Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

A Survey of Product Engineering Applied on Machine Learning Projects

Autor: Isaque Alves de Lima
Orientador: Dra. Carla Rocha

Brasília, DF
2021

Isaque Alves de Lima

A Survey of Product Engineering Applied on Machine Learning Projects

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dra. Carla Rocha

Brasília, DF

2021

Agradecimentos

Agradeço ao esforço das minha família: Aleciene, Izaias, Rute e Davi que mesmo de longe sempre me apoiaram durante essa jornada.

Agradeço aos meus pais postiços Ana França e Rogério Egewarth que sempre me apoiaram e nunca me deixaram desistir, me dando suporte e carinho. Agradeço também aos meus irmãos de coração João Henrique e Eliseu Egewarth, por trazer felicidade, amizade e companheirismo durante essa minha jornada. Sem eles esse sonho não teria sido realizado.

Agradeço à professora Dra. Carla Rocha, pelos ensinamentos e por acreditar e confiar no meu trabalho. À Leila Rosa, por ler e revisar todos os meus textos e todos os professores que contribuíram para a minha formação profissional e pessoal como cidadão brasileiro.

Resumo

O Aprendizado de Máquina (AM) é um subcampo de computação que evoluiu a partir de estudos de reconhecimento de padrões e da teoria de aprendizado computacional. O ML é mais usado para melhorar a usabilidade, reduzir tarefas de automação de trabalho mecânico e resolver problemas humanos complicados. Além de exigir uma equipe multidisciplinar e não ter práticas e estudos definidos, o ML ainda é extremamente dependente da quantidade e da qualidade dos dados. Devido aos desafios de desenvolver produtos de ML, muitas empresas, sejam grandes ou pequenas, encontram dificuldades e até falham durante o desenvolvimento de produtos de AM.

Este artigo apresenta uma investigação e discussão sobre o processo, desafios e particularidades para desenvolver um produto de AM, a partir de uma perspectiva acadêmica e de mercado. Apresentamos uma Revisão da Literatura Multivocal (MLR) e, usando a Teoria Fundamentada nos Dados (GT), criamos uma estrutura conceitual correlacionando os conceitos, práticas e ferramentas usadas durante o ciclo de vida do produto de AM. Em seguida, discutimos os resultados com base em nossas perguntas de pesquisa e apresentamos métodos e práticas relacionados à definição do problema e design da solução, gerenciamento de produto, dados e pipeline de modelo e entrega do produto. Também exploramos os desafios e vantagens de construir esse tipo de produto.

Palavras-chaves: Aprendizado de Máquina. Engenharia de Produto. Desafios. Práticas. Revisão Multivocal da Literatura.

Abstract

Machine Learning (ML) is a computation subfield that evolved from a pattern recognition study and computational learning theory. ML is more used to improve usability, reduce mechanical work automating tasks, and solve complicated human problems. In addition to requiring a multidisciplinary team and not having defined practices and studies, ML is still utterly dependent on the quantity and quality of the data. Due to the challenges of developing ML products, many companies, whether large or small, encounter difficulties and even fail during ML products' development.

This paper presents an investigation and discussion about the process, challenges, and particularities to develop an ML product, based on an academic and market perspective. We report a Multivocal Literature Review (MLR) and, using Grounded Theory (GT), to create a conceptual framework correlating the concepts, practices, and tools used during the ML product life cycle. Then, we discuss the results based on our research questions and present methods and practices related to the problem definition and solution design, product management, data and model pipeline, and product delivery. We also explore the challenges and advantages of constructing this kind of product.

Key-words: Machine Learning. Product Engineering. Challenges. Practices. Tools. Multivocal Literacture Review.

Sumário

1 Introduction

It all started when artificial intelligence pioneer Arthur Samuel, an engineer at MIT, coined the term "Machine learning"and affirmed that "The field of study that gives computers the ability to learn without being explicitly programmed"(Samuel, 1959). Since then, ML's studies and use have been increasing due to its capability to improve usability, recognize images, diagnose cancer, and estimate insurance, which is difficult to achieve with standard software (LIBBRECHT; NOBLE, 2015) (KOUROU et al., 2015).

Unlike traditional software, machine learning's products are a data-centered product, and it does not have engineers and developers establishing specific functions for particular solutions or needs. However, it can generate rules based on the patterns identified in the analyzed data, aiming to improve the performance of a task or, depending on the application, make the most appropriate decision for the context (REGALADO, 2018). Nevertheless, developing a product or resource with this technology requires more attention due to its particularities and challenges (AMERSHI et al., 2019a). ML is completely dependent on the quality and quantity of the data (SINGLA; BOSE; NAIK, 2018) and the process of cleaning, labeling, and train the model demand time and resources (SCHELTER et al., 2018).

In the last few years, machine learning is gaining more space in the market, reflecting the effectiveness and the profit companies have with these systems. According to Forbes magazine, the global machine learning market is projected to grow from \$7.3B in 2020 to \$30.6B in 2024 (COLUMBUS, 2020). The interest in ML is evident but is still new and has many gaps and points to be explored and studied. In contrast, according to The Machine magazine, 87% of the ML projects are not delivered (WHY. . . , 2019). One of the main factors is the lack of knowledge about this type of product development process. Compared to "common"software development, there are few studies in best practices, means, and particularities of design, develop and maintain this specific product. Nevertheless, as it is a new product on the market, the evolution, practices, and tools are still in the process of maturing in the industry (AMERSHI et al., 2019b).

In this context, our research problem is to develop a conceptual framework to identify and connect the terms used by industry and academia, to guide academics, product managers, and engineers in the exploration of ML product life cycle. Which, due to the lack of guides and content, still have many flaws. We use the Multivocal Literary Review (MLR) to understand the challenges and advantages of using machine learning products. We also look for the practices, methods, tools, and artifacts used during the life cycle. We select the MLR because we have not identified enough academic studies on this process. Therefore, we include gray literature in addition to published formal literature. We describe the process and protocol in Section 3. While Section 4 describe our source of knowledge, analyze and describe the selected studies.

Based on the reviewed literature, Section 5 presents and discusses the results obtained through the MLR organized by this work's research questions. We present the results in tables and comment on the main challenges, advantages, practices, and essential methods for the development of an ML product. We also present the artifacts and tools already found during the research. In Section **??**, we present the objectives of the project's evolution and planning the conceptual framework development.

2 Background

In the software engineering ecosystem, ensuring product quality is fundamental and requires efforts and costs that many organizations end up not giving the necessary attention. The production of quality software (FUTRELL; SHAFER; SHAFER, 2001) is essential to ensure users' acceptance and usability, but the process is not so simple. The guarantee of this quality must be taken into account in the process of understanding the problem and defining the scope, with meetings and artifacts (DITTRICH, 2014). In the development process, the execution and validation of tests guarantee the quality of the solution being implemented. Throughout this process, making continuous validations also helps meet expectations, quality, and deliver a quality product.

2.1 Product Engineering

Product Engineering is the process of innovating, designing, developing, testing, and deploying a software product (LINDEN; SCHMID; ROMMES, 2007). Unlike standard software engineering, when it comes to product engineering, other responsibilities must be added. Software engineering, for example, handles the process of eliciting, developing, maintaining, and deploying the product very well (ROSS; GOODENOUGH; IRVINE, 1975) (FICHMAN; KEMERER, 1993). However, product engineering involve other activities before this process begins, such as design and product ideation (NORI; SWAMINATHAN, 2006).

Currently, software engineering seeks to achieve product qualities indirectly through standards and process improvements. For example, the Capability Maturity Model (CMM) is an essential reference for quality software development (PAULK, 2002) (HERBSLEB et al., 1997). It has the Key Process "Software Product Engineering" focused on guaranteeing the quality of software products. Nevertheless, this model is still very much focused on the software process, aiming at software consistency, performance, and integrity, which does not guarantee the quality and properties of a product (NORI; SWAMINATHAN, 2006).

Currently, the market is still based and seeks to guarantee the quality of the product, focusing on the maturity of the process and the code's technical quality (HUMPHREY, 1988). Large or small companies widely use software engineering practices such as Agile methods and DevOps concepts and practices (LWAKATARE et al., 2019). These practices are well consolidated and evolved and guarantee the quality of the software and the development cycle, but they are not enough to guarantee the quality of a product and a delivery that meets the customer's needs or user. One of the significant criticisms of agile methods is that even with the improvement in product development quality, the success rate has not increased considerably compared to traditional models (ROSATO, 2018). Therefore, it is necessary to investigate more about the process of study and problem definition, solution design, and validation of this solution and the means used to guarantee the technical quality of the product.

The engineering process of a product is conceptually simple. It involves thinking about a problem/opportunity; designing the solution; building a solution; delivering to the customer; and checking if it meets their expectations (ANON; VILLAUMBROSIA, 2017) (ULRICH, 2003), as presented in Figure 1.

Figura 1 – The five phases of the product engineering workflow. The development consists of a cycle that allows feedback loops at any stage. In Phase 3, the presented feedback loop arrow aims to ensure greater alignment between the Build and Design the Solution.

However, the challenge is in the details. For example, in a business environment, the Phase 1 consists of understanding the organization strategies, the market niche involved, the type of product already developed, customers, and the target audience. With the environment in mind, the next step is to identify opportunities and search for data that can consolidate the need for product development. Throughout the process, collecting information from customers/users is crucial to analyzing existing needs and problems before evolving with the validation process to ensure that the opportunity is strong enough and meets the established data and metrics (ANON; VILLAUMBROSIA, 2017).

