The AI/ML Stack

10 books. 4 tiers. A book-by-book guide: who each one suits, what to know first, what to focus on, and a project to build.

How to read this guide

- This is a selected ten-book list, not a universal ranking.
- Tiers are a suggested grouping. Start where you are, not where the list starts.
- Level labels come from the publisher where one exists. "Prerequisites" and "focus" are my reading of each book's contents.
- Don't just collect books. Code along, take notes and ship one small project from each tier.

Follow @athrvam__ on Instagram for more breakdowns like this.

The stack at a glance

#	BOOK	TIER	LEVEL
1	**AI Engineering** Chip Huyen	Tier 1	Intermediate to advanced
2	**Build a Large Language Model (From Scratch)** Sebastian Raschka	Tier 1	Beginner to intermediate
3	**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow** Aurélien Géron	Tier 1	Intermediate to advanced
4	**Understanding Deep Learning** Simon J.D. Prince	Tier 2	Level label not given by publisher. Math-forward but written for newcomers.
5	**Designing Data-Intensive Applications** Martin Kleppmann (1st edition); Martin Kleppmann and Chris Riccomini (2nd edition)	Tier 2	Intermediate to advanced
6	**Designing Machine Learning Systems** Chip Huyen	Tier 2	Intermediate to advanced
7	**Natural Language Processing with Transformers** Lewis Tunstall, Leandro von Werra, Thomas Wolf	Tier 3	Intermediate to advanced
8	**Hands-On Large Language Models** Jay Alammar, Maarten Grootendorst	Tier 3	Beginner to intermediate
9	**Deep Learning** Ian Goodfellow, Yoshua Bengio, Aaron Courville	Tier 4	Level label not given by publisher. Textbook, theory-heavy.
10	**The Alignment Problem** Brian Christian	Tier 4	Non-technical. Narrative nonfiction.

Tier meanings

- **Tier 1, must-reads:** the three books placed first for people who want to build.
- **Tier 2, core skills:** deep learning theory, data systems and ML systems. These age slowly.
- **Tier 3, the LLM era:** transformers and large language models in practice.
- **Tier 4, go deeper:** theory and the human side of AI.

Tier 1 means "high priority", not "easiest". Hands-On Machine Learning is the most beginner-friendly of the three. AI Engineering and the LLM-from-scratch book are easier once you can code in Python and know basic ML terms.

Where should you start?

If you are new to ML
Hands-On Machine Learning (Part I), then Understanding Deep Learning (free), then Hands-On Large Language Models.

If you code and want to build with LLMs
Hands-On Large Language Models, then AI Engineering, then Build a Large Language Model From Scratch.

If you want to ship models to real users
Designing Machine Learning Systems, then Designing Data-Intensive Applications, then AI Engineering.

If you want theory
Understanding Deep Learning, then Deep Learning (Goodfellow et al.) as reference, then The Alignment Problem to zoom out.

One project per tier

TIER	PROJECT TO SHIP
1	A small RAG bot over your own notes, with 20 test questions to score it.
2	An ML API that logs its inputs and warns when the data drifts.
3	A semantic search tool or a fine-tuned text classifier on your own data.
4	A one-page risk note for a model you built: where it can be unfair, opaque or wrong.

Suggestion: one book at a time. Read a chapter, run the code, write three lines of notes. Finish one book before buying the next.

01

Tier 1 · Must-reads

AI Engineering

Chip Huyen

O'Reilly, Dec 2024 · O'Reilly level: intermediate to advanced

WHO IT SUITS

Developers who want to build products on top of existing foundation models, not train them from zero.

BEFORE YOU START

Comfortable Python and basic ML vocabulary (what training, loss and fine-tuning mean). You do not need to have trained a model yourself.

WHAT'S INSIDE

Intro to AI engineering and the AI stack; foundation models; evaluation methodology and evaluating AI systems; prompt engineering; RAG and agents; finetuning; dataset engineering; inference optimization; architecture and user feedback.

FOCUS ON

- Chapters 3 and 4 on evaluation. Evaluation is worth learning early.
- Chapter 5 on prompt engineering, including defenses against prompt attacks.
- Chapter 6 on RAG and agents, to see how the pieces connect.
- Chapter 10 for how a real AI app architecture grows step by step.

