameters of a shape you chose until a chosen error number is as small as it can be.

Fitting, not understanding

Here are four flats in Lagos, with monthly rent in naira.

Floor area (m²)	Rent (₦/month)
40	180,000
55	240,000
70	310,000
90	380,000

You want a program that guesses the rent of a flat it has never seen. You could write rules by hand. Instead you write down a shape with blanks in it:

```
rent = w * area + b
```

`w` and `b` are the blanks. They are called parameters. Every choice of `w` and `b` is a different program. Learning is the search for the pair that works best.

"Best" has to be a number

Guess `w = 4000`, `b = 0`. Run the four flats through it:

Area	Predicted	Actual	Off by
40	160,000	180,000	20,000
55	220,000	240,000	20,000
70	280,000	310,000	30,000
90	360,000	380,000	20,000

Average error: ₦22,500. That single number is called the **loss**. It is the only thing the machine can see about how it is doing.

Now try `w = 4000`, `b = 20000`:

Area	Predicted	Actual	Off by
40	180,000	180,000	0
55	240,000	240,000	0
70	300,000	310,000	10,000
90	380,000	380,000	0

Average error: ₦2,500.

The loss fell from 22,500 to 2,500. That drop is the entire content of the word "learning". No insight was gained. Two numbers moved and a third number got smaller.

The three ingredients

Every supervised model, from a two-parameter line to a language model with a trillion parameters, is made of the same three things.

1. **A shape with parameters.** A line. A tree of if-statements. A stack of matrix multiplications. You choose the shape.
2. **A loss.** One number saying how wrong the current parameters are, averaged over your data. You choose the loss, and the choice matters more than people expect.

3. **A procedure for changing the parameters to lower the loss.** For a straight line you can solve it with algebra. For anything larger you walk downhill, which is Lesson 8.

That is the whole mechanism. Everything else in this course is about doing it honestly.

What the model does not have

The fitted model has no concept of a building. Hand it `area = 500` and it will report ₦2,020,000 without hesitation, because it has never been told that flats of that size are rare, or that the market above 200 m^2 behaves differently. It has never been told anything. It has four rows and two knobs.

This also means the shape you picked is a hard ceiling. Suppose rents flatten out above 120 m^2, because very large flats sell rather than rent. A straight line cannot bend. No quantity of extra data will make `w * area + b` produce a curve — more data will only pin down the best possible straight line more precisely, and the best possible straight line is still wrong. If you want a curve, you have to pick a shape that can curve.

This is why "which model should I use" is a real question and not a detail. You are not asking which program is smarter. You are choosing which family of functions the search is allowed to look inside.

Parameters and hyperparameters

`w` and `b` are fitted from the data. But other numbers get chosen before fitting starts: how many terms the polynomial has, how deep a tree may go, how large each downhill step is. These are **hyperparameters**. They are picked by you, by trial, and where you are allowed to run those trials is the subject of Lesson 4.

Keep the distinction clear from the start. Parameters come from the data. Hyperparameters come from you.

BEFORE YOU MOVE ON

You fit `rent = w * area + b` to 50,000 flats. The true relationship curves: rent rises steeply up to 120 m^2 and then almost flattens. Training finishes. What has it accomplished?

A. It has found the best straight line through curved data, and the curve is permanently out of reach for this model.

B. With 50,000 rows it has enough data to discover the curve on its own.

C. Training will drive the loss to zero if it is allowed to run long enough.

D. The curve is a sign the data is noisy and should be cleaned before fitting.

The answer and why: p. 493

Lesson 2 · 7 min

With answers, and without

THE ONE THING TO KEEP

A supervised model learns your labels, not the reality behind them, so who made the labels is part of the model.

Two different things to have

A bank in Nairobi has ten million past transactions. For 4,000 of them, an investigator looked and wrote down `fraud` or `not fraud`. Those 4,000 rows have **labels**: recorded answers.