Another significant point in the first phase is establishing the scope, which consists of defining and specifying the customers and requirements for the solution (REBITZER et al., 2004) (KHAN, 2006). When conceiving a product in a more agile environment, it is common to hear the term Minimum Product Viable (MVP). A term from the Lean methodology, which means "What is the minimum that can be produced, that addresses customers' needs and can validate the opportunity "(LENARDUZZI; TAIBI, 2016). That is the most adopted approach because if we know that customers want some feature, implementing it from the start can reduce iteration cycles and consequently faster product delivery (MOOGK, 2012).

The Phase 2: Design the Solution, consists of seeking a viable solution to the proposed problem or challenge. That contrary to belief, the design does not just mean the appearance of the solution. In product engineering design involves aspects such as information architecture, definitions, and creative discussions of the best ways and techniques to develop a product and how the information will be used and presented and used by users (NORI; SWAMINATHAN, 2006). In this step, the widely used technique is creating prototypes to validate and collect feedback from stakeholders. It helps to validate the viable solutions with users. It is also essential to think about the product's quality and what metrics will be used. Align the product's objectives and requirements, and evaluate the product quantitatively and qualitatively to see if the objectives are being achieved.

The next phase (Phase 3) consists of developing the solution, where there are different approaches used, which depends on the project, the scope, and those involved. This phase consists of presenting technical solutions to solve the problem. Although Figure 1 presents a linear process, a product's life cycle is very iterative, and all the steps are continually revised, mainly between the design and the solution creation phase. Even during the design phase of the solution prototyping, specifying features, and requirements during development, there may be cases that were not discussed during the previous phase, so it is necessary to maintain active communication and integrated work to meet the needs and issues that arise during the product development process. Still, it is crucial to share prototypes and beta versions of the product with customers or people within the company to get an early evaluation of the product during this phase (ANON; VILLAUMBROSIA, 2017).

The product delivery (Phase 4) is also a significant one, as this will possibly be the customer's first impression. That is why companies usually invest in marketing, to ensure good communication and iteration of customers with the project. During this phase, communication is discovered and put into practice as the customer, collecting formal and informal feedback (ANON; VILLAUMBROSIA, 2017).

The Phase 5 of the product development life cycle consists of evaluating the development and iterations of the cycle, monitoring the metrics, and defining what should be done in the next steps to improve the project's quality. As the development of a product is a cycle, this phase specifies the objectives and improvements to be applied in the new phase. However, now that the product is being used, it is possible to follow the established metrics in the solution design phase and make decisions and actions based on data driven so that the product meets the real needs and expectations of customers (CLEMENTS,).

2.2 Product Engineering for Machine Learning

While traditional software applications are deterministic, machine learning models are probabilistic (AKKIRAJU et al., 2020). In other words, developing a machine learning product involves more uncertainties and risks throughout the cycle. Because the life cycle of an ML product is significantly different from the life cycle of standard software, techniques and practices must be adapted to suit the particularities that involve the relationship with the data (SINGLA; BOSE; NAIK, 2018).

The cycle shown in Figure 1 is also applied in ML products, but with a different approach. Before creating or deploying a machine learning feature, one must consider some crucial points for the creation and specification of the product (SHAMS, 2018). Developing an ML product requires a more complex process and more hardware processing (ZHANG; TSAI, 2003; HAZELWOOD et al., 2018). Therefore, the first phase is crucial for the development, and it is necessary to consider some factors about the client's needs and finally verify the need to use ML to solve this problem. The use of an algorithm or the power of ML occurs when the rules are not precise or are not known, but we have many data (AMERSHI et al., 2019a; AKKIRAJU et al., 2020).

Consequently, it is necessary a big problem to justify this product, with many unknown rules (so many that it is difficult to implement using standard software) (AMERSHI et al., 2019a). In this phase, two questions must be answered: (1) What is the problem to be solved, and what are the expected responses?; (2) Do we have access to many data?. If the previous two questions are answered quickly, developing a ML product is the best solution.

Still, in the first phase, four other important factors were taken into account in the conception and design of this product, they are: Very complex logic; Rapid scalability; Requires specialized customization; and Adaptation in real-time. If the product falls into any of these categories, the need to develop an ML product becomes more evident (ALPAYDIN, 2020).

In the design phase of the solution, it is essential to check the role of ML in the product, whether it will be the core of the application, or just a feature to be implemented, or whether it is a black box (e.g., users decide which data to enter) or open (HARRINGTON, 2012). In this phase, the objectives and metrics are also specified to assess whether the product has achieved the objectives and whether the objectives are aligned with those interested in the project.

The next phase in the product's life cycle is development, which consists of developing the product specified and designed in the previous phase. However, it is divided into two stages for an ML product: the data pipeline and model pipeline. Where engineers collect, clean, and organize data to prevent the model from learning from "dirty"data or having prejudices and information that, in addition to not being essential for the product, can hinder the model's learning. This phase requires care and can cost time and resources to guarantee the product's quality (e.g., data annotation) (SHAMS, 2018). The second stage consists of the model's construction and training, using techniques such as feature engineering (AMERSHI et al., 2019a), or reuse of code and features (AKKIRAJU et al., 2020). As in the standard cycle of product development, this phase is also closely linked to the solution's design, as during training and data analysis, new challenges or information may arise that add value to the product, and this must be considered together.

The delivery and monitoring of the ML product to users can also be a challenge, the data must be consistent with what was reported in the training phase, and new data collections should be carried out to maintain the correct functioning of the product (AMERSHI et al., 2019a). Another critical point is that this product requires more robust processing to ensure agility in a prediction or continuous learning model. Track user feedback, continuous monitoring to check if the model continues to have accuracy, performance and if meeting the metrics stipulated in the solution design phase is essential to define and improve the product in the next cycles of development and improvement (ANON; VILLAUMBROSIA, 2017) (AMERSHI et al., 2019a) (SINGLA; BOSE; NAIK, 2018).

3 Study Design

The recent growth of the use of ML in software products has not been accompanied by research and academic publications on this topic. However, on the other hand, it is already possible to find articles on blogs with reliable information on the techniques and practices adopted to create this type of product.

As shown in Figure 3, the results collected on academic basis were not significant for conducting this research, so we expand our research to gathering information on the leading blogs, thus expanding our source of information.

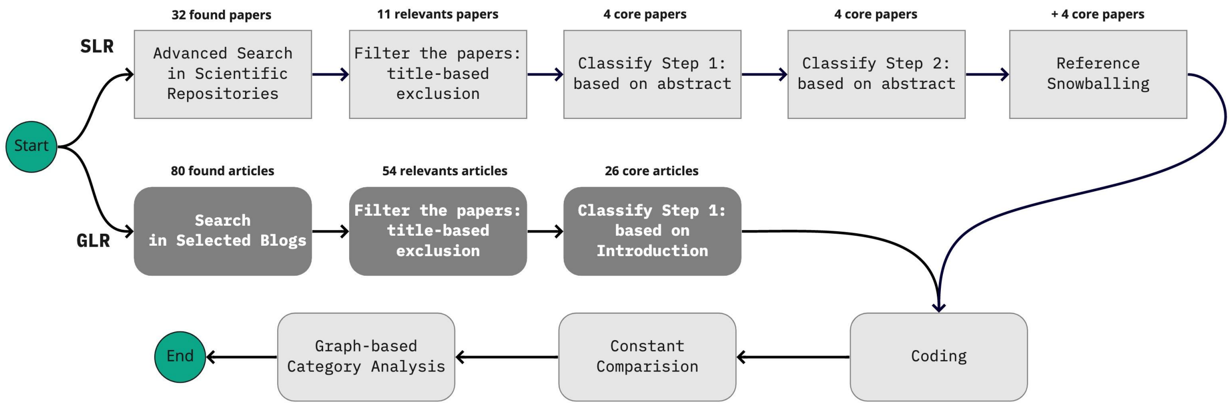

Figura 2 – Sequence of steps taken in our study.

Therefore, this survey adopted a Multivocal Literature Review (MLR), a form of systematic literature review that includes inputs from academic peer-reviewed papers and sources from the Grey Literature (GAROUSI; FELDERER; MÄNTYLÄ, 2019), to understand the ML product from academics and the industry perspective. In this section, we describe our procedures and protocols, outlied in Figure 2, inspired by (WOHLIN, 2014a) to limit analyzed studies and based on Ground Theory and Coding (CHARMAZ, 2008) we use as a way to collect and understand the study's empirical events and experiences to create our framework outcome. We define our gathering data protocol in two ways. First, we select the data of Formal Literature (FL) with advanced search in scientific papers repositories, inspired by the procedures of Systematic Literature Reviews (SLR) (BUDGEN; BRERETON, 2006) (KITCHENHAM et al., 2009) (BRERETON et al., 2007), and we research in the Gray Literature (GL) selecting the main bases and analyzing of the publications to identify those that do not have a market bias.

To guide our data analysis, we define the following research questions:

- (1) What are the challenges and benefits of developing ML products?

Tabela 1 – PICOC: Method used to describe the research questions.

Population	Product engineering techniques applied to ML products.
Intervention	ML product management.
Comparison	Software product
Outcomes	Processes, practices and tools used during the product life cycle.
Context	ML Product lifecycle management.

- (2) What are the practices and methods adopted in product development?
- (3) What are the artifacts and tools adopted in the life cycle of ideation, design, and construction of the ML product?

