Machine Learning Platform Engineering

Build an internal developer platform for ML and AI systems

Benjamin Tan Wei Hao
Shanoop Padmanabhan
Varun Mallya

MANNING

Machine Learning Platform Engineering

Build an internal developer platform for ML and AI systems

Benjamin Tan Wei Hao
Shanoop Padmanabhan
Varun Mallya

MANNING
Shelter Island

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity.

For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Development editor: Doug Rudder
Technical editor: Brian E. Heath
Review editor: Kishor Rit
Production editor: Andy Marinkovich
Copy editor: Julie McNamee
Proofreader: Olga Milanko
Typesetter: Tamara Švelić Sabljić
Cover designer: Marija Tudor

ISBN 9781633437333
Printed in the United States of America

To my parents, my biggest cheerleaders
To my three kids, Gu, Jun Wen, and Mao, my pride and joy
—Benjamin Tan Wei Hao

To my wife Priyanka, for your patience, encouragement, and belief in me
—Shanoop Padmanabhan

To my parents, who have always done their utmost to give me the very best
To my wife, Swathi, my greatest teacher in becoming a better person
To my cat, Skye, who always reminds me not to take life too seriously
—Varun Mallya

contents

preface

We've been fortunate to work in machine learning (ML) during one of the most exciting periods in technology. The field is evolving at a breathtaking pace—from breakthrough research to practical applications that touch billions of lives. Being part of this transformation, watching ML systems go from research papers to production services that power real businesses, has been nothing short of remarkable.

The three of us—Benjamin, Shanoop, and Varun—all started our careers as software engineers. We didn't set out to become ML engineers; we stumbled into it. In our respective organizations, we each found ourselves tasked with taking ML models from notebooks to production. We quickly discovered that while our software engineering backgrounds were invaluable, production ML required an entirely new set of skills and practices.

Our first production deployments were humbling experiences. Models that performed beautifully during training struggled in production. Systems broke in unexpected ways. We found ourselves navigating a fragmented landscape of tools, trying to figure out which ones actually worked for real-world problems. Through trial and error, late-night debugging sessions, and learning from our mistakes, we gradually developed an understanding of what it takes to build reliable ML systems.

This journey led us to write this book. We wanted to distill what we've learned and share it with the wider community. The ML tooling ecosystem is vast and fragmented—dozens of options for every component of an ML platform are available. Through experimentation in our respective organizations, we've identified tools and patterns that work well for production systems. This book captures what we've learned.

It's important to note that ML engineering is still a nascent field. Best practices are emerging, not established. You shouldn't treat anything in this book as gospel truth

because the field is evolving too rapidly for that. What works for us might not work for you, and better tools will undoubtedly emerge.

Our goal is to provide patterns and principles that transcend specific tools. When we first conceived this book, ChatGPT would be released just a few months later, transforming the landscape once again. Large language model operations (LLMOps) practices are still being figured out by the community, but we've included two chapters on our experiences building LLM applications. In the same spirit as our Optical Character Recognition (OCR) and movie recommendation projects, we provide practical guidance and open source tooling suggestions based on what has worked for us.

We use real projects throughout the book to demonstrate these concepts in practice. You'll build an OCR system, a movie recommender, and explore LLM applications. These aren't toy examples—they're simplified versions of systems we've built in production, with all the messy details that come with real-world ML engineering. Whether you're a software engineer curious about ML or a data scientist looking to deploy your models, this book will help you navigate the exciting, challenging world of production ML systems. What a time to be alive!

acknowledgments

You would think three authors sharing the load to write a single book means that it gets done three times faster, but we were sorely mistaken! This book wouldn't have been completed without the tireless efforts of our long-suffering development editor, Doug Rudder. His patience, guidance, and commitment to quality helped shape this book into what it is today.

Brian Heath, our technical editor, meticulously went through every snippet of code, keeping us honest and ensuring technical accuracy throughout.

We're deeply grateful to the Manning team who worked behind the scenes on production, editing, and bringing this book to life.

To all the reviewers, your suggestions helped make this a better book: Aleksei Kankov, Aliaksandra Sankova, Amit Singh, Andrew Dunleavy, Andrew R. Freed, Aniket Vashisht, Bikalpa Timilsina, Bin Hu, Charis Kaskiris, Dr. Robert Layton, Fatih Ozer, Frances Buontempo, G Abraham, Giovanni Alzetta, Harcharan S. Kabbay, Harsh Raval, Jaganadh Gopinadhan, Jean-François Morin, Jeremy Bryan, Jerry Kuch, Karrtik Iyer, Kevin H Gould, Kim Falk, Lakshminarayanan A. S, Larry Cai, Louis Luangkesorn, Lucian Mircea Sasu, Manas Talukdar, Maxim Volgin, Mikael Dautrey, Ninoslav Cerkez, Nupur Baghel, Pablo Chacin, Paul Soh, Prithvi Shivashankar, Ravikumar Sanapala, Recep Erol, Sachin Handiekar, Sai Krovvidi, Sandeep Sandhu, Saurabh Aggarwal, Sergio Govoni, Sharath Chandra Parashara, Shivendra Srivastava, Simeon Leyzerzon, Sonam Kanungo, Sri Ram Macharla, Srivathsan Srinivasagopalan, Sumit Pal, Theo Briscoe, Tymoteusz Wołodźko, Vidhya Vinay, Vinicios Wentz, William Jamir Silva, Xin Hu, and Zafar Hussain.

