GO UNDERNEATH

The Open Model Ecosystem

Llama, Mistral, Qwen, DeepSeek, Gemma, Phi: what is real and what is marketing.

First edition, September 2026

Addaly

Series editor: Dr Mustafa Mun

Draft, open for academic review

GO UNDERNEATH

The Open Model Ecosystem

Llama, Mistral, Qwen, DeepSeek, Gemma, Phi: what is real and what is marketing.

Series editor: Dr Mustafa Mun

Addaly

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

The Open Model Ecosystem

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). The Open Model Ecosystem. Addaly. addaly.com/learn/open-models.*

This book is the printed form of the course of the same name at addaly.com/learn/open-models. 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.

7 modules · 71 lessons · 28 figures · 7 case studies · 49,110 words · about 10 hours

Brief contents

1

MODULE 1

What open actually means

Given any model release, say precisely which freedoms it grants and which it withholds, trace its licence through every fine-tune in the chain, verify the files you downloaded are the files the publisher made, and state what an auditor or a regulator could still ask that the weights alone cannot answer.

Lesson 1 · 7 min

Open Weights Is Not Open Source

THE ONE THING TO KEEP

Open weights means you can run and adapt the model, not that you can see how it was made.

What actually lands on your disk

When a lab "open sources" a model, you get a folder. Look inside a typical one:

```
config.json                 # layer count, hidden size, vocab
size
tokenizer.json              # how text becomes numbers
model-00001-of-00004.safetensors
model-00002-of-00004.safetensors
...
README.md                   # the model card
LICENSE
```

That is the model. Billions of floating point numbers, plus enough metadata to load them.

Here is what is not in the folder: the training data. The code that filtered and mixed that data. The training code. The intermediate checkpoints. The human preference data used for the final tuning. The failed runs. In almost every case, none of that ships.

The analogy, and where it breaks

Weights are closer to a compiled binary than to source code. You can run it. You can redistribute it. You cannot rebuild it, and you cannot read what went in.

The analogy breaks in one useful place. You can meaningfully modify a binary blob of weights by fine-tuning it, which is cheap and works. So the practical freedom is larger than "here is a .exe" suggests. It is still not the freedom to reproduce.

Why this matters when you are not being pedantic

Four things you cannot do with weights alone:

- **Audit the data.** You cannot check whether your competitor's private documents, a copyrighted corpus, or the benchmark you are about to run are in there.
- **Reproduce.** You cannot rebuild the model with one thing changed. Nobody outside the lab can.
- **Remove something.** If the model learned a fact you need gone, you can suppress it at inference or fine-tune against it. You cannot delete it from the source.
- **Explain a behaviour by pointing at data.** Every explanation is behavioural, from the outside.

One thing you can do, which is the durable benefit: keep it. If the lab shuts the API down, deprecates the model, changes the price, or decides your use case is off-policy, your copy on disk still runs. That is the real reason open weights matter, and it is enough.

The definition people argue about

The Open Source Initiative published an Open Source AI Definition in October 2024. It asks for three things: the parameters, the complete training and inference code, and detailed information about the data — enough that a skilled person could build a substantially equivalent system. Almost no popular model meets it, because almost none release the data information.

So three labels are worth keeping separate:

- **Open weights.** You get the numbers. Terms vary. This is most of the ecosystem: Llama, Gemma, and much else.

- **Openly licensed weights.** You get the numbers under Apache 2.0 or MIT, with no use restrictions. Mistral's main line, Qwen3, DeepSeek-R1, Phi-4, IBM Granite. Still no data.
- **Fully open.** Weights, data, training code, checkpoints, evaluation. AI2's OLMo, EleutherAI's Pythia, Hugging Face's SmolLM. A small group, and the only one where "open source" is honest without qualification.

Most of the loud releases are in the first two buckets. That is not a scandal. It is just not what the word implies to someone who has used Linux.