GO SLOW OR SKIP

Chapters 7 to 9 (finetuning memory math, data synthesis, inference optimization) get heavy. Read them when you hit those problems, not before.

Project to try: Build a small RAG question-answering bot over your own notes. Write 20 test questions first and score its answers, so you practise evaluation, not just prompting.

Code and resources

02

Tier 1 · Must-reads

Build a Large Language Model (From Scratch)

Sebastian Raschka

Manning, Sept 2024 · O'Reilly level: beginner to intermediate

WHO IT SUITS

Anyone who wants to understand how a GPT-style model works by coding every part of it.

BEFORE YOU START

Python, and a first contact with PyTorch helps. Basic neural network ideas help, but the book builds up from text data.

WHAT'S INSIDE

Understanding LLMs; working with text data; coding attention mechanisms; implementing a GPT model to generate text; pretraining on unlabeled data; fine-tuning for classification; fine-tuning to follow instructions.

FOCUS ON

- Chapter 3 on attention. Type it out and print the tensor shapes.
- Chapter 4, where the pieces become one working GPT model.
- Chapter 5 on pretraining and loading pretrained weights.
- Chapters 6 and 7 for turning the model into a classifier and an instruction follower.

GO SLOW OR SKIP

Do not copy-paste the code. The point is that you wrote it. A GPU is recommended for training, but the author's materials are built to run on an ordinary laptop for the main path.

Project to try: Train your own tiny GPT on a small text file (song lyrics, a book you love), then fine-tune it as a spam or topic classifier.

Book companion page · Code

03

Tier 1 · Must-reads

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

Aurélien Géron

O'Reilly, 3rd edition, Oct 2022 · O'Reilly level: intermediate to advanced

WHO IT SUITS

A code-first route into classical ML and deep learning. Good if you learn by running notebooks.

BEFORE YOU START

Python and basic NumPy and pandas. Some comfort with linear algebra and probability helps, but the book teaches the ideas as it codes them.

WHAT'S INSIDE

Part I, fundamentals: ML landscape; an end-to-end project; classification; training models; SVMs; decision trees; ensembles and random forests; dimensionality reduction; unsupervised learning. Part II, neural networks: Keras; training deep nets; custom models in TensorFlow; data loading; CNNs; sequences with RNNs and CNNs; NLP with RNNs and attention; autoencoders, GANs and diffusion; reinforcement learning; deployment at scale.

FOCUS ON

- Chapter 2, the end-to-end project. Redo it on a dataset of your own.
- Chapters 3 and 4 for evaluation metrics and how training really works.
- Chapter 7 on ensembles. Worth knowing well.
- Chapters 10 and 11 as your first neural networks.

GO SLOW OR SKIP

It is long (over 800 pages). Part I is a good first goal. The TensorFlow-specific chapters in Part II matter less if you plan to work in PyTorch. The ideas still transfer.

Project to try: Pick any tabular dataset, run the Chapter 2 workflow end to end, and write a one-page report of what worked and what did not.

Notebooks

04

Tier 2 · Core skills

Understanding Deep Learning

Simon J.D. Prince

MIT Press, 2023 · Level label not given by publisher. Math-forward but written for newcomers.

WHO IT SUITS

Learners who want the why behind deep learning with clear figures, before or alongside coding.

BEFORE YOU START

Comfort with functions, vectors and basic probability. Chapters build on each other, so go in order early on.

WHAT'S INSIDE

Supervised learning; shallow and deep networks; loss functions; training; gradients and initialization; measuring performance; regularization; CNNs; residual networks; transformers; graph networks; unsupervised learning; GANs; normalizing flows; VAEs; diffusion models; deep reinforcement learning; why deep learning works.

FOCUS ON

- Chapters 1 to 9 as a complete foundation.
- Chapter 12 on transformers and the notebooks on self-attention and tokenization.
- Chapter 18 on diffusion models if image generation interests you.

GO SLOW OR SKIP

Chapters 13 and 15 to 17 are specialist topics. Come back to them when you need them. The book has 68 Python notebooks with missing code for you to fill in, so use them.

Project to try: Fill in the notebooks for chapters 3 to 7, then re-implement a small network from scratch in NumPy and compare it with the notebook.