With labels, you can do **supervised learning**. You show the model an input and the answer, it adjusts, and eventually it maps new inputs to answers. Almost everything with commercial value is supervised: spam filters, credit scoring, delivery-time estimates, medical triage, machine translation.

Without labels you can do **unsupervised learning**. You give the model ten million transactions and no answers, and ask what structure is in there. Clustering groups similar rows. Dimensionality reduction compresses 200 columns into 2 you can plot. Anomaly detection finds rows unlike the rest.

The distinction is not about the algorithm. It is about what you have.

Unsupervised results are questions, not answers

Run k-means on those transactions asking for six clusters, and you will get six clusters. You will get six clusters if the data is pure random noise, too. The algorithm has no way to tell you that the structure it found is not there.

And the clusters arrive unnamed. Cluster 4 is "rows 88,201, 88,540, 91,003 and 402,884 others". Someone has to look at them and say "these are salary deposits on the 25th" or "these are top-ups from a single agent network". That naming is human work, and it is where the mistakes live.

So: unsupervised methods are good at generating hypotheses and terrible at confirming them. If a cluster looks like a fraud ring, that is a lead for an investigator, not a finding.

The label defines the task

Here is the part that gets skipped, and it is the most important idea in this lesson.

A supervised model does not learn the truth. It learns your label. If your label is imperfect, the model learns the imperfection with the same enthusiasm it learns everything else.

The bank's label is not "fraud". It is "transactions an investigator looked at and confirmed as fraud". Those are different things:

- Fraud nobody noticed is labelled `not fraud`.
- Fraud noticed only because it matched the old rule-based system is over-represented.
- Fraud in a customer segment nobody audits is invisible.

Train on that and you get a model that reproduces the investigators' existing reach. It may be genuinely useful — it can rank a queue and save time — but it is a model of the detection process, not of fraud.

The same trap in other clothes:

- A hospital labels "patients with condition X" using billing codes. The model learns which patients get coded, which depends on the clinic, the insurer and the country.
- A recruiter labels "good hire" as "still employed after two years". The model learns retention, which is partly about managers.
- A logistics firm labels "late delivery" using customer complaints. The model learns which customers complain.

Before you look at a single metric, write one sentence: *this label was created by _____, and it misses _____*. If you cannot fill in the blanks, you do not yet know what you are building.

The middle ground, and where modern AI sits

Two shapes sit between the extremes and are worth knowing by name.

Semi-supervised: a small labelled set plus a large unlabelled one. Common in medicine, where labelling means a specialist's time.

Self-supervised: labels manufactured from the data's own structure. Hide a word in a sentence and predict it. Hide a patch of an image and predict it. No human labelled anything, yet the task is supervised in mechanism, and there is an unlimited supply of it.

Self-supervision is how large language models are trained, and it is the main reason they exist at all. The bottleneck on supervised learning was always labels. Self-supervision removed the bottleneck by inventing a task where the answer is already sitting in the data.

BEFORE YOU MOVE ON

A payments team trains a fraud classifier on a label built from cases their investigators confirmed as fraud. On held-out data it performs well. What has the model most likely learned?

A. Which transactions resemble the ones this team's investigators already catch.

B. What fraud looks like in general, since every confirmed case genuinely was fraud.

C. Nothing worth deploying, because the labels are biased.

D. It would avoid this problem if they used unsupervised clustering instead.

The answer and why: p. 494

Lesson 3 · 9 min

What a row is, and why that decision is the whole project

THE ONE THING TO KEEP

Decide what one row is and when the target becomes knowable before you write any code, because every split, metric and deployment decision inherits that choice silently.

The two objects

Almost all of classical machine learning operates on two objects. A matrix `X` with one row per thing and one column per measurement, and a vector `y` with one entry per row: the answer you want the model to produce.

CASE STUDY · MODULE 1

Two definitions of a dropout, in Kota

An illustrative case, built from documented patterns.

A coaching centre in Kota with 2,300 students, two counsellors, and an analyst borrowed from the accounts department.