Three questions to ask

When somebody tells you a model is open, ask:

1. Can I run it offline, on my own hardware, forever? (Almost always yes.)
2. Can I use it commercially without asking anyone? (Usually, with conditions. Next lesson.)
3. Can I see what went into it? (Almost never.)

If you say "open weights" instead of "open source", you have said the true thing and you have lost nothing. Use the accurate word and you will make better decisions downstream, because the accurate word tells you which questions remain open.

BEFORE YOU MOVE ON

A team plans to adopt an open-weights model in a hiring tool, arguing that being open means they can audit it for bias before deployment. What is the flaw in that argument?

A. Open weights let them measure the model's behaviour on test cases, but not inspect the training data, so the audit can only be behavioural.

B. The model's licence forbids auditing, so the plan is not permitted.

C. There is no flaw, because the weights are a compressed encoding of the training data and can be decoded back into it.

D. The model must meet the OSI Open Source AI Definition before an audit of it would be legally valid.

The answer and why: p. 381

Lesson 2 · 8 min

Reading the Licence Before You Ship

THE ONE THING TO KEEP

A licence travels with every fine-tune and every download; check the chain, not the badge on the page.

I am not a lawyer and this is not legal advice. It is a list of the things that surprise people, so you know what to hand to someone who is.

Three families

Genuine open source. Apache 2.0 or MIT. Mistral's main line, Qwen3, DeepSeek-R1, Phi-4, OLMo, IBM Granite, SmolLM. Keep the notice, do what you like. No user caps, no acceptable use policy, no naming rules. If two candidates are close and one is Apache 2.0, that is a real tiebreaker.

Bespoke community licences. Llama, Gemma, and some Qwen releases. Free for nearly everyone, with conditions attached. Written by the lab, not by lawyers who expected you to redistribute.

Non-commercial or non-production. Cohere's Command R family under CC-BY-NC. Mistral's Codestral under a non-production licence. A large share of research fine-tunes on Hugging Face. You can prototype. You cannot ship.

The clauses people miss

User thresholds. Llama's licence stops being free above 700 million monthly active users, measured before the model's release date. Some Qwen releases use 100 million. This will not bite you. It bites the company that acquires you, and it is the first thing their diligence checks.

Naming and attribution. Llama requires "Built with Llama" displayed prominently, a specific attribution line in your notice file, and any model you derive from it must have a name that *starts with* "Llama". Teams discover this after printing the branding.

Acceptable use policies incorporated by reference. Gemma's terms attach a prohibited use policy that Google can update, and reserve Google's right to restrict use it believes violates that policy. You are agreeing to a document that can change after you ship.

Territory. Llama 4's licence carried a restriction on the multimodal models for entities domiciled in the EU. Read the geography clause if you have European users or a European entity.

What you may do with outputs. Some licences restrict using generations to train other models. Llama's recent terms allow it if the resulting model's name begins with "Llama". If your plan is to generate synthetic training data, this clause is the whole plan.

The inheritance trap

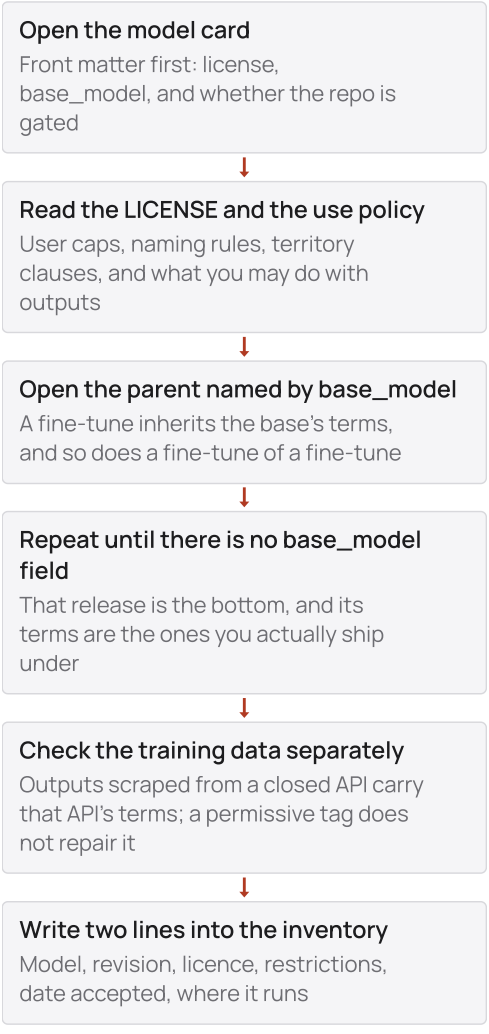
This is the one that catches careful people.