Free PDF, notebooks and slides (author's site)

05

Tier 2 · Core skills

Designing Data-Intensive Applications

Martin Kleppmann (1st edition); Martin Kleppmann and Chris Riccomini (2nd edition)

O'Reilly, 1st ed. 2017; 2nd ed. Feb 2026 · O'Reilly level: intermediate to advanced

WHO IT SUITS

Anyone who will put ML into real systems and needs to understand databases, pipelines and distributed data.

BEFORE YOU START

Some programming and database exposure. No ML needed. It is dense, so read slowly.

WHAT'S INSIDE

Trade-offs in data systems; nonfunctional requirements; data models and query languages; storage and retrieval; encoding and evolution; replication; sharding; transactions; distributed systems trouble; consistency and consensus; batch processing; stream processing; a philosophy of streaming systems; doing the right thing.

FOCUS ON

- Chapters 1 and 2 to learn the vocabulary: reliability, scalability, maintainability.
- Chapter 4 on storage and retrieval and Chapter 5 on encoding.
- Chapters 11 and 12 on batch and stream processing, the closest to ML pipelines.

GO SLOW OR SKIP

Chapters 9 and 10 (distributed systems, consensus) are the hardest. Skim first, return later. The chapter list here is from the 2nd edition (2026). If you have the 1st edition, the core ideas carry over, but chapters differ.

Project to try: Draw the data flow of an ML app you use or build: where data is stored, how it is replicated, where it can go stale. Then write down three failure cases.

Publisher page (O'Reilly)

06

Tier 2 · Core skills

Designing Machine Learning Systems

Chip Huyen

O'Reilly, May 2022 · O'Reilly level: intermediate to advanced

WHO IT SUITS

People moving from notebooks to models that run in production.

BEFORE YOU START

You have trained at least one model. Python and basic ML evaluation ideas.

WHAT'S INSIDE

Overview of ML systems; system design; data engineering fundamentals; training data; feature engineering; model development and offline evaluation; deployment and prediction service; data distribution shifts and monitoring; continual learning and testing in production; infrastructure and tooling for MLOps; the human side of ML.

FOCUS ON

- Chapter 2 for how to frame a project from business goal to ML task.
- Chapters 4 and 5 on training data and features.
- Chapter 8 on distribution shifts and monitoring. Worth reading before you deploy.

GO SLOW OR SKIP

Chapter 10 on tooling can date quickly, so treat tool names as examples and learn the concepts.

Project to try: Take a model you trained, wrap it in a small API, log its inputs and outputs, and write a monitoring check that warns when the inputs drift.

Resources

07

Tier 3 · The LLM era

Natural Language Processing with Transformers

Lewis Tunstall, Leandro von Werra, Thomas Wolf

O'Reilly, May 2022 (revised edition listed) · O'Reilly level: intermediate to advanced

WHO IT SUITS

People who want to use the Hugging Face ecosystem to do real NLP tasks.

BEFORE YOU START

Python and PyTorch basics. A first idea of what a neural network is.

WHAT'S INSIDE

Hello Transformers; text classification; transformer anatomy; multilingual named entity recognition; text generation; summarization; question answering; making transformers efficient in production; dealing with few to no labels; training transformers from scratch; future directions.

FOCUS ON

- Chapter 2 for the full fine-tune workflow on a small task.
- Chapter 3 on transformer anatomy.
- Chapter 8 on making models faster and smaller.
- Chapter 9 for when you have little labeled data.

GO SLOW OR SKIP

Check library versions. Hugging Face tools change fast, so run the official notebooks first and fix version errors before changing the code.

Project to try: Fine-tune a small model to classify your own text (support messages, notes, headlines). Then compare it with a zero-shot baseline.

Notebooks

08

Tier 3 · The LLM era

Hands-On Large Language Models

Jay Alammar, Maarten Grootendorst

O'Reilly, Sept 2024 · O'Reilly level: beginner to intermediate

WHO IT SUITS

A visual, practical tour of how LLMs work and how to use them. A good first LLM book.

BEFORE YOU START

Python. No deep ML background required, but basic ML terms help.

WHAT'S INSIDE

Part I: introduction to LLMs; tokens and embeddings; looking inside LLMs. Part II: text classification; text clustering and topic modeling; prompt engineering; advanced text generation tools; semantic search and RAG; multimodal LLMs. Part III: creating text embedding models; fine-tuning representation models for classification; fine-tuning generation models.

