Preface

Everybody's talking about machine learning. It's moved from an academic discipline to one of the most exciting technologies around. From understanding video feeds in self-driving cars to personalizing medications, it's becoming important in every industry. While the model architectures and concepts have received a lot of attention, machine learning has yet to go through the standardization of processes that the software industry experienced in the last two decades. In this book, we'd like to show you how to build a standardized machine learning system that is automated and results in models that are reproducible.

What Are Machine Learning Pipelines?

During the last few years, the developments in the field of machine learning have been astonishing. With the broad availability of graphical processing units (GPUs) and the rise of new deep learning concepts like Transformers such as BERT, or Generative Adversarial Networks (GANs) such as deep convolutional GANs, the number of AI projects has skyrocketed. The number of AI startups is enormous. Organizations are increasingly applying the latest machine learning concepts to all kinds of business problems. In this rush for the most performant machine learning solution, we have observed a few things that have received less attention. We have seen that data scientists and machine learning engineers are lacking good sources of information for concepts and tools to accelerate, reuse, manage, and deploy their developments. What is needed is the standardization of machine learning pipelines.

Machine learning pipelines implement and formalize processes to accelerate, reuse, manage, and deploy machine learning models. Software engineering went through the same changes a decade or so ago with the introduction of continuous integration (CI) and continuous deployment (CD). Back in the day, it was a lengthy process to test and deploy a web app. These days, these processes have been greatly simplified by a few tools and concepts. Previously, the deployment of web apps required collaboration between a DevOps engineer and the software developer. Today, the app can be tested and deployed reliably in a matter of minutes. Data scientists and machine learning engineers can learn a lot about workflows from software engineering. Our intention with this book is to contribute to the standardization of machine learning projects by walking readers through an entire machine learning pipeline, end to end.

From our personal experience, most data science projects that aim to deploy models into production do not have the luxury of a large team. This makes it difficult to build an entire pipeline in-house from scratch. It may mean that machine learning projects turn into one-off efforts where performance degrades after time, the data scientist spends much of their time fixing errors when the underlying data changes, or the model is not used widely. An automated, reproducible pipeline reduces the effort required to deploy a model. The pipeline should include steps that:

- Version your data effectively and kick off a new model training run
- Validate the received data and check against data drift
- Efficiently preprocess data for your model training and validation
- Effectively train your machine learning models
- Track your model training
- Analyze and validate your trained and tuned models
- Deploy the validated model
- Scale the deployed model
- Capture new training data and model performance metrics with feedback loops

This list leaves out one important point: choosing the model architecture. We assume that you already have a good working knowledge of this step. If you are getting started with machine or deep learning, these resources are a great starting point to familiarize yourself with machine learning:

- *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*, 1st edition by Nikhil Buduma and Nicholas Locascio (O’Reilly)

- *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd edition by Aurélien Géron (O'Reilly)

Who Is This Book For?

The primary audience for the book is data scientists and machine learning engineers who want to go beyond training a one-off machine learning model and who want to successfully productize their data science projects. You should be comfortable with basic machine learning concepts and familiar with at least one machine learning framework (e.g., PyTorch, TensorFlow, Keras). The machine learning examples in this book are based on TensorFlow and Keras, but the core concepts can be applied to any framework.

A secondary audience for this book is managers of data science projects, software developers, or DevOps engineers who want to enable their organization to accelerate their data science projects. If you are interested in better understanding automated machine learning life cycles and how they can benefit your organization, the book will introduce a toolchain to achieve exactly that.

Why TensorFlow and TensorFlow Extended?

Throughout this book, all our pipeline examples will use tools from the TensorFlow ecosystem, and in particular TensorFlow Extended (TFX). There are a number of reasons behind our choice of this framework:

- The TensorFlow ecosystem is the most extensively available for machine learning at the time of writing. It includes multiple useful projects and support libraries beyond its core focus, such as TensorFlow Privacy and TensorFlow Probability.
- It is popular and widely used in small and large production setups, and there is an active community of interested users.

- The supported use cases span from academic research to machine learning in production. TFX is tightly integrated with the core TensorFlow platform for production use cases.
- Both TensorFlow and TFX are open source tools, and there are no restrictions on their usage.

However, all the principles we describe in this book are relevant to other tools and frameworks as well.

Overview of the Chapters

In each chapter, we will introduce specific steps for building machine learning pipelines and demonstrate how these work with an example project.

Chapter 1: *Introduction* gives an overview of machine learning pipelines, discusses when you should use them, and describes all the steps that make up a pipeline. We also introduce the example project we will use throughout the book.