`DeepSeek-R1-Distill-Llama-70B` is a Llama model that was trained on R1's reasoning traces. DeepSeek-R1 itself is MIT. The distill is not. It carries Meta's community licence, because the base weights are Meta's.

A fine-tune inherits the base model's licence. So does a fine-tune of a fine-tune. Follow the chain to the bottom:

Figure 1.1

Tracing a licence to the bottom of the chain



DeepSeek-R1 is MIT. DeepSeek-R1-Distill-Llama-70B is not, because the base weights are Meta's. The badge on the page is the first link in the chain, never the answer.

```
# the front matter of a model card names its parent
hf download deepseek-ai/DeepSeek-R1-Distill-Llama-70B README.md
--local-dir ./card
head -20 ./card/README.md    # look for: license:, base_model:
```

Repeat on the parent. Keep going until you reach something with no `base_model` field.

And separately from the weights: the *data* a fine-tune was trained on has its own terms. A great many popular community fine-tunes were trained on outputs scraped from a closed commercial API, which typically breaches that API's terms of service. An Apache 2.0 badge on the resulting weights does not repair that.

A five minute check

For every model you are seriously considering:

```
hf download <org>/<model> LICENSE USE_POLICY.md README.md --
local-dir ./chk
```

Read the LICENSE. Read the use policy. Read the README front matter for `license`, `base_model`, and whether the repo is gated.

On gated repos: clicking "agree" on Hugging Face is you entering an agreement. A CI job with a shared token that inherits your acceptance is not a separate acceptance, and mirroring the weights to a public bucket to dodge the gate breaks the terms you agreed to.

Write it down once

Keep a two-line-per-model inventory in the repo:

```
Qwen/Qwen3-8B | rev 9a3f21c | Apache-2.0 | no restrictions |
prod: summarizer
google/gemma-3-4b-it | rev 4b1e07d | Gemma Terms | prohibited-
use policy, updatable | prod: on-device
```

It takes ten minutes to start and saves a bad week later, when someone asks what you are shipping and the honest answer has to be assembled from memory.

BEFORE YOU MOVE ON

You fine-tune DeepSeek-R1-Distill-Llama-8B on your own support tickets and want to release the result publicly as "Acme-Assist". Which statement is correct?

A. The distill is built on Llama weights, so Meta's community licence and its naming rules apply to your release; R1's MIT licence does not cover it.

B. DeepSeek-R1 is MIT licensed, so anything distilled from it inherits MIT and you can release freely.

C. Because you fine-tuned it on your own data, the resulting weights are a new original work under whatever licence you choose.

D. MIT and community licences are compatible, so you can pick whichever of the two suits your release best.

The answer and why: p. 382

Lesson 3 · 9 min

Why a lab gives away a model

THE ONE THING TO KEEP

A lab releases weights when a cheap, ubiquitous model helps something else it sells, so openness tracks strategy rather than principle and can be withdrawn from the next version at any time.

The question worth asking first

Training a model at the top of the open ecosystem costs somewhere between a few million and a few hundred million dollars in compute alone, before salaries. Then the lab publishes the result for anyone to download. If you cannot explain why, you will be repeatedly surprised by what happens next: a licence that tightens, a family that goes quiet, a flagship that stays behind an API while only the small siblings ship.

CASE STUDY · MODULE 1

The MIT badge that was not theirs

An illustrative case, built from documented patterns.