Vidya Path enrols 2,300 students across one-year and two-year programmes for the engineering entrance exam. Fees arrive in three instalments. A student who stops coming after the second instalment costs the centre the third, and costs the family a great deal more than that.

The management asked for a model that would "predict dropouts early enough to do something about it". They had five years of attendance registers, weekly test scores, fee receipts and hostel allocations, and they had Reshma, moved across from accounts, who had worked through a free machine learning course on her phone over four months.

Her first week produced no code. It produced one sentence, and an argument about it.

The row was easy. One student-month. A student appears once for every month they were enrolled.

The target was not. Two definitions were available, and they were not variations on a theme.

Definition A: a student is a dropout if they do not re-enrol for the following term. Unambiguous, already recorded in the fee system, and exactly what the director meant by the word. It also produces one label per student per year — about 2,300 rows a year and roughly 180 positives.

Definition B: a student is a dropout if they attend no class for twenty-one consecutive days. One label per student-month, which over five years is about 27,000 rows and roughly 1,900 positives. It is also wrong sometimes. A student with dengue, or one who went home for a cousin's wedding and came back, gets the same label as one who has left for good.

The director wanted A. "That is what a dropout is." Reshma wanted B, for a reason that had nothing to do with elegance. With 180 positives and forty candidate features, the events-per-feature arithmetic said nothing she fitted would be stable, and a validation set holding thirty-six positives could not tell a good model from a lucky one.

The cost fell on both sides and neither was small. Definition A would produce a model whose per-student predictions could not be trusted, wearing a validation score with an interval wide enough to cover both success and failure. Definition B would produce a model that fired on students who were about to walk back through the gate, and every false alarm was a counsellor telephoning a family that had done nothing wrong. In Kota, where the pressure on these students is a matter of public concern, an unnecessary call from the institute is not a neutral event.

The moment of prediction settled less than either of them expected, and it settled it the same way for both definitions. Features had to be everything known as of the last day of the month and nothing else. That excluded `refund_processed`, `hostel_vacated_date` and `final_instalment_status`, all three of which were in the export the office had handed over, and all three of which are written after the outcome they would have been used to predict. Reshma put the exclusion list at the top of the notebook before she opened a single CSV.

The baseline took twenty minutes and changed the conversation. Predicting "nobody drops out" scored ninety-three per cent under definition A. More usefully, the rule the counselling team already used — flag any student whose attendance over the last fortnight fell below sixty per cent — caught forty-one per cent of eventual dropouts and produced about nine flags a day, which two counsellors could just about work through.

Nine flags a day was the real constraint, and nobody had written it down before. The question stopped being "can we predict dropouts" and became "can we beat forty-one per cent recall at nine flags a day, without asking the counsellors to make calls they do not have time to make".

YOUR ANSWERS

1. Definition B gave ten times as many labelled events. Why did that not simply make the model better?

2. The refund column is strongly associated with dropping out. Why exclude it?

3. Why was 'nine flags a day' a better target than 'maximise AUC'?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 50.

CASE STUDY · MODULE 1

How it played out

They trained on definition B and evaluated on definition A. The noisy monthly label gave the optimiser enough events to learn from; the clean annual label, available for the two cohorts that had completed, was what they reported. At nine flags a day the model reached sixty-three per cent recall against the rule's forty-one.

The first twenty calls found three students on recorded medical leave. Rather than retrain, the centre added a suppression rule outside the model — never flag a student with an approved leave record — which is the rule-and-model hybrid from lesson seven of the module, arrived at by being embarrassed rather than by design.

The director's requirement, added late and kept since, was one line in the handover note: the model predicts a twenty-one-day absence, not a departure, and the counsellor is told which of those they are calling about.

The questions, discussed

1. Definition B gave ten times as many labelled events. Why did that not simply make the model better?

Because the extra events are noisier. A twenty-one-day absence includes students who return, so a share of the positive labels do not correspond to the outcome anyone cares about. The model learns the label it is given, so it learns 'who will disappear for three weeks', which overlaps with dropping out without being it. The gain in estimation stability was real and the cost in label fidelity was real; the resolution was to train on the plentiful label and measure against the clean one.