We also want to acknowledge the vibrant open source community. This book wouldn't exist without the countless contributors to Kubernetes, Kubeflow, MLflow,

Feast, and the many other tools we discuss. We often take these tools for granted, but they represent years of collective effort that enables all of us to build better ML systems.

Benjamin: My thanks go to my coauthors, Shanoop and Varun, who bravely embarked on this experiment with me—I'm so glad we made it! I'm also grateful to my kids for understanding all those times I had to sneak away to write, and to my mum, who never failed to ask how the book writing was coming along—your interest and encouragement kept me going.

Shanoop: I am deeply grateful to my colleagues and leadership for their support throughout the development and experimentation phases of implementing the MLOps platform. Your trust and willingness to invest in innovation made all the difference.

Varun: Thank you to Ben and Shanoop, from whom I had the privilege of learning so much. I'm also grateful to all my colleagues, who made it possible to build and rebuild the platform, and who showed great patience when things didn't go as planned. And finally, thank you to my friends and family, whose encouragement inspired me to take on this venture.

about this book

Most machine learning projects never make it to production. The challenge isn't building models—it's deploying them reliably, monitoring their performance, and maintaining them at scale. *Machine Learning Platform Engineering* teaches you how to build the complete infrastructure and workflows needed to operationalize ML systems, from experiment tracking to production deployment.

By the end of this book, you'll have built a complete ML platform from the ground up. You'll containerize and orchestrate ML workloads, automate training pipelines, deploy models as scalable APIs, and implement comprehensive monitoring—all using industry-standard tools such as Docker, Kubernetes, MLflow, and Kubeflow. The final chapters extend these practices to LLMs, showing you how to build and secure production Retrieval-Augmented Generation (RAG) applications.

Who should read this book?

This book is for data scientists and software engineers who want to move beyond Jupyter Notebooks to production ML systems. You should be comfortable with Python and have basic familiarity with ML concepts. No prior experience with Docker, Kubernetes, or machine learning operations (MLOps) tools is required—we'll build everything from scratch. Experienced ML practitioners will benefit from the systematic approach to infrastructure and the modern LLMOps coverage in the final chapters.

What you'll build

You'll construct three complete systems:

- *An ML platform infrastructure* with containerization (Docker), orchestration (Kubernetes), experiment tracking (MLflow), feature stores (Feast), and automated pipelines (Kubeflow)

- *Two traditional ML applications*: an object detection system for ID cards and a movie recommendation engine—covering data preparation, training, validation, deployment, and monitoring
- *An LLM-powered RAG system* called DakkaBot that answers questions about company documentation, including prompt management, semantic testing, safety guardrails, and cost optimization

How this book is organized: A road map

The book has four parts, covering 13 chapters; it also includes two appendices. Part 1 explains what MLOps entails and how to build the infrastructure foundation for ML systems:

- Chapter 1 introduces the ML life cycle, essential MLOps skills, and the foundational components needed to build a production ML platform from scratch.
- Chapter 2 explores the iterative MLOps life cycle, compares MLOps to traditional DevOps, and examines organizational maturity levels in implementing ML operations.
- Chapter 3 covers the infrastructure backbone of ML platforms, including containerization with Docker, orchestration with Kubernetes, continuous integration/continuous deployment (CI/CD) automation, and monitoring with Prometheus and Grafana.

Part 2 focuses on building the core platform capabilities that transform ad hoc ML processes into production-ready systems:

- Chapter 4 demonstrates how to track ML experiments with MLflow, manage models in a model registry, and organize features using the Feast feature store for reproducible ML workflows.
- Chapter 5 teaches pipeline orchestration using Kubeflow Pipelines to automate batch inference workflows, demonstrating how to build reusable components and combine them into production-ready pipelines.
- Chapter 6 shows how to deploy ML models as API endpoints using BentoML and monitor data drift in production using Evidently for both batch and real-time use cases.

Part 3 demonstrates MLOps principles in action through two complete, real-world projects.

- Chapter 7 guides you through data analysis using Kubeflow notebooks and building robust data preparation pipelines for two capstone projects: an ID card detector and a movie recommender system.
- Chapter 8 focuses on designing modular training components, capturing metrics and artifacts, and implementing model validation strategies through practical examples using You Only Look Once (YOLO) object detection.