Chapter 2: *Introduction to TensorFlow Extended* introduces the TFX ecosystem, explains how tasks communicate with each other, and describes how TFX components work internally. We also take a look at the ML MetadataStore and how it is used in the context of TFX, and how Apache Beam runs the TFX components behind the scenes.

Chapter 3: *Data Ingestion* discusses how to get data into our pipelines in a consistent way and also covers the concept of data versioning.

Chapter 4: *Data Validation* explains how the data that flows into your pipeline can be validated efficiently using TensorFlow Data Validation. This will alert you if new data changes substantially from previous data in a way that may affect your model’s performance.

Chapter 5: *Data Preprocessing* focuses on preprocessing data (the feature engineering) using TensorFlow Transform to convert raw data to features suitable for training a machine learning model.

Chapter 6: *Model Training* discusses how you can train models within machine learning pipelines. We also explain the concept of model tuning.

Chapter 7: *Model Analysis and Validation* introduces useful metrics for understanding your model in production, including those that may allow you to uncover biases in the model's predictions, and discusses methods to explain your model's predictions. "Analysis and Validation in TFX" explains how to control the versioning of your model when a new version improves on a metric. The model in the pipeline can be automatically updated to the new version.

Chapter 8: *Model Deployment with TensorFlow Serving* focuses on how to deploy your machine learning model efficiently. Starting off with a simple Flask implementation, we highlight the limitations of such custom model applications. We will introduce TensorFlow Serving and how to configure your serving instances. We also discuss its batching functionality and guide you through the setup of clients for requesting model predictions.

Chapter 9: *Advanced Model Deployments with TensorFlow Serving* discusses how to optimize your model deployments and how to monitor them. We cover strategies for optimizing your TensorFlow models to increase your performance. We also guide you through a basic deployment setup with Kubernetes.

Chapter 10: *Advanced TensorFlow Extended* introduces the concept of custom components for your machine learning pipelines so that you aren't limited by the standard components in TFX. Whether you want to add extra data ingestion steps or convert your exported models to TensorFlow Lite (TFLite), we will guide you through the necessary steps for creating such components.

Chapter 11: *Pipelines Part I: Apache Beam and Apache Airflow* connects all the dots from the previous chapters. We discuss how you can turn your components into pipelines and how you'll need to configure them for the orchestration platform of your choice. We also guide you through an entire end-to-end pipeline running on Apache Beam and Apache Airflow.

Chapter 12: *Pipelines Part 2: Kubeflow Pipelines* continues from the previous chapter and walks through end-to-end pipelines using Kubeflow Pipelines and Google's AI Platform.

Chapter 13: *Feedback Loops* discusses how to turn your model pipeline into a cycle that can be improved by feedback from users of the final product. We'll discuss what type of data to capture to improve the model for future versions and how to feed data back into the pipeline.

Chapter 14: *Data Privacy for Machine Learning* introduces the rapidly growing field of privacy-preserving machine learning and discusses three important methods for this: differential privacy, federated learning, and encrypted machine learning.

Chapter 15: *The Future of Pipelines and Next Steps* provides an outlook of technologies that will have an impact on future machine learning pipelines and how we will think about machine learning engineering in the years to come.

Appendix A: *Introduction to Infrastructure for Machine Learning* gives a brief introduction to Docker and Kubernetes.

Appendix B: *Setting Up a Kubernetes Cluster on Google Cloud* has some supplementary material on setting up Kubernetes on Google Cloud.

Appendix C: *Tips for Operating Kubeflow Pipelines* has some useful tips for operating your Kubeflow Pipelines setup, including an overview of the TFX command-line interface.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`

Shows commands or other text that should be typed literally by the user.

`Constant width italic`

Shows text that should be replaced with user-supplied values or by values determined by context.

TIP

This element signifies a tip or suggestion.

NOTE

This element signifies a general note.

WARNING

This element indicates a warning or caution.

Using Code Examples

Supplemental material (code examples, etc.) is available for download at *https://oreil.ly/bmlp-git*.

If you have a technical question or a problem using the code examples, please email *bookquestions@oreilly.com* and *buildingmlpipelines@gmail.com*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Building Machine Learning Pipelines* by Hannes Hapke and Catherine Nelson (O'Reilly). Copyright 2020 Hannes Hapke and Catherine Nelson, 978-1-492-05319-4."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

O'Reilly Online Learning

NOTE

For more than 40 years, *O'Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, visit *http://oreilly.com*.