2. The refund column is strongly associated with dropping out. Why exclude it?

Because at the moment the prediction is made — the last day of a month, for a student still enrolled — the refund has not been processed. It exists in the training table only for students whose outcome is already known, so it is a consequence of the target rather than a predictor of it. Including it would raise

every validation score and produce a model that has nothing to say about a student who is still attending, which is the only kind of student the counsellors can act on.

3. Why was 'nine flags a day' a better target than 'maximise AUC'?

Because the counsellors are the constraint, and the decision the model supports is which nine students get a call today. AUC averages the model's behaviour over thresholds that will never be used. Precision and recall at the operating point of nine flags describe what will actually happen, they are comparable against the rule already in place, and they make the trade-off legible to a director who does not know what an ROC curve is.

MODULE 1 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 492. If two go wrong, re-read the module before the exam.

1 A delivery-time model is trained on data where most trips take 20 to 40 minutes and a handful take three hours because of a road closure. The team switches the training loss from squared error to absolute error and changes nothing else. What changes about the fitted model?

A. It fits closer to the median trip and the rare long trips pull it much less.

B. It becomes more accurate on the long trips, because every minute of error now counts equally.

C. Its predictions stop being biased, because absolute error is an unbiased estimator of the mean.

D. Nothing changes until the learning rate is also lowered to match the new scale of the loss.

2 A churn model reaches 0.99 AUC on its first run, with no tuning, on a problem the operations team finds genuinely difficult. What is the most likely explanation?

A. The features are unusually strong, and churn is simply easier to predict than the team believes.

B. The test set is too small, so the estimate is noisy and this run happened to come out high.

C. A column recorded after the outcome has reached the feature set.

D. The classes are imbalanced, and AUC is inflated whenever one class dominates the data.

3 A bank builds a default model where one row is one loan, and customers hold up to eleven loans each. The rows are split at random. What does the validation score actually measure?

A. How well the model predicts for customers it has never seen, because the individual loans differ.

B. Nothing at all, because a loan-level row cannot be used to predict a customer-level outcome.

C. The right thing, but pessimistically, because one customer's loans are correlated with each other.

D. How well the model predicts for customers whose other loans it trained on.

4 A telecom model finds that customers who call support are far more likely to churn. Management proposes making the support line harder to reach. What is wrong with that reasoning?

A. The coefficient is unstandardised, so its size cannot be compared with the other features.

B. The call and the churn share a cause, so removing the call removes the signal.

C. The model was trained on historical data and would need retraining before it is acted upon.

D. Support calls are rare, so the association is unlikely to be statistically significant.

5 A government office applies an eligibility rule written into a regulation: five conditions, each exactly specified. A team proposes training a classifier on past decisions instead. Why is that a worse design?

A. A classifier needs at least ten thousand examples and the office has fewer decisions than that.

B. The classifier would learn the regulation correctly and then drift as the input data changes.

C. The rule is a specification, not a pattern, so a model can only approximate it.

D. The model's probabilities would be uncalibrated, so the office could not state a confidence level.

6 A team evaluates on the test set, sees a disappointing number, changes two features, and evaluates again. What is the state of their test estimate now?

A. It is optimistic, because a decision was made using it.

B. It is unchanged and still valid, because they never trained on the test rows themselves.

C. It is pessimistic, because the second evaluation used a model tuned for a different feature set.

D. It is valid as long as they report the first number rather than the second one.

That is gradient descent, complete. The hill is the loss as a function of the parameters. Your position is the current parameter values. The slope is the gradient. The step size is the learning rate.

An actual walk

One parameter, and a loss with a known bottom:

```
L(w) = (w - 3)²
```

The slope at any `w` is `2(w - 3)`. Start at `w = 0` with learning rate 0.1. The rule is: new `w` = old `w` − learning rate × slope.