To refine and structure the objective, we use the PICOC (Population, Intervention, Comparison, Output, Context), a method proposed by Petticrew (PETTICREW; ROBERTS, 2008). Table 1 detailed our scenario.

To reduce the classification process's subjectivity, we establish the following rules to classify Formal and Gray literature, and all authors need to agree to maintain the same focus. We classify the works found in three categories by assigning stars to them. Works that are less relevant or do not answer research questions in any way tend to receive one star and be discarded. The publications that are completely aligned with this work's research objectives tend to receive two stars and are considered core papers.

- If the paper focus on developing an ML product, it tends to be a core paper and receive two stars;
- If the paper discusses peculiarities and challenges for developing or managing an ML product, it tends to be two stars;
- If the paper talks about using some product engineering practices or correlate this with ML products, it tends to be two stars.
- If the paper discusses the ML product life cycle process or workflow, it tends to be a core paper;
- If the paper discusses: Developing ML products, challenges in managing ML products, or something connected with developing/validating ML products, it tends to be two stars;
- If the paper talks about any topic (e.g., big data) and uses ML products only for the background contextualization, it tends to be 1 star;

- If the paper only talks about applying ML in a specific context without elaboration beyond the state-of-the-art, it tends to be 1 star;
- Only a proposal, without validation, tends to be 1 star;
- If the paper talks about a tool that may be important for ML engineering (e.g., applying some Python library), but the paper is too centered on this single tool and does not discuss ML product itself, it tends to be one star.

To help with the collection process, we use Mendeley to maintain the papers and Google Sheets to select and classify the articles. We present the core chosen studies at the end of this article. We define a COD to help identify how we found the paper: **ACM** "ACM Digital Library"; **IEEE** "IEEE Explorer"; **SC** "Scopus"; **SL** "Springer Link"; **M** "Medium"; **TDS** "Toward Data Science"; and **SB** for "Snowballing". For the coding process, we define colors to highlight the articles' keywords and more easily identify important information related to our research questions. For the first research question, we define red for challenges and purple for advantages. For the second research question, we define blue to identify the methods and practices used. Finally, to identify the artifacts and tools, which correspond to the third question, we use the color green.

3.1 Formal Literature Protocol

To construct our query string, we define the following keywords from the questions and the PICOC model: *Machine Learning Product*, *Product Engineering*, *Product Management*, *Practices*, *Process*, and *Tools*.

After defining the keywords, we set up the following string by inserting synonyms. This research focuses on *Machine Learning*, and we know that ML is only a type of *Artificial Intelligence*, but it was necessary to insert these term because some articles refer to Artificial Intelligence as a synonym for ML. We note some papers refer to *Product Engineering*, *Product Management*, and *Product Development* as synonymous. Therefore we adjust our query string to accept all these terms.

Some studies have different ways of using the terms: *"build*"*, *"manage*"*, and *"develop*"* as building, management, managing, manage product, and development or developing, so we add these terms in this format to achieve the most significant number possible of information related to building, management, and development for these products.

In Table 2, it is possible to find the search string adapted to each scientific base chosen for the research they are (1) ACM digital library, (2) IEEE Xplore, (3) Scopus, and (4) SpringerLink.

Tabela 2 – Search strings used in scientific bases and the results.

Base	**String**	**Results**
ACM digital library	[[All: "ml product"] OR [All: "machine learning product"] OR [All: "ai product"] OR [All: "artificial intelligence product"]] AND [[All: "product engineering"] OR [All: "product management"] OR [All: "product development"]] AND [[All: build*] OR [All: manage*] OR [All: develop*]] AND [[All: practices] OR [All: methods] OR [All: tools]] AND [[All: workflow] OR [All: process*] OR [All: lifecycle]]	7
IEEE Xplore	(("All Metadata":"ML Product" OR "All Metadata":"Machine Learning Product" OR "All Metadata":"AI Product" OR "All Metadata":"Artificial Intelligence Product") AND ("All Metadata":"product engineering" OR "All Metadata":"product management" OR "All Metadata":"product development") AND ("All Metadata":build* OR "All Metadata":manage* OR "All Metadata":develop*) AND ("All Metadata":practices OR "All Metadata":methods OR "All Metadata":tools) AND ("All Metadata":workflow OR "All Metadata":process* OR "All Metadata":lifecycle))	18
Scopus	(TITLE-ABS-KEY(("ML Product" OR "Machine Learning Product"OR "AI Product" OR "Artificial Intelligence Product")) AND TITLE-ABS-KEY(("product engineering" OR "product management"OR "product development")) AND TITLE-ABS-KEY((build* OR manage* OR develop*)) AND TITLE-ABS-KEY((practices OR methods OR tools)) AND TITLE-ABS-KEY((workflow OR process* OR lifecycle)))	2
Springer Link	(("ML Product"OR "Machine Learning Product" OR "AI Product"OR "Artificial Intelligence Product") AND ("product engineering"OR "product management" OR "product development") AND (build* OR manage* OR develop*) AND (practices OR methods OR tools) AND (workflow OR process* OR lifecycle))	5

Tabela 3 – Exclusion Criteria for papers.

	Exclusion Criteria
EC01	Not a paper
EC02	Improper source
EC03	Repeated
EC04	Studies that are not written in English
EC05	If the paper has less then 3 pages

To select the papers, we define four steps; they are:

- **Advanced Search in Scientific Repositories:** Automated research on a scientific basis. Table 2 exhibits the results. It will only include papers from journal, conference, congress or symposium;
- **Filter the papers:** A title based exclusion follow the Exclusion Criteria presented on Table 3;
- **Classify Step 1:** Based on the abstract, and, in some cases, read their content too. We also verify if it addresses our problem and research questions;
- **Classify Step 2:** To be more sure of the selected papers and If there is any doubt with the first validation, the support of the second author will be necessary for validation and to define the classification of the paper.

We define five exclusion criteria, presented in Table 3, to help and align what studies will be selected. We extract from the bases 32 papers, as presented in Table 2. Then we classify each paper with 1 or 2 stars: (1) discard or (2) core paper. To perform the classification process, we read the abstracts and, in some cases, the content to be more sure about the proper classification. We used the exclusion criteria presented in Table 3 and identified that irrelevant works use the search terms only for the background contextualization. Of the 32 publications found, only four were selected as core papers.

To ensure that we find the largest number of studies possible and not be collected by our search string, we also applied the snowballing process (WOHLIN, 2014b). We search for significant references and publications cited by one of the core papers previously selected. Therefore, we will now consider four additional core articles.

3.2 Gray Literature Protocol

Based on our RQs and the context defined with the PICOC method, we search by "ML Product"on significant blog repositories related to our work, namely Towards Data

Science and Medium. We also did searches on Open AI, but we did not get any significant results for that search. We define our selection criteria, supported by the guideline presented by Vahid et al. (GAROUSI; FELDERER; MÄNTYLÄ, 2019).

- **Reputation:** Verify the producer's authority, check if the author had some experience in this area, confirm if the author publishes other work in the field and if the publishing organization is reputable.
- **Methodology:** Verify if have a clearly stated aim and methodology, check if references support the article and if the work covers specific questions.
- **Objectivity:** In this case, we focus on verifying if existing business interests, confirm if the data support the conclusions, and if the works seem to be balanced.
- **Impact:** Verify if the paper has likes and comments, and if possible, how many readers.

4 Source of Knowledge

The diversity of information published on the development of software products is still little in the scientific community. Therefore, we search for techniques present in the papers found, and together with carefully selected blog articles, we collect practices, challenges, and tools used during the project's life cycle. We apply Peer-reviewed literature (SMITH, 2006), a right way of the scientific community to guarantee the work's quality and credibility. Figure 3 shows the evolution of publications, be they blogs such as Medium or Towards Data Science, or scientific bases such as Springer Link, IEEE, and ACM Digital Library. It is possible to notice a massive increase in publications since 2018, and this number has been growing considerably. Most of the information comes from blogs, but most have scientific foundations presented in scientific works.

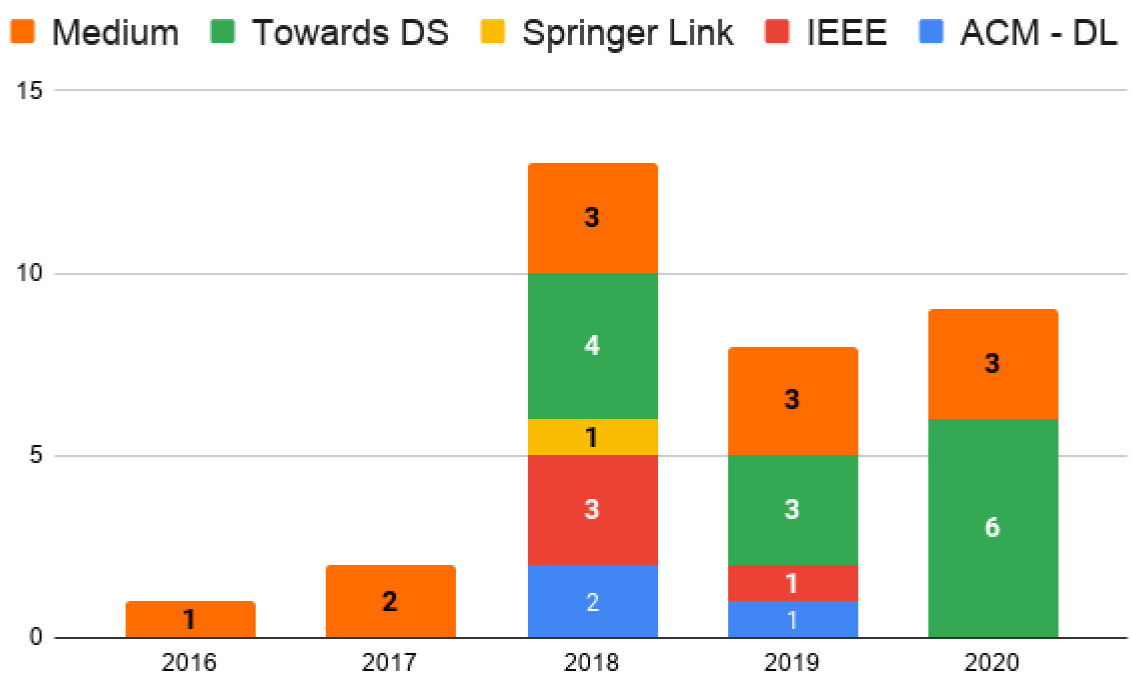

Figura 3 – Publications by source and publication year of core papers.