How to Contact Us

Both authors would like to thank you for picking up this book and giving it your attention. If you would like to get in touch with them, you can contact them via their website *www.buildingmlpipelines.com* or via email at *buildingmlpipelines@gmail.com*. They wish you every success in building your own machine learning pipelines!

Please address comments and questions concerning this book to the publisher:

- O’Reilly Media, Inc.
- 1005 Gravenstein Highway North
- Sebastopol, CA 95472
- 800-998-9938 (in the United States or Canada)
- 707-829-0515 (international or local)
- 707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *https://oreil.ly/build-ml-pipelines*.

Email *bookquestions@oreilly.com* to comment or ask technical questions about this book.

For news and information about our books and courses, visit *http://oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

Acknowledgments

We’ve had so much support from many wonderful people throughout the process of writing this book. Thank you so much to everyone who helped make it a reality! We would like to give an especially big thank you to the following people.

Everyone at O’Reilly has been fantastic to work with throughout the whole life cycle of this book. To our editors, Melissa Potter, Nicole Taché, and Amelia Blevins, thank you for your amazing support, constant encouragement, and thoughtful feedback. Thank you also to Katie Tozer and Jonathan Hassell for their support along the way.

Thank you to Aurélien Géron, Robert Crowe, Margaret Maynard-Reid, Sergii Khomenko, and Vikram Tiwari, who reviewed the entire book and provided many helpful suggestions and insightful comments. Your reviews have made the final draft a better book. Thank you for your hours reviewing the book in such detail.

Thank you to Yann Dupis, Jason Mancuso, and Morten Dahl for your thorough and in-depth review of the machine learning privacy chapter.

We have had fantastic support from many wonderful people at Google. Thank you for helping us find and fix bugs, and thank you for making these tools available as open-source packages! As well as the Googlers mentioned, thanks especially to Amy Unruh, Anusha Ramesh, Christina Greer, Clemens Mewald, David Zats, Edd Wilder-James, Irene Giannoumis, Jarek Wilkiewicz, Jiayi Zhao, Jiri Simsa, Konstantinos Katsiapis, Lak Lakshmanan, Mike Dreves, Paige Bailey, Pedram Pejman, Sara Robinson, Soonson Kwon, Thea Lamkin, Tris Warkentin, Varshaa Naganathan, Zhitao Li, and Zohar Yahav.

Thanks go out to the TensorFlow and Google Developer Expert community and its amazing members. We owe deep gratitude to the community. Thank you for supporting this endeavor.

Thank you to the other contributors who helped in the various stages: Barbara Fusinska, Hamel Husain, Michał Jastrzębski, and Ian Hensel.

Thank you to the people at Concur Labs (past and present) and elsewhere in SAP Concur for great discussions and helpful ideas. In particular, thank you to John Dietz and Richard Puckett for your amazing support of the book.

Hannes

> I would like to thank my wonderful partner, Whitney, for her tremendous support throughout the course of writing this book. Thank you for your constant encouragement and feedback, as well as putting up with me spending long hours writing. Thank you to my family, especially my parents, who let me follow my dreams throughout the world.
>
> This book would not have been possible without amazing friends. Thank you, Cole Howard for being a wonderful friend and teacher. Our collaboration back in the day kick-started this publication and my thinking about machine learning pipelines. To my friends, Timo Metzger and Amanda Wright: thank you for teaching me about the power of language. And thank you to Eva and Kilian Rambach as well as Deb and David Hackleman. Without your help, I wouldn't have made it all the way to Oregon.
>
> I would like to thank my previous employers like Cambia Health, Caravel, and Talentpair for letting me implement concepts of this publication in production settings, even though the concepts were novel.
>
> This publication would not have been possible without my coauthor, Catherine. Thank you for your friendship, your encouragement, and your endless patience. I am glad we met due to sheer randomness in life. I am very happy we accomplished this publication together.

Catherine

> I've written a lot of words in this book, but there aren't words to express how much I appreciate the support I've had from my husband, Mike. Thank you for all your encouragement, cooking, help-

ful discussions, sarcasm, and insightful feedback. Thank you to my parents for planting the seed of programming so long ago—it took a while to grow, but you were right all along!

Thank you to all the wonderful communities I have been fortunate to be a part of. I've met so many great people through Seattle PyLadies, Women in Data Science, and the wider Python community. I really appreciate your encouragement.

And thank you to Hannes for inviting me on this journey! It wouldn't have happened without you! Your depth of knowledge, attention to detail, and persistence have made this whole project a success. And it's been a lot of fun, too!