Step	w	Slope	New w
1	0.000	−6.00	0.600
2	0.600	−4.80	1.080
3	1.080	−3.84	1.464
4	1.464	−3.07	1.771
…			
20	2.965	−0.07	2.972

It closes in on 3, taking smaller steps as the slope flattens, because near the bottom there is less slope to push it. Nobody told it the answer was 3. It only ever felt the ground where it stood.

Now set the learning rate to 1.1 and start again:

Step	w	Slope	New w
1	0.00	−6.00	6.60
2	6.60	7.20	−1.32
3	−1.32	−8.64	8.18
4	8.18	10.37	−3.22

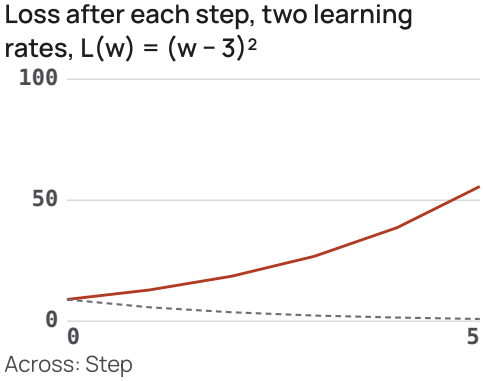
Each step overshoots the bottom, lands further up the far side, and the next slope is steeper still. Within thirty steps the numbers exceed what a float can hold and your loss prints as `nan`. This is not an exotic failure. It is the single most common way training dies, and the fix is to lower the learning rate — usually by a factor of ten and then look again.

Too large diverges. Too small crawls: at a learning rate of 0.0001 the walk above needs tens of thousands of steps. There is no formula for the right value. You try 0.1, 0.01, 0.001 on validation and watch the loss curve.

Real models have many parameters

With two parameters the loss is a surface and the gradient is a pair of numbers, one slope per parameter, each saying how much the loss changes when that one parameter moves. With ten million parameters it is a vector of ten million slopes, and the update rule is the same line of arithmetic applied to each. The mental picture does not need to scale; the arithmetic does, and it does.

Figure 2.1



From w = 0, a rate of 0.1 brings the loss from 9 to under 1 in five steps, each step smaller as the slope flattens. A rate of 1.1 overshoots the bottom every time and lands higher on the far side: 9, 13, 19, 27, 39, 56. Within thirty steps this prints as NaN.

EXAMINATION

Machine Learning, Foundations — final examination

This paper covers the whole course:

- framing a problem as a dataset with a defined row, target and moment of prediction
- fitting and diagnosing linear and logistic models
- building a preprocessing pipeline that leaks nothing
- designing a validation scheme that matches the structure of the data
- measuring a classifier honestly and choosing a threshold
- trees, forests and gradient boosting
- clustering, dimensionality reduction and embeddings
- neural networks and transfer learning
- taking a model to a running service and keeping it honest

63 questions, the whole pool. The site draws 25 for a sitting, pass mark 70%. Work any 25 under your own conditions. Answers from page 555.

1 Module 1 · What learning is, and what it is not

A team's write-up says the tree's max_depth was learned from the data. What is wrong with that description?

A. Depth is chosen by you on validation data, not fitted from the rows.
B. Depth is fitted from the rows, but only once the split thresholds have been chosen.
C. Depth is fitted on the test set, which is where hyperparameters are supposed to be chosen.
D. Nothing is wrong; depth is a parameter of the model exactly like a coefficient.

2 Module 1 · What learning is, and what it is not

A hospital labels 'patients with condition X' using billing codes and trains a model on those labels. What has the model learned?

A. The presence of condition X, since billing codes are audited for accuracy each year.
B. Nothing usable, because billing codes are categorical rather than numeric quantities.
C. Which patients get coded, which varies by clinic and insurer.
D. The severity of condition X, because codes are assigned in bands by severity.