- Chapter 9 demonstrates how to scale training pipelines by using Kubernetes Persistent Volumes for efficient data management, integrate TensorBoard for training visualization, and use MLflow for comprehensive experiment tracking and model versioning.
- Chapter 10 guides you through deploying ML models as production services using BentoML, covering local development, containerization, observability endpoints, and integration with MLflow for seamless model life cycle management.
- Chapter 11 shows how to implement comprehensive monitoring for ML applications through metrics collection, alerting with Alertmanager, log aggregation with Loki, data drift detection using Deepchecks, and model explainability techniques to understand prediction behavior.

Part 4 extends MLOps to generative AI through LLMOps, building production RAG systems with vector databases, prompt management, and safety controls, and then hardening them for enterprise deployment with testing, guardrails, and cost optimization.

- Chapter 12 introduces LLMOps by building a production RAG system called DakkaBot, covering document ingestion, vector databases, LangChain orchestration, Chainlit UI development, and comprehensive observability with Langfuse.
- Chapter 13 focuses on hardening LLM applications for production through prompt engineering with Langfuse, semantic testing with DeepEval and G-Eval, adversarial security testing with Promptfoo, implementing safety guardrails, and cost optimization strategies for token-based pricing models.

The appendices provide essential setup instructions and reference materials to support the hands-on exercises throughout the book:

- Appendix A walks through the complete installation and setup process for the MLOps platform on your local machine, covering command-line tools (yq, Kustomize, kubectl), Kubernetes distributions (k3s for Linux, microK8s for Mac), deploying Kubeflow using Argo CD, and setting up supporting infrastructure like MLflow, PostgreSQL, MinIO, Redis, BentoML, and Evidently UI.
- Appendix B provides a comprehensive reference guide to YAML syntax and best practices, covering key-value pairs, lists, nested structures, data types, aliases and anchors, block vs. flow styles, and common pitfalls. Since Kubernetes configurations rely heavily on YAML files, this appendix serves as an essential reference for understanding and troubleshooting the manifest files used throughout the book's exercises and deployments.

About the code

This book contains many examples of source code both in numbered listings and in line with normal text. In both cases, source code is formatted in a `fixed-width font like this` to separate it from ordinary text.

In many cases, the original source code has been reformatted; we've added line breaks and reworked indentation to accommodate the available page space in the book. Additionally, comments in the source code have often been removed from the listings when the code is described in the text. Code annotations accompany many of the listings, highlighting important concepts.

You can get executable snippets of code from the liveBook (online) version of this book at https://livebook.manning.com/book/machine-learning-platform-engineering. The complete code for the examples in the book is available for download from the Manning website at www.manning.com and from GitHub at https://github.com/practical-mlops.

liveBook discussion forum

Purchase of *Machine Learning Platform Engineering* includes free access to liveBook, Manning's online reading platform. Using liveBook's exclusive discussion features, you can attach comments to the book globally or to specific sections or paragraphs. It's a snap to make notes for yourself, ask and answer technical questions, and receive help from the authors and other users. To access the forum, go to https://livebook.manning.com/book/machine-learning-platform-engineering/discussion. You can also learn more about Manning's forums and the rules of conduct at https://livebook.manning.com/discussion.

Manning's commitment to our readers is to provide a venue where a meaningful dialogue between individual readers and between readers and the authors can take place. It is not a commitment to any specific amount of participation on the part of the authors, whose contribution to the forum remains voluntary (and unpaid). We suggest you try asking the authors some challenging questions lest their interest stray! The forum and the archives of previous discussions will be accessible from the publisher's website for as long as the book is in print.

about the authors

Benjamin Tan is a principal engineer and product manager for data science at DKatalis where he leads a team of talented machine learning engineers, data scientists, and data engineers. In this capacity, he applies data science and MLOps techniques to enhance and optimize the Bank Jago application, thereby improving the digital experience for millions of Indonesians.

He is also the author of *The Little Elixir and OTP Guidebook* (Manning, 2016), *Building an ML Pipeline with Kubeflow* (Manning liveProject), and *Mastering Ruby Closures* (Pragmatic Publishing, 2017).

Outside of his professional pursuits, Benjamin enjoys quality time with his three kids and indulges in his passion for building scale models.

Shanoop Padmanabhan is a software engineering manager at Continental Automotive, where he leads a team of software engineers focusing on ML-based perception for autonomous vehicles. In this role, he is responsible for designing data-intensive workflows and the associated infrastructure, as well as for enabling ML engineers to deliver and test models in a research and advanced engineering setting.

An avid electronics hobbyist, Shanoop enjoys experimenting and immersing himself in the creative process of building electronic projects.

Varun Mallya is a ML engineer working at Bank Jago (the leading digital bank in Indonesia), where he is responsible for the setup and maintenance of the bank's ML platform. He is responsible for ensuring data science projects move from development to the production environment while working with multiple stakeholders and data scientists in the process. He has also applied ML and MLOps techniques in other industries such as ad tech, electronic manufacturing, and logistics.

Outside of work, he enjoys long nature hikes, traveling, and spending time with his pet cat.