FOCUS ON

- Chapters 1 to 3 for the picture of how an LLM is built.
- Chapter 8 on semantic search and RAG.
- Chapters 10 to 12 once you want to train and fine-tune.

GO SLOW OR SKIP

The book uses many libraries and models. Run the repo notebooks in the cloud first if your laptop has no GPU.

Project to try: Build a semantic search over a folder of PDFs or notes, then add a small answer-generation step on top.

Code

09

Tier 4 · Go deeper

Deep Learning

Ian Goodfellow, Yoshua Bengio, Aaron Courville

MIT Press, 2016 · Level label not given by publisher. Textbook, theory-heavy.

WHO IT SUITS

Readers who want the mathematical foundations and are ready for a textbook.

BEFORE YOU START

Linear algebra, probability and calculus. Part I of the book reviews them.

WHAT'S INSIDE

Part I: applied math and ML basics (linear algebra, probability, numerical computation, ML basics). Part II: modern practical deep networks (feedforward nets, regularization, optimization, CNNs, sequence models, practical methodology, applications). Part III: deep learning research (linear factor models, autoencoders, representation learning, probabilistic models, Monte Carlo methods, partition function, approximate inference, deep generative models).

FOCUS ON

- Part I as a math refresher, only what you need.
- Chapters 6 to 8 for the core mechanics.
- Chapter 11 on practical methodology.

GO SLOW OR SKIP

Published in 2016, so it does not cover transformers or LLMs. Part III is research-level, so skip it on a first pass. Use it as reference, not a cover-to-cover read.

Project to try: Pick one chapter (for example regularization), derive one result by hand, then verify it with a short experiment.

Free online version (authors' site)

10

Tier 4 · Go deeper

The Alignment Problem

Brian Christian

W. W. Norton, 2020 · Non-technical. Narrative nonfiction.

WHO IT SUITS

Anyone who wants to understand what goes wrong when AI learns from data, and how researchers try to fix it.

BEFORE YOU START

None. It is the least technical book in the stack, even though it sits in the last tier.

WHAT'S INSIDE

Part I, Prophecy: representation, fairness, transparency. Part II, Agency: reinforcement, shaping, curiosity. Part III, Normativity: imitation, inference, uncertainty.

FOCUS ON

- Part I if you build models that affect people.
- Part II if reinforcement learning or RLHF interests you.
- Part III for how systems learn what humans want.

GO SLOW OR SKIP

Written in 2020, so it predates the chat-assistant era. Read it for the ideas and history, not the latest methods.

Project to try: Pick one model you built. Write a short note on how it could be unfair, opaque or wrong in a way that matters, and one check you would add.

Author's page

Notes and sources

Also seen in the reel

The reel's animated cards also show *Transformers and Large Language Models: A Hands-On Guide to RAG and Agentic AI* by Ahmed Fawzy Gad (Apress, 2026). It is not part of the ten on the shelf, so it is not ranked here. It covers machine learning basics, transformers, advanced transformer architectures, model customization and retrieval-augmented generation.

Free, legal copies

Understanding Deep Learning and Deep Learning (Goodfellow et al.) are free online from their authors' sites, linked above. For the other books, buy them or borrow them. No pirated copies here.

Editions and versions

Designing Data-Intensive Applications has a 2nd edition (2026, with Chris Riccomini as co-author). The shelf photo may show the 1st edition, and the chapter list in this guide is from the 2nd. Library versions in code repos change, so run each book's official notebooks first.

WHERE FACTS COME FROM

- O'Reilly book pages: AI Engineering, Designing Machine Learning Systems, Hands-On Machine Learning, Designing Data-Intensive Applications, NLP with Transformers, Hands-On Large Language Models
- Manning and O'Reilly pages for Build a Large Language Model (From Scratch)
- udlbook.github.io and deeplearningbook.org
- brianchristian.org and publisher catalogue records for The Alignment Problem
- Springer page for the Gad book

Chapter lists are from those pages. "Focus" and "Before you start" are suggestions, not the authors' words.

Made by ATHRVAM

Follow @athrvam__ for more such breakdowns.