3 Module 1 · What learning is, and what it is not

k-means returns six clusters from a bank's transaction table. What does that establish?

A. That six customer types exist, because the algorithm minimises within-cluster distance.
B. Very little, since six clusters come back from random noise too.
C. That the transactions carry at least six dimensions of genuine structure.
D. That labelling can now begin, because each cluster has a natural name.

4 Module 1 · What learning is, and what it is not

Which grain is the only one that can answer 'should we have approved this application'?

A. One row per customer, aggregated across every loan that customer has held.

B. One row per loan, because that is the level at which default is actually recorded.

C. One row per customer-month, because it supports a rolling thirty-day prediction.

D. One row per application, including the rejected ones.

5 Module 1 · What learning is, and what it is not

A binary model predicts 0.0001 for a row whose true label is 1. Roughly what does log loss charge for that row?

A. About 1.0, because the label is one and the prediction is close to zero.

B. About 9.2, and it rises without limit as the prediction nears zero.

C. About 0.0001, because the loss is simply the predicted probability itself.

D. Exactly 1, since the row is counted as one mistake like any other.

6 Module 1 · What learning is, and what it is not

A team sets class_weight='balanced' and then reads the model's output of 0.6 as a sixty per cent chance. What is the error?

A. There is no error; class weights change the loss but never the predicted probabilities.

B. The error is reading 0.6 at all, because a decision threshold must always be 0.5.

C. Weighting distorts the probabilities, so 0.6 is no longer a rate.

D. Balanced weighting is defined only for trees and not for logistic regression.

Module 1 · What learning is, and what it is not

The module starts on p. 11.

MODULE 1 TEST

Module 1 test, Q1, p. 52
A delivery-time model is trained on data where most trips take 20 to 40 minutes and a handful take three hours because of a road…

Answer A: It fits closer to the median trip and the rare long trips pull it much less.

Why: Minimising squared error fits the mean and minimising absolute error fits the median. Squaring makes one three-hour trip cost as much as many small errors, so the fitted line is dragged towards it; absolute error charges an error of 100 exactly 100, so the long trips lose their special weight and the model settles on the typical trip instead.

Module 1 test, Q2, p. 52
A churn model reaches 0.99 AUC on its first run, with no tuning, on a problem the operations team finds genuinely difficult. What is the…

Answer C: A column recorded after the outcome has reached the feature set.

Why: A result far better than expected, arriving with no effort, is the signature of leakage. On a problem humans find hard, 0.99 is a bug report rather than a result, and the usual cause is a column such as cancellation_reason or account_closed_date that only exists once the outcome has happened. AUC is not inflated by imbalance, and sampling noise does not move a score that far.

Module 1 test, Q3, p. 53
A bank builds a default model where one row is one loan, and customers hold up to eleven loans each. The rows are split at…

Answer D: How well the model predicts for customers whose other loans it trained on.

Why: A random row split scatters one customer′s eleven loans across both sides. Eleven rows share almost every column, so the model can recognise the customer rather than learn the pattern, and the held-out loans belong to customers it has already seen thirty times. The score is inflated by however much of the signal is customer-specific, and the fix is to split by customer.

Module 1 test, Q4, p. 53
A telecom model finds that customers who call support are far more likely to churn. Management proposes making the support line harder to reach. What…

Answer B: The call and the churn share a cause, so removing the call removes the signal.

Why: Prediction requires only that a correlation holds when the world is left alone. Intervention requires knowing what happens when you reach in and change something. The underlying problem causes both the call and the churn, so closing the phone line deletes the measurement while leaving the cause in place, and the frustrated customers now cannot even complain.

Module 1 test, Q5, p. 54

A government office applies an eligibility rule written into a regulation: five conditions, each exactly specified. A team proposes training a classifier on past decisions…

Answer C: The rule is a specification, not a pattern, so a model can only approximate it.