In this session, we present the results obtained with the Multivocal Review explained in more detail, as our core papers and related works are divided. First, we select core papers and relate to research questions, we also categorize related works and present how they are connected to that work.

In all, we select 32 core papers, as they are directly related to one or more research questions. The coding process was initiated by the scientific bases to verify better which techniques, practices, and challenges are present and confirmed in articles in the gray literature. For example, the practice Statement of Expectation and Intention was cited by two scientific papers and then cited by seven works by gray literature. It was also possible to notice the practices inserted in the blogs that were not mentioned in the scientific works, such as the Problem Identification process, which 12 works in the gray literature but were not discussed in the scientific works. Figure 4 present our core papers divided by years, where 2018 was the year we most collected Core papers. In the following years, there was a reduction in the number of white literature selected, but in compensation for gray literature remained constant.

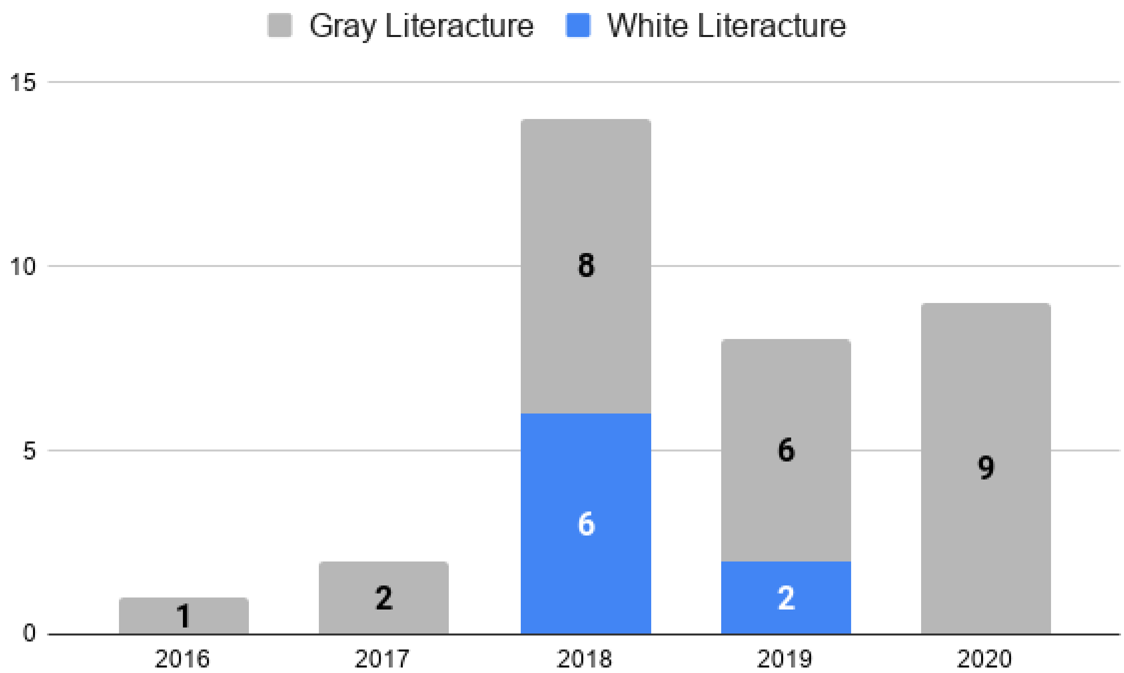

Figura 4 – Selected works divided between gray and white literature present by year.

5 Results and Discussion

Building a machine learning product can be a challenge because it is an exploratory and creative process. The sheer amount of processes and components that need to work together makes it challenging. Often the projects fail, not because of the ML product itself, but because of a problem in the definition and design, the environment and support, data processing, and other internal and external factors that interfere with creating the product. Here we present the challenges and the benefits if applied; the context of using practices and methodologies; tools and artifacts that assist in the problem identification process, solution design, data and model pipeline, and product integration, automation, and monitoring tools.

5.1 Challenges and Benefits

Of the 32 carefully selected core papers, we seek to answer our first research question *"What are the challenges and benefits of developing ML Products?"*. We analyze and identify a total of 12 challenges and nine benefits in applying or developing ML products. We present the complete list as follow.

CHALLENGES

- **I** OSS technologies are still new and may not fully meet requirements (HOLSTEIN et al., 2019; SCHELTER et al., 2018; HUANG, 2019; CHANDRASEKHAR, 2020; MEWALD, 2019; MEWALD, 2018a)

- **II** It does not have well-defined processes and good practices (HOLSTEIN et al., 2019; SCHELTER et al., 2018; HUANG, 2019; CHANDRASEKHAR, 2020; MEWALD, 2019; MEWALD, 2018a)

- **III** Multidisciplinary or cross-functional team (SHAMS, 2018; AMERSHI et al., 2019a; AKKIRAJU et al., 2020; HOLSTEIN et al., 2019; SCHELTER et al., 2018; HUANG, 2020; HUANG, 2019; CHANDRASEKHAR, 2020; MUKHERJEE, 2018b; GAVISH, 2017; FABRI, 2020; MEWALD, 2019)

- **IV** Build ML Products involves uncertain and require technical and organizational changes (HUANG, 2019; PATHA, 2020; MUKHERJEE, 2018b; MUKHERJEE, 2018a; SYNTHESIZED, 2020)

- **V** Require a deep understanding of ML algorithms and the consequences for the corresponding system (SCHELTER et al., 2018; AMPIL, 2019; MUKHERJEE, 2018b; MUKHERJEE, 2018a; MEWALD, 2019)

- **VI** Requires a hight power of processing (SHAMS, 2018; AMERSHI et al., 2019a; HOLSTEIN et al., 2019; SCHELTER et al., 2018)

- **VII** Automate all the process (AMERSHI et al., 2019a)

- **VIII** Labeling datasets is costly and time-consuming (SHAMS, 2018; AKKIRAJU et al., 2020; SCHELTER et al., 2018; AMPIL, 2019)

- **IX** Data Availability, Collection, Cleaning, and Management (AMERSHI et al., 2019a; HOLSTEIN et al., 2019)

- **X** Data change frequently (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; HOLSTEIN et al., 2019; MEWALD, 2018b; MEWALD, 2019; MEWALD, 2018a)

- **XI** ML is completely dependent on the quality and quantity of the data (SINGLA; BOSE; NAIK, 2018; HOLSTEIN et al., 2019; SCHELTER et al., 2018; PATHA, 2020; ATAEE, 2020; MUKHERJEE, 2018a; MEWALD, 2019; MEWALD, 2018a)

- **XII** Demand many times to take experiments (SCHELTER et al., 2018; HUANG, 2019; MUKHERJEE, 2018a; MEWALD, 2019; MEWALD, 2018a)

BENEFITS

- **I** Helps in business model innovation (METELSKAIA et al., 2018; REGALADO, 2018; SHAMS, 2018; SCHELTER et al., 2018; AKKIRAJU et al., 2020; HUANG, 2020; I., 2020; BARLASKAR, 2018; AMPIL, 2019)

- **II** Trends and patterns identification (SCHELTER et al., 2018; AKKIRAJU et al., 2020; HUANG, 2019; DEZHIC, 2018; BARLASKAR, 2018; AMPIL, 2019)

- **III** Help with decision making via Recommendations (HOLSTEIN et al., 2019; REGALADO, 2018; SHAMS, 2018; SCHELTER et al., 2018; HUANG, 2019; DEZHIC, 2018; BARLASKAR, 2018; AMPIL, 2019)

- **IV** Wide range of applications (HOLSTEIN et al., 2019; SINGLA; BOSE; NAIK, 2018; SCHELTER et al., 2018; AKKIRAJU et al., 2020; HUANG, 2019; BARLASKAR, 2018)

- **V** Helps medical diagnosis (HOLSTEIN et al., 2019; AKKIRAJU et al., 2020; AMPIL, 2019)

- **VI** Anomaly detection (HUANG, 2020; DEZHIC, 2018; AMPIL, 2019; CHANDRASEKHAR, 2020; MUKHERJEE, 2018b)
- **VII** Helps in data analysis (CHANDRASEKHAR, 2020; MUKHERJEE, 2018b)
- **VIII** Inovation (SCHELTER et al., 2018; AKKIRAJU et al., 2020; METELSKAIA et al., 2018; REGALADO, 2018; HUANG, 2019; I., 2020; BARLASKAR, 2018; PATHA, 2020; AGARWAL, 2019; ATAEE, 2020; MUKHERJEE, 2018a)
- **IX** Make the products more personalized, automated, and precise (HOLSTEIN et al., 2019; SINGLA; BOSE; NAIK, 2018; SCHELTER et al., 2018; AKKIRAJU et al., 2020; HUANG, 2020; DEZHIC, 2018; AMPIL, 2019; CHANDRASEKHAR, 2020; MUKHERJEE, 2018b)

First, Machine Learning is new compared to the standard software. It still does not have many product and software engineering studies that define the best practices and an established process. Also, the tools are very new, and most of the Open Source tools are still emerging and may not fully meet the requirements for development (HUANG, 2019; CHANDRASEKHAR, 2020). For example, in a project that needs to develop an image classifier, the library selected by the team has a gap in the data collection part and must build something to solve the gap. This ends up becoming a challenge and creates risks for the project, so these points must be taken into account during the solution's design. People with more technical knowledge can identify possible problems with the selected tools. Therefore, a multidisciplinary and Cross-functional team is required, including not only ML engineers and scientists but also data engineers, software engineers, UX UI specialists, and hardware engineers to ensure the quality of the established product (AKKIRAJU et al., 2020; HUANG, 2020; HUANG, 2019; CHANDRASEKHAR, 2020; MUKHERJEE, 2018b; GAVISH, 2017; FABRI, 2020).