A four-person education company in Pune, three weeks from shipping a doubt-solving app to Maharashtra state board students.

Ketki and her three colleagues had built the thing properly. Their app took a photograph of a maths problem from a Class 10 textbook, read it, and walked a student through the solution in Marathi or English. They had chosen a reasoning model because a wrong early step ruins a proof, and the distilled reasoning model they picked was good enough that two teachers at a pilot school had stopped correcting it.

The repository they had built on was `DeepSeek-R1-Distill-Llama-8B`. The DeepSeek papers were public, the R1 release was MIT, and MIT is about as free as a licence gets. Ketki had written "MIT" in the one-line technology section of the deck she showed schools, and nobody had asked a second question about it.

The second question arrived from a procurement officer at a chain of eleven schools in Nashik, who wanted a signed statement of every third-party component and its licence before a purchase order could be raised. It was a form. Her lawyer filled it in and then stopped at the model row and asked where the MIT text was.

It was not there. The repository's front matter named a `base_model`, and the base was Meta's. DeepSeek-R1 is MIT; a distillation into Llama weights is not, because the weights being distilled into are Meta's and Meta's community licence travels with them. Ketki had been reading the licence of the model that produced the training traces rather than the licence of the model she was actually running.

What the real licence asked for

None of it was onerous in isolation. The Llama community terms wanted "Built with Llama" displayed prominently. They wanted a specific attribution line in a notice file. They required that any model she derived from it have a name beginning with "Llama" — and she had fine-tuned an adapter on four thousand solved problems and called the result Ganak. They attached an acceptable use policy that Meta could update after she shipped, and they stopped being free above seven hundred million monthly active users, which was not her problem and would be the first thing any acquirer's diligence checked.

The app icon, the splash screen and two thousand printed flyers for the Nashik schools all said Ganak. The flyers had gone to press eleven days earlier.

The two ways out, both expensive

Comply. Rename the model — not the app, only the model, which the terms actually constrain — to something beginning with Llama, add the attribution, put the badge in the About screen, reprint nothing. Cost: about a day of work, a conversation with the procurement officer explaining that the licence in her deck had been wrong, and a permanent dependency on a document another company can edit. Her lawyer's note was blunt: you are agreeing now to terms that can change later, and you cannot prove which version you accepted unless you keep a copy with a date.

Swap. Move to Qwen3-8B, which is Apache 2.0 with no naming rule, no use policy and no user cap. Cost: the prompt was tuned against the distilled model's habit of thinking before answering and did not transfer. The adapter would have to be retrained against a different base. Their forty-example evaluation would have to be rerun. Ketki's estimate was nine working days, which put them past the start of the school term, which was the only window that mattered that year.

The third option, which one of the four argued for at length, was to ship and deal with it later. The licence was not being enforced against anybody they knew of. The procurement form could say MIT and nobody would check.

What made it a decision rather than a choice

The argument that ended it was not about risk of enforcement. It was that they were about to sign a statement of components for a school, and the statement would be false. Ketki said she was not willing to put her name on it, and that settled the immediate question without settling the larger one, which was what they should be running a year from now.

YOUR ANSWERS

1. Ketki read the licence of DeepSeek-R1 and concluded her model was MIT. Which specific field would have caught the mistake in fifteen seconds, and why does that field exist?

2. Her colleague argued that nobody enforces these terms and the form could say MIT. Setting aside enforcement, what does that argument cost the company later?

3. The swap to an Apache 2.0 base was estimated at nine days, mostly prompt and adapter work. What does that estimate tell you about a decision they made much earlier?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 56.

This page is blank so that how it played out stays out of view until you turn the page.

CASE STUDY · MODULE 1

How it played out

They complied for the launch and swapped afterwards. The model was renamed to Llama-Ganak-8B inside the app, "Built with Llama" went into the About screen and the notice file, and the procurement form was corrected before it was signed — which cost one awkward phone call and no schools. The nine-day swap to an Apache 2.0 base was scheduled for the gap between terms and took eleven days. What survived as a permanent change was smaller and more useful than either: a two-line-per-model inventory in the repository listing model, revision, licence, date accepted and where it runs, plus a copy of every licence and use policy at the version they accepted. When Meta's terms were next edited, they could see what had changed rather than guessing.