Why: Machine learning earns its cost when a decision rule is real but nobody can write it down. Here it is already written down and exact. A model fitted to past decisions will get 99.4 per cent of them right, which is strictly worse than the if-statement that gets 100 per cent, and it adds a failure mode the regulation does not have.

Module 1 test, Q6, p. 54

A team evaluates on the test set, sees a disappointing number, changes two features, and evaluates again. What is the state of their test estimate…

Answer A: It is optimistic, because a decision was made using it.

Why: The test set is a single-use measuring device. The moment a set is used to choose between options, it has been spent: the chosen model is partly good and partly a match for that set's particular noise. Not training on it is not enough. Iterating against it converts it into a second validation set, and the honest report says how many times it was consulted.

THE LESSON CHECKS

Lesson 1, p. 15

You fit `rent = w * area + b` to 50,000 flats. The true relationship curves: rent rises steeply up to 120 m^2 and then…

Answer A: It has found the best straight line through curved data, and the curve is permanently out of reach for this model.

Why: Learning means adjusting the parameters of a shape you chose until a chosen error number is as small as it can be.

B With 50,000 rows it has enough…: Learning sounds like discovery, so it feels as though a large enough dataset should let the model outgrow the shape it was given.

C Training will drive the loss to…: Training is described as lowering the loss, so it is natural to assume the loss keeps falling toward zero rather than bottoming out at whatever the best line can manage.

D The curve is a sign the…: Dirty data really is the culprit in many bad fits, so blaming the data is a habit that usually pays off — here the data is fine and the model is the problem.

Lesson 2, p. 18

A payments team trains a fraud classifier on a label built from cases their investigators confirmed as fraud. On held-out data it performs well. What…

Answer A: Which transactions resemble the ones this team's investigators already catch.

Why: A supervised model learns your labels, not the reality behind them, so who made the labels is part of the model.

B What fraud looks like in general…: The labels contain no false positives, which feels like the definition of a trustworthy label — the damage is in what the label omits, not in what it asserts.

C Nothing worth deploying, because the labels…: Once bias is named it invites a clean rejection, but a model of the detection process can still rank a review queue usefully as long as everyone knows that is what it is.

D It would avoid this problem if…: No labels sounds like no label bias, but the same features, the same sampled population and the same human interpretation of the clusters carry the bias straight through.

Lesson 3, p. 22

A team building a hospital readmission model has one row per hospital *visit*, and patients often visit many times. They split the rows randomly into…

Answer A: Visits are not independent — the same patient's rows land in both train and test, so the model can recognise the patient rather than the medical pattern.

Why: Decide what one row is and when the target becomes knowable before you write any code, because every split, metric and deployment decision inherits that choice silently.

B AUC is the wrong metric for…: Blames the metric for a data-splitting fault; the same contamination would inflate accuracy, F1 and every other score equally.

C The dataset has too many columns…: Describes ordinary overfitting, which would show up as a train-test gap — here the test set itself is compromised, so the gap never appears.

D Readmission is a rare event, so…: Class imbalance distorts accuracy, but AUC is comparatively robust to base rate and would not be lifted to 0.94 by imbalance alone.

START HERE

Machine Learning, Foundations

The classical ground under modern AI, for someone who can read code.

WHAT IT TEACHES	How a machine actually learns from data: what fitting means mechanically, why the features you choose matter more than the algorithm you pick, how to hold out data honestly, why accuracy lies on rare events, what the classic algorithms are good at, and how gradient descent and neural networks follow from all of it.
WHAT IS INSIDE	Nine modules and 92 lessons, 30 figures drawn from data, nine case studies, nine module tests, a 63-question examination, and every answer with the reason behind it.
LICENCE	CC BY-NC-SA 4.0. Copy, print, share and adapt it for any non-commercial purpose, with credit, under the same licence. There is no paid edition.
THE COURSE	Free to read, with the lessons, the films and the examination, at **addaly.com/learn/machine-learning-foundations**

Scan to open the course

Addaly

First edition, September 2026 · build 62a26075