Before starting to build a machine learning product, it is essential to also take into account that build ML products involves uncertain and require technical and organizational changes (HUANG, 2019; PATHA, 2020; MUKHERJEE, 2018b; MUKHERJEE, 2018a; SYNTHESIZED, 2020). It is necessary because the companies are accostumed to developing products that can be directly implemented with functions and scripts inserted into the code. However, machine learning products require special care with the data and demand time to conduct research and do tests to select the best model and parameters to be used, and this requires time and financial investment without knowing the expected result will be achieved.

The Data pipeline is the biggest challenge identified in the study because first, ML is entirely dependent on the quality and quantity of the data (SINGLA; BOSE; NAIK, 2018; PATHA, 2020; ATAEE, 2020; MUKHERJEE, 2018a). Therefore, to guarantee quality and correct quantity, the labeling datasets process is sometimes necessary but is costly and time-consuming. For example, in cases that do not have structured data, it guarantees a useful dataset (gold standard) is essential (SHAMS, 2018; AMPIL, 2019). Ensuring Data Availability, Collection, Cleaning, and Management (AMERSHI et al., 2019a) is a challenge too because Data Change Frequently (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; MEWALD, 2018b), but the evolution and updating of the data are significant to ensure continuous learning.

However, depending on the scenario, machine learning is indispensable, and sometimes the only solution, because these machines make predictions and improve insights based on patterns that are useful for challenges problems that humans had difficulty in programming. Besides, this product is being adopted because it guarantees a significant improvement in the company's business model's innovation, which gets to know its customers and users more (METELSKAIA et al., 2018; REGALADO, 2018). Many companies are adopting development because they have a significant amount of data, and through analysis, they can make better decisions and make the products more personalized, automated, and precise (HOLSTEIN et al., 2019; SINGLA; BOSE; NAIK, 2018; SCHELTER et al., 2018; AKKIRAJU et al., 2020; HUANG, 2020; DEZHIC, 2018; AMPIL, 2019; CHANDRASEKHAR, 2020; MUKHERJEE, 2018b).

5.2 Methods and Practices

To answer our second research question *"What are the practices and methods adopted in product development?"*. We found practices and methods that help understand the problem and define the product (e.g., Verify if ML is Really Necessary (HOLSTEIN et al., 2019; CHANDRASEKHAR, 2020) and Establish what the outcome is and what the data can offer (SHAMS, 2018)) that have fundamental roles in the initial ML product development process. We also find more technical practices and methods related to data and model management and automation techniques (e.g., Continuous Monitoring (AMERSHI et al., 2019a; SCHELTER et al., 2018; AKKIRAJU et al., 2020) and Build Pipelines Specialized (AMERSHI et al., 2019a; AKKIRAJU et al., 2020)). To guarantee a brief organization, the list below presents methods and practices organized into the following categories: Problem Definition and Solution Design, Product Management, Data Management, Model Management, and Delivery and Runtime.

Problem Definition and Solution Design

- **I** Bussines Continuous Validation (REGALADO, 2018; AKKIRAJU et al., 2020; HUANG, 2019; PATHA, 2020; MUKHERJEE, 2018b)
- **II** Verify if ML is Really Necessary (HUANG, 2020; HUANG, 2019; I., 2020; DEZHIC, 2018; BARLASKAR, 2018; CHANDRASEKHAR, 2020; MUKHERJEE, 2018b; MUKHER-

JEE, 2018a; GAVISH, 2017; HURLOCK, 2016; PATHAK, 2018; FABRI, 2020)

- **III** Define the Role of ML on Product (HOLSTEIN et al., 2019; CHANDRASEKHAR, 2020; AGARWAL, 2019; MUKHERJEE, 2018a; GAVISH, 2017; FABRI, 2020)
- **IV** Statement of Expectation and Intention (SHAMS, 2018; AKKIRAJU et al., 2020; HUANG, 2019; CHANDRASEKHAR, 2020; MUKHERJEE, 2018b; MUKHERJEE, 2018a; GAVISH, 2017; HURLOCK, 2016; SYNTHESIZED, 2020)
- **V** Build the Product Trust (SHAMS, 2018; AKKIRAJU et al., 2020; HUANG, 2019)
- **VI** Define the Desired Outcome (AKKIRAJU et al., 2020; SINGLA; BOSE; NAIK, 2018; HUANG, 2019; DEZHIC, 2018; AMPIL, 2019, 2019; PATHA, 2020; MUKHERJEE, 2018b; GAVISH, 2017; MEWALD, 2018b; HURLOCK, 2016)

Product Management

- **I**Improvement Using User Feedback (SHAMS, 2018; CHANDRASEKHAR, 2020; AGARWAL, 2019; MUKHERJEE, 2018a; MEWALD, 2018b; PATHAK, 2018)
- **II**Establish what is the outcome and what the data can offer (SHAMS, 2018; I., 2020; MUKHERJEE, 2018a)
- **III**Review the Literature (AMPIL, 2019; MUKHERJEE, 2018b; MUKHERJEE, 2018a)
- **IV**Learn From Retrospective Meetings and Logs (SHAMS, 2018; AKKIRAJU et al., 2020; SINGLA; BOSE; NAIK, 2018; I., 2020; AGARWAL, 2019)
- **V**Risk Management (SHAMS, 2018; AMPIL, 2019; PATHA, 2020; PATHAK, 2018)
- **VI**Multiple Interactions with users and stakeholders to colect feedback (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; MUKHERJEE, 2018b; MUKHERJEE, 2018a)
- **VII**A/B testing or Split Testing (SHAMS, 2018; SCHELTER et al., 2018; AMPIL, 2019; MUKHERJEE, 2018a; FABRI, 2020)
- **VIII**Evaluate Results, Define Metrics and Baselines (SCHELTER et al., 2018; AKKIRAJU et al., 2020; HUANG, 2019; I., 2020; DEZHIC, 2018; AMPIL, 2019; AGARWAL, 2019; MUKHERJEE, 2018b; GAVISH, 2017; HURLOCK, 2016; FABRI, 2020)
- **IX**Test Early and Frequently from end to end (SHAMS, 2018; HOLSTEIN et al., 2019; AKKIRAJU et al., 2020; HUANG, 2019; DEZHIC, 2018; GAVISH, 2017; HURLOCK, 2016)

- **X**Define the Data Strategy (AKKIRAJU et al., 2020; HUANG, 2019; I., 2020; PATHA, 2020; GAVISH, 2017; MEWALD, 2018b)

- **XII**Feedback Loops (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; BARLASKAR, 2018; AGARWAL, 2019; GAVISH, 2017; HURLOCK, 2016; PATHAK, 2018; SCHELTER et al., 2018; MEWALD, 2018a)

Data Management

- **I** Data Requirements (AKKIRAJU et al., 2020; GAVISH, 2017; MEWALD, 2018b; SCHELTER et al., 2018; FANOUS, 2020)

- **II** Ensure the Reliability and Availability of Data (SHAMS, 2018; AMERSHI et al., 2019a; AKKIRAJU et al., 2020; HUANG, 2019; I., 2020; GAVISH, 2017; MEWALD, 2018b; HURLOCK, 2016; FABRI, 2020; SCHELTER et al., 2018; FANOUS, 2020)

- **III** Define the Data Pipeline (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; HUANG, 2019; ATAEE, 2020; LATHIA, 2017; SCHELTER et al., 2018; FANOUS, 2020)

- **IV** Data Collection and Evolution (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; BARLASKAR, 2018; AMPIL, 2019; PATHA, 2020; AGARWAL, 2019; ATAEE, 2020; GAVISH, 2017; MEWALD, 2018b; LATHIA, 2017; RAGHUWANSHI, 2019; HURLOCK, 2016; PATHAK, 2018; COGAN, 2019; HOLSTEIN et al., 2019; SCHELTER et al., 2018; FANOUS, 2020)

- **V** Data Cleaning (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; I., 2020; DEZHIC, 2018; BARLASKAR, 2018; AGARWAL, 2019; GAVISH, 2017; HURLOCK, 2016; FABRI, 2020; HOLSTEIN et al., 2019; SCHELTER et al., 2018; FANOUS, 2020)

