Product Engineering for Machine Learning: A Grey Literature Review

Isaque Alves
Faculty of Gama (FGA)
University of Brasília (UnB)
Brasília, Brazil
isaquealvesdl@gmail.com

Leonardo Leite, Paulo Meirelles
Free and Open Source Software Competence Center
Institute of Mathematics and Statistics (IME)
University of São Paulo (USP)
São Paulo, Brazil
{leofl,paulormm}@ime.usp.br

Carla Rocha
Faculty of Gama (FGA)
University of Brasília (UnB)
Brasília, Brazil
caguiar@unb.br

***Abstract*—Our research aims to identify the existing product-engineering methods and practices adopted in the industry for building applications and platforms relying on machine learning. We conducted a Grey Literature Review (GLR) to investigate and discuss the methods and practices applied to the ML product lifecycle from the industry perspective. We mapped 58 practices and methods in 6 categories related to the processes of designing, developing, testing, and deploying ML products. It is crucial to guide product managers, data scientists, and software engineers to better understanding the challenges of ML product lifecycle.**

***Index Terms*—Machine Learning, Product Engineering, Machine Learning Systems, Grey Literature Review.**

I. Introduction

The lifecycle of a Machine Learning (ML) product is different from the traditional software lifecycle, thus techniques and practices must be adapted to suit the particularities that involve the relationship between the data, the trained model, and the source code. Due to the novelty, the velocity that organizations are adopting ML products, and the longer publication process of peer-reviewed academic literature, publications not indexed by scientific repositories give a vast amount of up-to-date and emerging information regarding the theme. Our research aims to identify the existing methods and practices adopted in the industry for building applications and platforms relying on machine learning. To grasp the industry and practitioners' perspective, we conducted a Grey Literature Review (GLR) to identify these methods and practices applied to the ML product lifecycle.

By applying our search strings on the Medium and Toward Data Science blogs, we found 80 posts. The first selection round, using the quality criteria, reduced this set to 64 posts. Finally, the second selection round focused on verifying the essays' scope and led us to 41 selected articles.

II. Results and Implications

We mapped 58 practices and methods related to the ML product development process (e.g., *verify how necessary is ML for the product*), from the practitioners' viewpoint. We grouped them into the following categories: *Problem definition and solution design*; *Product management*; *Data management*; *Model management*; *Software management*; *Delivery and runtime*.

The list of practices and the frequency they are mentioned is available in our supplementary material at https://github.com/alvesisaque/PE_for_ML and it suggests the importance and challenges of such practices throughout the ML products' lifecycle. While most of the *data management* category practices have been cited in multiple blog posts, we have **agile practices** with only three blog post citations. It implies that adopting agile practices to ML workflows is not an issue for practitioners. On the other hand, the results suggest that practitioners are currently discussing challenges regarding data maintenance and quality practices. The most cited were: **data strategy**; **data collection and evolution**; **ensure the reliability and availability of data**; and **data cleaning and labeling**. The focus on the *data management* category is extremely relevant, indicating that the crucial concerns for engineers shifted from source code to data.

Implication #1: *ML product teams require more skilled software engineers. Versioning data and data schemes, cleaning, reusing, labeling, and configuring automated pipelines are examples of software engineers' assignments in a project with ML modules.*

The *Product management* category, with 17 methods and practices, presents more practices and methods than other categories. It suggests practitioners face challenges in adapting traditional software engineering practices and workflows to ML product development. Managing the ML product workflow requires defining potentially new team roles, adjusting agile practices to incorporate ML design's experimental nature, and incorporating machine learning workflows, tools, and environments. **Feedback loop** practice illustrates how this is an emerging research topic. The term **feedback loop** is a technical debt of ML systems when the model may directly influence the selection of its future training data or indirectly influences the training data of another model. However, in our coding, **feedback loop** appeared as an agile practice intensified over the development of an ML system since the process is more experimental than for traditional software.

Implication #2: *ML product management requires more skilled software managers and engineers: while managers should adapt the process to ML product development workflow, engineers must revisit data, features, and models often.*