The questions, discussed

1. Ketki read the licence of DeepSeek-R1 and concluded her model was MIT. Which specific field would have caught the mistake in fifteen seconds, and why does that field exist?

The `base_model` field in the model card's YAML front matter. It names the parent a fine-tune was built from, and a fine-tune inherits the base model's licence — as does a fine-tune of a fine-tune. Reading the front matter and following `base_model` upward until a repository has no such field is the whole check. Its absence on an obvious derivative is itself a signal: the uploader either did not know or chose not to say.

2. Her colleague argued that nobody enforces these terms and the form could say MIT. Setting aside enforcement, what does that argument cost the company later?

Three things. A false component statement to a customer is a representation the company has made in writing, and correcting it later is worse than correcting it now. The user-cap and naming clauses are exactly what an acquirer's or investor's diligence checks first, so the problem surfaces at the moment with the least room to fix it. And a team that does not know which

licence it is under cannot answer what it may do with the model's outputs, which is the clause that decides whether they can ever generate synthetic training data from it.

3. The swap to an Apache 2.0 base was estimated at nine days, mostly prompt and adapter work. What does that estimate tell you about a decision they made much earlier?

That they had designed around a single family. Prompts do not transfer between families, and an adapter is bound to the base revision it was trained against, so the cost of moving was real — but nine days for an 8B swap is the lock-in that open weights were supposed to prevent. Keeping the prompt versioned per model, the evaluation harness model-agnostic behind one OpenAI-compatible endpoint, and a copy of the weights they depend on turns a nine-day migration into an afternoon plus a measurement.

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. 380. If two go wrong, re-read the module before the exam.

1 DeepSeek-R1 is released under MIT. A team plans to ship DeepSeek-R1-Distill-Llama-70B commercially and files it in their inventory as MIT. What have they got wrong?

A. The base weights are Meta's, so the distill carries Meta's community licence

B. MIT permits commercial use only under a separate attribution agreement

C. A derivative carries no licence until its uploader selects one

D. Training on another model's reasoning traces transfers that model's MIT licence to the result

2 A customer's security review asks whether the benchmark a team quotes was in the model's training data. Which axis of the six-axis openness rubric do they need?

A. Licence, because a permissive licence grants the right to audit the corpus

B. Evaluation artefacts, because raw per-task outputs expose memorised answers

C. Checkpoints, because intermediate checkpoints reveal exactly when a benchmark score jumped

D. Training data, because only searching the corpus can answer it

3 A publisher adds CCBot to robots.txt in 2026 after finding their archive quoted verbatim by an open model. What does that achieve?

A. It removes the publisher's pages from existing Common Crawl snapshots

B. It makes the model unlearn the archive at its next fine-tuning run, because opt-out signals propagate through derivatives

C. Nothing for the trained model; it only affects crawls made after the change

D. It obliges the lab to retrain without those documents at the next release

4 A team refuses every pickle file and downloads only safetensors, then loads a new architecture with trust_remote_code set to true. What have they left open?

A. The tokenizer, because tokenizer.json is a pickle in several repositories

B. The flag imports Python from the repository, which then runs on their machine

C. Only the risk that a third party quantised the file without stating the source revision

D. Nothing; safetensors cannot execute code, so the whole load is now safe

5 Two community fine-tunes of the same base are candidates. One has two hundred thousand downloads and a benchmark table naming no harness. The other has nine hundred downloads, names its datasets and lists what it is bad at. Which is the stronger evidence?

A. The second, because writing down weaknesses means somebody actually evaluated it

B. The first, because a benchmark table is comparable across cards even without a harness

C. Neither, because only the base model's own evaluation transfers to a fine-tune

D. The first, because downloads aggregate many people's judgement of quality

6 A European team publishes a fine-tune under Apache 2.0 and assumes the AI Act's open-source exemption removes every general-purpose model duty. Which duties survive the exemption?