- **VI** Data Labeling (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; BARLASKAR, 2018; AMPIL, 2019; MEWALD, 2018b; HURLOCK, 2016; SCHELTER et al., 2018; FANOUS, 2020)

- **VII** Data Integrations (AKKIRAJU et al., 2020; SCHELTER et al., 2018; FANOUS, 2020)

- **VIII** Data Management (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; AGARWAL, 2019; ATAEE, 2020; FABRI, 2020; COGAN, 2019; SCHELTER et al., 2018; FANOUS, 2020)

- **IX** Data Transformation (AKKIRAJU et al., 2020; SCHELTER et al., 2018; FANOUS, 2020)

- **X** Data Reuse (AMERSHI et al., 2019a; MEWALD, 2018a; FANOUS, 2020)

- **XI** Data Versioning (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; SCHELTER et al., 2018; FANOUS, 2020)

Model Management

- **I** Research ML Libraries and Frameworks to be Used (SHAMS, 2018; AKKIRAJU et al., 2020; I., 2020; BARLASKAR, 2018; GAVISH, 2017; LATHIA, 2017; HURLOCK, 2016; SYNTHESIZED, 2020; SCHELTER et al., 2018; FANOUS, 2020)

- **II** Code Reusability (SHAMS, 2018; AKKIRAJU et al., 2020; I., 2020; AMPIL, 2019; ATAEE, 2020; FANOUS, 2020)

- **III** Ensemble Learning (AMERSHI et al., 2019a; GAVISH, 2017; FABRI, 2020; SCHELTER et al., 2018)

- **IV** Model Requirements (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; BARLASKAR, 2018; HURLOCK, 2016)

- **VI** Test Multiple Hypotheses (SHAMS, 2018; AKKIRAJU et al., 2020; SINGLA; BOSE; NAIK, 2018; GAVISH, 2017; FABRI, 2020; SCHELTER et al., 2018)

- **VII** Model Training (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; AMPIL, 2019; AGARWAL, 2019; ATAEE, 2020; GAVISH, 2017; HURLOCK, 2016; FABRI, 2020; COGAN, 2019; SCHELTER et al., 2018)

- **VIII** Modularizing Train Code (AKKIRAJU et al., 2020; FABRI, 2020; SCHELTER et al., 2018)

- **IX** Measure Precision, Recall, and Accuracy (AKKIRAJU et al., 2020; SINGLA; BOSE; NAIK, 2018; SCHELTER et al., 2018; HUANG, 2019; BARLASKAR, 2018; ATAEE, 2020; GAVISH, 2017; SYNTHESIZED, 2020)

- **X** Model Evaluation (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; AMPIL, 2019; AGARWAL, 2019; ATAEE, 2020; MEWALD, 2018b; RAGHUWANSHI, 2019; HURLOCK, 2016; FABRI, 2020; SCHELTER et al., 2018; MEWALD, 2018a)

- **XI** Model Versioning (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; FABRI, 2020)

- **XII** Feature Engineering (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; DEZHIC, 2018; AMPIL, 2019; AGARWAL, 2019; LATHIA, 2017; HURLOCK, 2016; COGAN, 2019)

Delivery and Runtime

- **I** Model Deployment (AMERSHI et al., 2019a; AMPIL, 2019; SCHELTER et al., 2018; FANOUS, 2020)
- **II** Build Pipelines Specialized (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; I., 2020; AGARWAL, 2019; ATAEE, 2020; FABRI, 2020; COGAN, 2019)
- **III** Automation (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; AMPIL, 2019; AGARWAL, 2019; LATHIA, 2017; SYNTHESIZED, 2020; HOLSTEIN et al., 2019; MEWALD, 2018a)
- **IV** Focus on Infrastructure (AKKIRAJU et al., 2020; DEZHIC, 2018; ATAEE, 2020)
- **V** Continuous Monitoring (AMERSHI et al., 2019a; SCHELTER et al., 2018; AKKIRAJU et al., 2020; I., 2020; AGARWAL, 2019; LATHIA, 2017; PATHAK, 2018; FABRI, 2020)

The first category is related to Phases 1 and 2 of the development of an ML product, where one must analyze and study the problem and then design solutions for them. One of the main points for developing the project that must be taken into account from the beginning is the Continuous Validation Bussines practice, which assists the business's growth and verifies if the product is related to the goals (REGALADO, 2018), and if the product meets the requirements established (AKKIRAJU et al., 2020; HUANG, 2019). During the problem definition phase, it is essential to identify the problem and Verify if ML is Really Necessary to develop the product, as the cost and challenges of developing an ML product are high.

After the problem is defined, it is possible to start the solution design process (Phase 2), consisting of design, establishing, and aligning the solution among all involved. The first method is Define the role of ML on product (CHANDRASEKHAR, 2020), which consists of verifying whether the ML is a core or a feature of the final product and if it is a black box product (e.g., deep learning), or the team has more access in building the model (AGARWAL, 2019). After, it is necessary to make sure all the stakeholders and the company agree on problem-solving and how to use the solution. One technique used is The Statement of Expectation and Intention (SHAMS, 2018; AKKIRAJU et al., 2020), which consists of establishing the product's objectives, aligning the specifications and intentions of the same among the interested parties.

Defines the Desired Outcome before starting the project is essential to explain what the model is trying to predict or identify patterns and ensure that the results are linked to the company's objectives (AKKIRAJU et al., 2020; SINGLA; BOSE; NAIK, 2018). Some methods, such as prototyping (SHAMS, 2018; GAVISH, 2017), design thinking (REGALADO, 2018; SYNTHESIZED, 2020), and lean canvas (REGALADO, 2018; METELSKAIA et al., 2018), help understand the potential business opportunity and define the solution.

The category Product Management can be considered transversal, as it has methods and practices that must be carried out throughout the project's life cycle and even outside of it. We identify 12 methods and practices related to that. One of the most cited is Improvement Using Explicit and Implicit User Feedback, which should be used throughout the project, regardless of its stage, receiving explicit feedback from prototype users and implicit feedback from the end-users (SHAMS, 2018).

Learn From Retrospective Meetings is a technique of using retrospective and postmortems to help plan new products to improve the current project in development (SHAMS, 2018; AKKIRAJU et al., 2020; SINGLA; BOSE; NAIK, 2018), and Learn From Logs refers to the use of decision-making logs during prototyping. During product construction, this can help to define the phases of new products avoiding errors commented in the past and using practices or decisions that positively impacted the result Final (SHAMS, 2018; I., 2020; AGARWAL, 2019).

During all the pipeline, it is necessary to apply techniques to ensure alignment on the results found with team members and those interested in the product. Therefore, a widely cited technique is Multiple Interactions to improve, receive feedback, and test the models (AMERSHI et al., 2019a; AKKIRAJU et al., 2020; MUKHERJEE, 2018b; MUKHERJEE, 2018a). The A/B testing (Split testing) is used to determine the best product or workflow. This testing also has a clearly defined set of KPIs to decide upon (SHAMS, 2018; AMPIL, 2019; MUKHERJEE, 2018a; FABRI, 2020). The definition of an MVP is also used as it assists in the development focused on the user's needs and is excellent for receiving feedback from users and improving (MUKHERJEE, 2018a; HURLOCK, 2016).

Evaluate results, define metrics and baselines is a technique used to determine the metrics early and identify the success and failure of the product (GAVISH, 2017; HURLOCK, 2016; FABRI, 2020). Fail fast research and deploy/test early and frequently from end to end using agile methods, prototype, research and deploy fast to receive feedbacks and identify mistakes (AKKIRAJU et al., 2020; HUANG, 2019).

Finally, defining the Data Strategy before starting the project is essential to determine and think about the advantage of using this data related to the competitors. Does the marketing team understand the nature of ML products? The pros and cons? And the trade-offs we are making? What are the consequences and costs of making a wrong prediction? Is the company prepared to answer all these questions? The strategy of data will be the differential related with the competitors (AKKIRAJU et al., 2020; HUANG, 2019; GAVISH, 2017; MEWALD, 2018b).

The Data Management and Model Management categories are related to Phase 3. They consist of the work performed by scientists and data engineers to ensure product development as specified in Phase 2. Therefore, it is first necessary to ensure data quality because machine learning is entirely dependent on the quantity and quality of the data. The methods used in this category are Data Requirement (AKKIRAJU et al., 2020), Ensure the Reliability and Availability of Data (AMERSHI et al., 2019a; HUANG, 2019), Data Collection and Evolution (GAVISH, 2017; COGAN, 2019), Data Cleaning (DEZHIC, 2018; BARLASKAR, 2018), and Data Labeling (MEWALD, 2018b; HURLOCK, 2016), because whenever you start product development, the first thing to do is to ensure quality and reliability of the data and whether it meets the specified requirements. Therefore, an exploration is made on the data to check and identify possible errors and understand if the data is beneficial for the proposed problem. As this phase consists mainly of ensuring data quality, data management (AGARWAL, 2019), Data Reuse (AMERSHI et al., 2019a) and Data Versioning (SCHELTER et al., 2018) techniques and practices must also be applied to guarantee a management of the data thus allowing them to be used in new projects.

Phases 4 and 5 of the product development cycle are represented by the Delivery and Runtime category, which consists of the Deploy of the model (AMERSHI et al., 2019a; AMPIL, 2019), using the technique of developing a Pipeline specialized for the situation (AMERSHI et al., 2019a; AKKIRAJU et al., 2020) and allowing engineers to experiment with different permutations of algorithms without complications and decoupling the models from the infrastructure components, to allow for easy updates (I., 2020). Another technique widely used in this phase is Automation (AMPIL, 2019; AGARWAL, 2019; LATHIA, 2017; SYNTHESIZED, 2020), which allows teams to aggregate data, extract resources and synthesize more efficiently labeled examples throughout the development cycle.

