Chapter 2
A Software Engineering Perspective on Building Production-Ready Machine Learning Systems

Petra Heck
Fontys University of Applied Sciences, The Netherlands

Gerard Schouten
Fontys University of Applied Sciences, The Netherlands

Luís Cruz
Delft University of Technology, The Netherlands

ABSTRACT

This chapter discusses how to build production-ready machine learning systems. There are several challenges involved in accomplishing this, each with its specific solutions regarding practices and tool support. The chapter presents those solutions and introduces MLOps (machine learning operations, also called machine learning engineering) as an overarching and integrated approach in which data engineers, data scientists, software engineers, and operations engineers integrate their activities to implement validated machine learning applications managed from initial idea to daily operation in a production environment. This approach combines agile software engineering processes with the machine learning-specific workflow. Following the principles of MLOps is paramount in building high-quality production-ready machine learning systems. The current state of MLOps is discussed in terms of best practices and tool support. The chapter ends by describing future developments that are bound to improve and extend the tool support for implementing an MLOps approach.

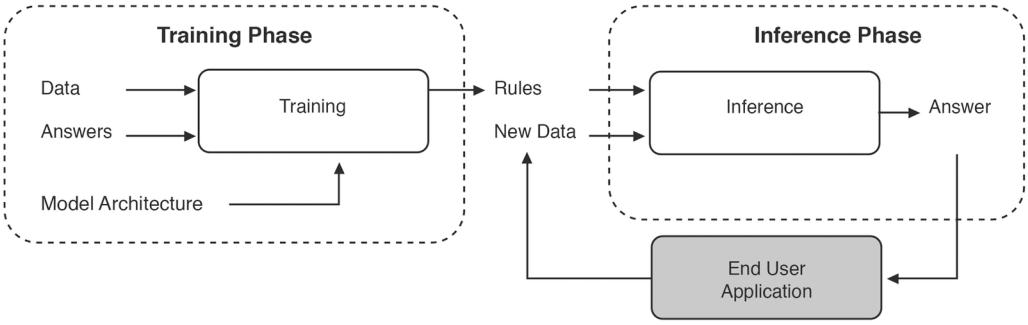
DOI: 10.4018/978-1-7998-6985-6.ch002

INTRODUCTION

The application of data science and artificial intelligence (AI) in business and industry is not a niche anymore. AI is used in traditional areas like machine vision, speech recognition, and translation, and in a wide variety of novel areas like detecting fraud in transaction data, decoding and identifying hand-written text, medical diagnosis, or even tracking wildlife. Machine learning (ML) is seen as a subset of AI. The term *machine learning* denotes a set of algorithms that learn from data. Machine learning (ML) also includes Deep Learning (DL), which denotes a set of algorithms that use multi-layered neural networks to learn from data. Currently most AI implementations are ML implementations. In order to build solutions that can be delivered to customers, ML should be connected to software. The software solution ensures that what the ML model learns from the data is transformed into meaningful predictions or decisions for the end-user. The need for such software solutions is growing fast as the number of AI applications increases. This chapter discusses software solutions that contain an ML component and focuses on the engineering approach required to construct them. The remainder of this chapter refers to this type of software solutions as "ML systems", to indicate that they consist of both an ML model and a software solution or software system.

Figure 1 illustrates the relevant concepts of an ML system. This chapter focuses on supervised ML, where a model needs to be trained on historical data, labeled with answers. After a model has been trained successfully, it must be deployed somewhere such that a software solution can feed it new data and retrieve answers for this new data. This is called inference in ML terminology.

Figure 1. Incorporating a (supervised) ML model into a software application (adapted from Cai et al., 2020)



As the application of ML in business and industry matures, more and more organizations reach the point where they need to run an ML software solution in their production environments. Yet, the data scientists designing the models have not been trained in the software engineering skills required to put their models into production. On the other hand, the addition of ML components introduces some new challenges for software developers building the solution.

This chapter discusses those challenges and provides the solutions (in terms of good engineering practices and supporting tools) available to date. The chapter starts by providing some background on