A. None, because a free and open-source licence removes every provider obligation

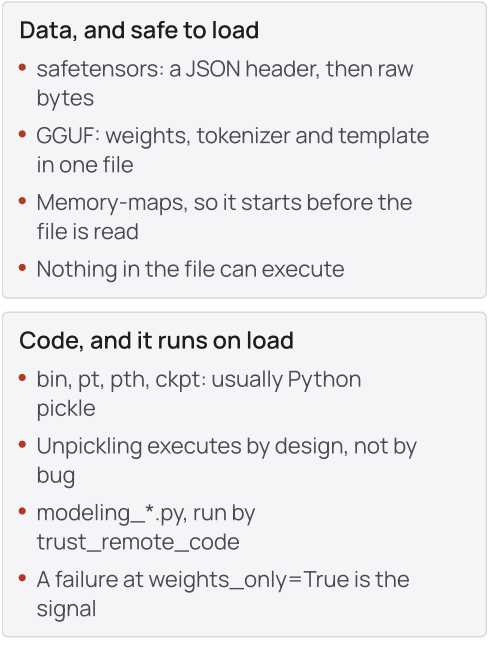
B. Only the systemic-risk duties, which attach to any published fine-tune regardless of scale

C. The technical documentation and a third-party conformity assessment

D. The copyright policy and the published training-content summary

Figure 2.1

Two classes of file in the same repository



Hub-side scanning is a filter, not a proof: it catches known-bad pickle patterns and cannot see a backdoor in the weights. If a repository has safetensors, use them. If it has only pickle, convert it yourself in a container with no credentials and no network.

In practice: if a repo has safetensors, use them. If it only has `.bin`, either find a converted mirror, convert it yourself inside a container with no network and no credentials, or skip the model. And keep PyTorch's safe default:

```
import torch
sd = torch.load("pytorch_model.bin", weights_only=True)  #
refuses arbitrary code; the default since torch 2.6
```

If loading fails with `weights_only=True`, that is a signal, not an inconvenience to switch off.

Sizes, so you can plan a download

Before you start a 140GB transfer on a metered connection, do the arithmetic. Roughly two bytes per parameter at bf16:

Model	bf16 files	Q8 GGUF	Q4_K_M GGUF
4B	~8 GB	~4.3 GB	~2.5 GB
8B	~16 GB	~8.5 GB	~4.9 GB
27B	~54 GB	~29 GB	~16 GB
70B	~140 GB	~75 GB	~42 GB

Fetch only what you need:

```
hf download bartowski/Qwen2.5-7B-Instruct-GGUF \
  --include "*Q4_K_M.gguf" --local-dir ./gguf
```

The rest of the hub

Datasets are the same git plus parquet arrangement. Spaces are hosted demos, which are useful for trying a model for thirty seconds before committing to a download. Inference providers route API calls to third parties serving those same open weights, which is how you test six models without downloading any of them — the subject of the last lesson.

EXAMINATION

The Open Model Ecosystem — final examination

This paper covers the whole course:

- what open weights grant and withhold and how a licence travels down a chain of fine-tunes
- the map of families, modalities and serving options
- reading a repository's own files to predict size, memory, token cost and conversation format
- sizing, quantising and serving a model on hardware you actually have
- fixing bad output at the sampler, the prompt, the schema or the guardrail
- deciding whether to fine-tune and proving what it cost you
- settling a model choice with evidence you produced yourself

56 questions, the whole pool. The site draws 25 for a sitting, pass mark 70%. Work any 25 under your own conditions. Answers from page 429.

1 Module 1 · What open actually means

A lab deprecates a hosted model a team depends on. Given that the team can neither audit nor reproduce the open model they switch to, which property actually protects them?