5.3 Tools and Artifacts

Answering our third research question *"What are the artifacts and tools adopted in the life cycle of ideation, design, and construction of the ML product?"*. On list bellow, we organize the artifacts and tools found in three categories: Management, Development, and Deploy and Runtime.

Management

- **I** Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (SHAMS, 2018)
- **II** Key Performance Indicator (KPI) (SHAMS, 2018; HOLSTEIN et al., 2019; HUANG, 2019; MUKHERJEE, 2018b; COGAN, 2019)

- **III** Product Requirement Document (PRD) (MEWALD, 2018b)
- **IV** DataFlowDiagrams (DFD) (SHAMS, 2018)
- **VI** Service Level Agreement (SLAs) (AGARWAL, 2019)
- **VII** Data Scorecard (PATHA, 2020)
- **VIII** Balanced Scorecard (PATHA, 2020)
- **IX** Root-mean-square Deviation (RMSE) (AGARWAL, 2019)

Development

- **I** fastai (I., 2020)
- **II** Amazon recognition (I., 2020)
- **III** Interactive Computing Notebooks (jupyter) (SHAMS, 2018)
- **IV** Tensorflow (AKKIRAJU et al., 2020; SCHELTER et al., 2018; CHANDRASEKHAR, 2020)
- **V** PyTorch (SCHELTER et al., 2018; CHANDRASEKHAR, 2020)
- **VI** FAISS (CHANDRASEKHAR, 2020)
- **VII** keras (AKKIRAJU et al., 2020; I., 2020)
- **VIII** Sklearn (SCHELTER et al., 2018; I., 2020; HURLOCK, 2016)
- **IX** Pandas (SCHELTER et al., 2018)
- **X** MXNet (SCHELTER et al., 2018)
- **XI** SparkML (SCHELTER et al., 2018)

Deploy and Runtime

- **I** crontab, Airflow, Luigi (AMPIL, 2019)
- **II** AWS EC2 (AMPIL, 2019)
- **III** Sagemaker (CHANDRASEKHAR, 2020)
- **IV** Kubeflow (CHANDRASEKHAR, 2020)

The first category discussed is related to the artifacts used to manage and guarantee the products' quality under development. To ensure the alignment of the established solution, it is necessary first to develop an artifact called Product Requirement Document (MEWALD, 2018b) (POHL, 2010) that is used whenever possible to collect and document the requirements that a product needs to meet the needs of the user or the customer. This artifact is also widely used in agile methodologies with user stories (COHN, 2004), for example. Therefore, this artifact can be built in the way that most adds to the product. Still, its existence is essential to align the product's understanding between the stakeholders and the development team. During this process, it is necessary to make an agreement with the customers, using Service Level Agreement (SLA) (AGARWAL, 2019) on the solution proposals and the development deadlines. It is essential to highlight the risks and particularities already presented above about the product development.

To guarantee product quality is necessary to find metrics and indicators to assess risks, the quality of models and data, and the development cycle. In this case, the Balanced Scorecard (PATHA, 2020) is widely used. It consists of a measurement and performance management methodology that aligns the objectives with the project's evolution, asks questions, monitors external and internal risks, and maintains transparent communication. This document will monitor the project's timeline, milestones, alignment on critical issues, and architectural changes. External threats are more linked to the customers' expectations, checking if there is any gap between the cost and the added value, the quality of the model, and if it presents any results outside of expectations and violates any law or company rules.

Another essential point is to verify the quality of the data and the model used. That is why the Data Quality Scorecard (PATHA, 2020) is also used, which consists of a spreadsheet to check the quality and follow the issues that impact the data, verifying if the quality of labels is satisfactory, and most important verify if the data representative of production. Root-mean-square Deviation (RMSE) (AGARWAL, 2019) is used to measure a model's error in predicting quantitative data and identifies the differences between the values predicted by a model and the observed values.

Of the development tools, Interactive Computing Notebooks, also known as Jupiter notebook (SHAMS, 2018), is widely used as a log by engineers and can help in planning phases of other products because of record decision they make and every decision they change. TensorFlow (AKKIRAJU et al., 2020; SCHELTER et al., 2018; CHANDRASEKHAR, 2020) and PyTorch (SCHELTER et al., 2018; CHANDRASEKHAR, 2020) libraries are widely used Open Source tools and applicable to a wide variety of tasks. TensorFlow is widely used to create and train neural networks to identify and detect patterns and correlations. PyTorch, on the other hand, is a tool developed by Facebook's AI Research lab (FAIR) (FACEBOOK. . . , 2020), and is used to build solutions that involve computer vision and natural language processing. The chances of finding a project that uses the panda's library, as it is a powerful tool used for data manipulation and analysis. The Keras (AKKIRAJU et al., 2020; I., 2020) tool can be used mainly on TensorFlow as it allows you to do quick experiments with deep neural networks. It focuses on being easy to use, modular, and extensible.

Most of the tools related to the infrastructure were developed by the big companies that apply ML in their products, as in Airbnb (AIRBNB. . . , 2020) (online accommodation management service) that created AirFlow, an Open Source software that allows schedule workflows and monitors them via a user interface (AMPIL, 2019). If real-time predictions are necessary, then the model will likely be deployed using AWS EC2 (AMPIL, 2019), a tool used to assist in cloud computing that enables the prediction script running on the cloud. The infrastructure tools cited were: Sagemaker (CHANDRASEKHAR, 2020), which consists of a machine learning platform in the cloud launched in November 2017 by Amazon that allows the creation, training, and deployment of machine learning models in the cloud. And Kubeflow (CHANDRASEKHAR, 2020), which enables the use of machine learning pipelines to orchestrate complicated workflows running on Kubernetes. These tools are still very new on the market, which can significantly impact deploying the application.

6 Conclusion and Future Works

This work consisted in determining which practices and methods are adopted during the process of developing a Machine Learning product. So far, we have collected information using the Multivocal Literature Review (MLR), searching the primary databases and blogs for information on the Machine Learning product development cycle.

Analyzing this information, we identified a significant amount of techniques and methods, some very similar to those applied in standard software development. But as machine learning is especially about data, some methods are used to improve the quality in the maintenance of this kind of product. These techniques help developers, engineers, and project managers develop products with more quality. Still, in compensation, the works found do not delve much into the challenges and advantages of applying ML, nor do we see much information about the tools and artifacts used by the market and academia that satisfy the product's need. Therefore, it will be necessary to increase the information source to collect information from other data base that contain and are heavily used by the community.

A summary of this work was presented at the International Conference on Software Engineering (ICSE). WAIN'21 - 1st Workshop on AI Engineering – Software Engineering for AI. Annex A presents the position paper submitted and presented.

Referências

AGARWAL, R. *Six Important Steps to Build a Machine Learning System.* 2019. Blog post at Towards Data Science <https://towardsdatascience.com/6-important-steps-to-build-a-machine-learning-system-d75e3b83686>. Accessed on Nov 2020. Code: TDS-09. Citado 9 vezes nas páginas 29, 31, 32, 33, 34, 35, 36, 37 e 38.

AIRBNB home page. 2020. <https://www.airbnb.com>. Accessed on Nov 2020. Citado na página 39.

AKKIRAJU, R. et al. Characterizing machine learning processes: A maturity framework. In: SPRINGER. *International Conference on Business Process Management.* [S.l.], 2020. p. 17–31. Accessed on Nov 2020. Code: SB-01. Citado 15 vezes nas páginas 15, 16, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38 e 39.

ALPAYDIN, E. *Introduction to machine learning.* [S.l.]: MIT press, 2020. Citado na página 16.

AMERSHI, S. et al. Software engineering for machine learning: A case study. In: IEEE. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP).* [S.l.], 2019. p. 291–300. Code: IEEE-02. Citado 12 vezes nas páginas 11, 16, 17, 27, 28, 30, 31, 32, 33, 34, 35 e 36.

AMERSHI, S. et al. Software engineering for machine learning: A case study. In: IEEE. *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP).* [S.l.], 2019. p. 291–300. Citado na página 11.

AMPIL, L. *Basics of Data Science Product Management: The ML Workflow.* 2019. Blog post at Towards Data Science <https://towardsdatascience.com/basics-of-ai-product-management-orchestrating-the-ml-workflow-3601e0ff6baa>. Accessed on Nov 2020. Code: TDS-06. Citado 11 vezes nas páginas 28, 29, 30, 31, 32, 33, 34, 35, 36, 37 e 39.

ANON, J.; VILLAUMBROSIA, C. G. de. *The Product Book.* [S.l.]: Product School, 2017. Citado 3 vezes nas páginas 14, 15 e 17.

ATAEE, P. *Build An ML Product — 4 Mistakes To Avoid.* 2020. Blog post at Towards Data Science <https://towardsdatascience.com/build-an-ml-product-4-mistakes-to-avoid-bce30d98bd24>. Accessed on Nov 2020. Code: TDS-10. Citado 5 vezes nas páginas 28, 29, 32, 33 e 34.

BARLASKAR, U. *Machine Learning for Product Managers.* 2018. Blog post at Towards Data Science <https://towardsdatascience.com/machine-learning-for-product-managers-part-i-problem-mapping-5436132c3a6e>. Accessed on Nov 2020. Code: TDS-05. Citado 7 vezes nas páginas 28, 29, 30, 31, 32, 33 e 36.