A. The training data is available, so they could rebuild the model if needed

B. The licence obliges a successor release under the same terms

C. The Open Source Initiative's definition obliges the lab to keep publishing the weights

D. Their copy on disk still runs, whatever the lab decides next

2 Module 1 · What open actually means

The Open Source Initiative's AI definition asks for three things, and almost no popular model meets it. Which requirement do they overwhelmingly fail?

A. Detailed information about the training data

B. An OSI-approved licence attached to the weights

C. Downloadable parameters

D. Inference code that a third party can run

3 Module 1 · What open actually means

A team fine-tunes a Llama model and publishes it as Sahayak-7B under Apache 2.0. Name the two obligations they have broken.

A. The user cap and the territorial restriction, both of which bind any derivative

B. The naming prefix and a licence they had no right to grant

C. The acceptable use policy, which forbids publishing derivatives at all

D. Nothing; a fine-tune is a new work and its author chooses the licence

4 Module 1 · What open actually means

A company whose only product is its model releases the weights for free. Which outcome do the economics predict most strongly?

A. It will become the most widely deployed family within a year
B. The licence must stay permissive, because reverting it is legally impossible
C. The terms tighten, or the weights become a paid tier, at some later release
D. The model will improve faster than one from a lab with a cloud business

5 Module 1 · What open actually means

An audit of a cleaned English web corpus found its bad-word filter had removed a disproportionate amount of text written in African American English and text about LGBT topics. What does that illustrate?

A. Bad-word filtering is the only step that removes text; deduplication only compresses
B. Language identification is the step that removes the most data at any scale
C. Filtering improves quality uniformly, at a small cost in corpus size
D. Filters encode a judgement, and remove some communities′ text more than others

6 Module 1 · What open actually means

A founder asks whether they can own the marketing copy a model generated for them. Which of the three copyright questions is that?

A. Whether the output can be protected, which turns on human authorship
B. Whether training infringes, which the lab's fair-use position settles
C. Whether the output infringes, which turns on similarity to a protected work
D. All three at once, because they are always answered together

Module 1 · What open actually means

The module starts on p. 9.

MODULE 1 TEST

Module 1 test, Q1, p. 58

DeepSeek-R1 is released under MIT. A team plans to ship DeepSeek-R1-Distill-Llama-70B commercially and files it in their inventory as MIT. What have they got wrong?

Answer A: The base weights are Meta's, so the distill carries Meta's community licence

Why: A licence travels with the weights, not with the data that shaped them. The distill is a Llama model that was trained on R1's traces, so the parameters on disk are Meta's and Meta's terms apply — the naming rule, the attribution line and the user cap included. Follow the base_model field down the chain until you reach a repository that has none; that is where the obligations start.

Module 1 test, Q2, p. 58

A customer's security review asks whether the benchmark a team quotes was in the model's training data. Which axis of the six-axis openness rubric do…

Answer D: Training data, because only searching the corpus can answer it

Why: Each axis buys one concrete capability, and contamination checking is bought by data disclosure alone. If the corpus is published you can grep it and answer the question; if it is not, every answer is a guess. That is why a release scoring two on weights and zero on data can still be useless for a regulator or a security review, and why the rubric is more useful than arguing about the word open.

Module 1 test, Q3, p. 59

A publisher adds CCBot to robots.txt in 2026 after finding their archive quoted verbatim by an open model. What does that achieve?

Answer C: Nothing for the trained model; it only affects crawls made after the change

Why: Every opt-out mechanism is prospective. Snapshots already taken are published and mirrored, and a model that has already learned from them cannot have that learning removed short of retraining. The line in robots.txt is worth adding, but it protects work published from today onwards, not the archive that was on the open web in 2019.

Module 1 test, Q4, p. 59

A team refuses every pickle file and downloads only safetensors, then loads a new architecture with trust_remote_code set to true. What have they left open?

Answer B: The flag imports Python from the repository, which then runs on their machine

Why: Safetensors closes one execution path: the weights file is a header plus raw bytes and loading it runs nothing. The flag opens a different one by design — it downloads the repository's own modelling and configuration Python and imports it. The remedy is unglamorous: read the file, which is usually a few hundred lines of tensor operations, and pin the revision so it cannot change under you afterwards.

Module 1 test, Q5, p. 60

Two community fine-tunes of the same base are candidates. One has two hundred thousand downloads and a benchmark table naming no harness. The other has…

Answer A: The second, because writing down weaknesses means somebody actually evaluated it

Why: Downloads track whichever quantised repository got linked from a popular post, and popularity in this ecosystem moves in days. A limitations section costs the uploader effort and admits something unflattering, which is exactly why it carries information: nobody can write one without having run the model on things it failed. It is the strongest single signal on a stranger's model card.

Module 1 test, Q6, p. 60

A European team publishes a fine-tune under Apache 2.0 and assumes the AI Act's open-source exemption removes every general-purpose model duty. Which duties survive the…

Answer D: The copyright policy and the published training-content summary

Why: The exemption is partial and two duties sit outside it: a copyright policy that respects machine-readable rights reservations, and a published summary of the training content following the AI Office template. Both survive an open release. The exemption also stops entirely above the systemic-risk compute threshold, which is a frontier-scale number that ordinary fine-tuning is nowhere near.

THE LESSON CHECKS

Lesson 1, p. 13

A team plans to adopt an open-weights model in a hiring tool, arguing that being open means they can audit it for bias before deployment…

Answer A: Open weights let them measure the model's behaviour on test cases, but not inspect the training data, so the audit can only be behavioural.

Why: Open weights means you can run and adapt the model, not that you can see how it was made.

B The model's licence forbids auditing, so…: Confuses use restrictions in a licence with a lack of technical inspection rights; licences rarely restrict testing.

C There is no flaw, because the…: Treats the weights as a lossless archive of the corpus rather than a lossy statistical summary.

D The model must meet the OSI…: Treats a community definition as a legal permission gate rather than a naming standard.

Lesson 2, p. 18

You fine-tune DeepSeek-R1-Distill-Llama-8B on your own support tickets and want to release the result publicly as "Acme-Assist". Which statement is correct?

Answer A: The distill is built on Llama weights, so Meta's community licence and its naming rules apply to your release; R1's MIT licence does not cover it.

Why: A licence travels with every fine-tune and every download; check the chain, not the badge on the page.

B DeepSeek-R1 is MIT licensed, so anything…: Assumes the teacher model's licence flows to the student, when the licence follows the base weights that were actually trained.

C Because you fine-tuned it on your…: Treats a derivative work as an original one; fine-tuning modifies licensed weights rather than replacing them.

D MIT and community licences are compatible…: Treats licence selection as a preference rather than an obligation determined by the upstream terms.

Lesson 3, p. 21

A company with no product other than its model releases strong open weights under Apache 2.0. What is the most useful inference to draw about…

Answer A: The Apache licence is irrevocable for the version released, so keep your own copy of those weights and do not assume the successor arrives on the same terms.

Why: A lab releases weights when a cheap, ubiquitous model helps something else it sells, so openness tracks strategy rather than principle and can be withdrawn from the next version at any time.

B Apache 2.0 cannot be changed, so…: Confuses the licence on a released artefact with a promise about future artefacts; the terms on version 2 are set fresh by whoever owns it then.

C The release is unsustainable, so the…: Treats a business-model concern as a quality signal; the weights are as good as they measure, whatever happens to the company.

D Because the licence is permissive, the…: Conflates the licence on the weights with anything at all about the data, which is a separate question the licence does not answer.

GO UNDERNEATH

The Open Model Ecosystem

Llama, Mistral, Qwen, DeepSeek, Gemma, Phi: what is real and what is marketing.

WHAT IT TEACHES	Llama, Mistral, Qwen, DeepSeek, Gemma and Phi are not interchangeable, and the leaderboard that ranks them is mostly measuring itself.
WHAT IS INSIDE	Seven modules and 71 lessons, 28 figures drawn from data, seven case studies, seven module tests, a 56-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/open-models**

Scan to open the course

Addaly

First edition, September 2026 · build 62a26075