BRERETON, P. et al. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, Elsevier, v. 80, n. 4, p. 571–583, 2007. Citado na página 19.

BUDGEN, D.; BRERETON, P. Performing systematic literature reviews in software engineering. In: *Proceedings of the 28th international conference on Software engineering*. [S.l.: s.n.], 2006. p. 1051–1052. Citado na página 19.

CHANDRASEKHAR, G. *7 Elements of AI Product Strategy*. 2020. Blog post at Towards Data Science <https://towardsdatascience.com/defining-your-ai-product-strategy-7-areas-of-focus-2cf112c82c07>. Accessed on Nov 2020. Code: TDS-08. Citado 8 vezes nas páginas 27, 29, 30, 31, 34, 37, 38 e 39.

CHARMAZ, K. Grounded theory as an emergent method. *Handbook of emergent methods*, v. 155, p. 172, 2008. Citado na página 19.

CLEMENTS, P. *Software product lines*. [S.l.: s.n.]. Citado na página 15.

COGAN, J. *The Surprising Truth About What it Takes to Build a Machine Learning Product*. 2019. Blog post at Medium <https://medium.com/thelaunchpad/the-ml-surprise-f54706361a6c>. Accessed on Nov 2020. Code: M-09. Citado 4 vezes nas páginas 32, 33, 34 e 36.

COHN, M. *User stories applied: For agile software development*. [S.l.]: Addison-Wesley Professional, 2004. Citado na página 38.

COLUMBUS, L. *Roundup Of Machine Learning Forecasts And Market Estimates*. 2020. <https://www.forbes.com/sites/louiscolumbus/2020/01/19/roundup-of-machine-learning-forecasts-and-market-estimates-2020/?sh=748ec7595c02>. Accessed on Oct 2020. Citado na página 11.

DEZHIC, E. *Machine Learning Rules in a Nutshell*. 2018. Blog post at Towards Data Science <https://towardsdatascience.com/machine-learning-rules-in-a-nutshell-dfc3e6839163>. Accessed on Nov 2020. Code: TDS-04. Citado 8 vezes nas páginas 28, 29, 30, 31, 32, 33, 34 e 36.

DITTRICH, Y. Software engineering beyond the project–sustaining software ecosystems. *Information and Software Technology*, Elsevier, v. 56, n. 11, p. 1436–1456, 2014. Citado na página 13.

FABRI, S. *Creating ML AI Products — The Smart Products Pipeline*. 2020. Blog post at Medium <https://medium.com/@simon.fabri_25096/the-smart-products-pipeline-404472842bfb>. Accessed on Nov 2020. Code: M-08. Citado 8 vezes nas páginas 27, 29, 30, 31, 32, 33, 34 e 35.

FACEBOOK AI research home page. 2020. <https://ai.facebook.com>. Accessed on Nov 2020. Citado na página 38.

FANOUS, K. *Software 2.0 vs Software 1.0*. 2020. Blog post at Medium <https://medium.com/@karimfanous/software-2-0-vs-software-1-0-5df3f0d6d62a>. Accessed on Nov 2020. Code: M-19. Citado 3 vezes nas páginas 32, 33 e 34.

FICHMAN, R. G.; KEMERER, C. F. Adoption of software engineering process innovations: The case of object orientation. *Sloan management review*, v. 34, p. 7–7, 1993. Citado na página 13.

FUTRELL, R. T.; SHAFER, L. I.; SHAFER, D. F. *Quality software project management*. [S.l.]: Prentice Hall PTR, 2001. Citado na página 13.

GAROUSI, V.; FELDERER, M.; MÄNTYLÄ, M. V. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, Elsevier, v. 106, p. 101–121, 2019. Citado 2 vezes nas páginas 19 e 24.

GAVISH, Y. *The Step-By-Step PM Guide to Building Machine Learning Based Products*. 2017. Blog post at Medium <https://medium.com/@yaelg/product-manager-pm-step-by-step-tutorial-building-machine-learning-products-ffa7817aa8ab>. Accessed on Nov 2020. Code: M-01. Citado 8 vezes nas páginas 27, 29, 30, 31, 32, 33, 35 e 36.

HARRINGTON, P. *Machine learning in action*. [S.l.]: Manning Publications Co., 2012. Citado na página 16.

HAZELWOOD, K. et al. Applied machine learning at facebook: A datacenter infrastructure perspective. In: IEEE. *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. [S.l.], 2018. p. 620–629. Citado na página 16.

HERBSLEB, J. et al. Software quality and the capability maturity model. *Communications of the ACM*, ACM New York, NY, USA, v. 40, n. 6, p. 30–40, 1997. Citado na página 13.

HOLSTEIN, K. et al. Improving fairness in machine learning systems: What do industry practitioners need? In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. [S.l.: s.n.], 2019. p. 1–16. Citado 8 vezes nas páginas 27, 28, 29, 30, 31, 32, 34 e 36.

HUANG, B. *How to Manage Machine Learning Products*. 2019. Blog post at Towards Data Science <https://towardsdatascience.com/how-to-manage-machine-learning-products-part-1-386e7011258a>. Accessed on Nov 2020. Code: TDS-02. Citado 10 vezes nas páginas 27, 28, 29, 30, 31, 32, 33, 34, 35 e 36.

HUANG, B. *Three Questions Every ML Product Manager Must Answer*. 2020. Blog post at Towards Data Science <https://towardsdatascience.com/three-questions-every-ml-product-manager-must-answer-35d73127cd5d>. Accessed on Nov 2020. Code: TDS-01. Citado 5 vezes nas páginas 27, 28, 29, 30 e 31.

HUMPHREY, W. S. Characterizing the software process: a maturity framework. *IEEE software*, IEEE, v. 5, n. 2, p. 73–79, 1988. Citado na página 13.

HURLOCK, J. *Building Machine Learning based Products*. 2016. Blog post at Medium <>. Accessed on Nov 2020. Code: M-05. Citado 7 vezes nas páginas 30, 31, 32, 33, 35, 36 e 37.

I., C. *My Framework For Helping Startups Build And Deploy Data Science*. 2020. Blog post at Towards Data Science <https://towardsdatascience.com/my-framework-for-helping-startups-build-and-deploy-data-science-43cf40bc1a1e>. Accessed on Nov 2020. Code: TDS-03. Citado 11 vezes nas páginas 28, 29, 30, 31, 32, 33, 34, 35, 36, 37 e 39.

KHAN, A. Project scope management. *Cost engineering*, American Association of Cost Engineers, v. 48, n. 6, p. 12–16, 2006. Citado na página 14.

KITCHENHAM, B. et al. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, Elsevier, v. 51, n. 1, p. 7–15, 2009. Citado na página 19.

KOUROU, K. et al. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, Elsevier, v. 13, p. 8–17, 2015. Citado na página 11.

LATHIA, N. *Machine Learning for Product Managers*. 2017. Blog post at Medium <https://medium.com/@neal_lathia/machine-learning-for-product-managers-ba9cf8724e57>. Accessed on Nov 2020. Code: M-03. Citado 4 vezes nas páginas 32, 33, 34 e 36.

LENARDUZZI, V.; TAIBI, D. Mvp explained: A systematic mapping study on the definitions of minimal viable product. In: IEEE. *2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. [S.l.], 2016. p. 112–119. Citado na página 14.

LIBBRECHT, M. W.; NOBLE, W. S. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, Nature Publishing Group, v. 16, n. 6, p. 321–332, 2015. Citado na página 11.

LINDEN, F. J. Van der; SCHMID, K.; ROMMES, E. *Software product lines in action: the best industrial practice in product line engineering*. [S.l.]: Springer Science & Business Media, 2007. Citado na página 13.

LWAKATARE, L. E. et al. Devops in practice: A multiple case study of five companies. *Information and Software Technology*, Elsevier, v. 114, p. 217–230, 2019. Citado na página 13.

METELSKAIA, I. et al. A business model template for ai solutions. In: *Proceedings of the International Conference on Intelligent Science and Technology*. [S.l.: s.n.], 2018. p. 35–41. Code: ACM-02. Citado 4 vezes nas páginas 28, 29, 30 e 35.

MEWALD, C. *3 Common Problems With Your Machine Learning Product and How to Fix Them*. 2018. Blog post at Medium <https://medium.com/thelaunchpad/how-to-protect-your-machine-learning-product-from-time-adversaries-and-itself-ff07727d6712>. Accessed on Nov 2020. Code: M-13. Citado 5 vezes nas páginas 27, 28, 32, 33 e 34.

MEWALD, C. *Data: A key requirement for your Machine Learning (ML) product*. 2018. Blog post at Medium <https://medium.com/thelaunchpad/data-a-key-requirement-for-your-machine-learning-ml-product-9195ace977d4>. Accessed on Nov 2020. Code: M-02. Citado 9 vezes nas páginas 28, 30, 31, 32, 33, 35, 36, 37 e 38.

MEWALD, C. *Your Deep-Learning-Tools-for-Enterprises Startup Will Fail*. 2019. Blog post at Medium <https://medium.com/thelaunchpad/your-deep-learning-tools-for-enterprises-startup-will-fail-94fb70683834>. Citado 2 vezes nas páginas 27 e 28.

MOOGK, D. R. Minimum viable product and the importance of experimentation in technology startups. *Technology Innovation Management Review*, v. 2, n. 3, 2012. Citado na página 14.

MUKHERJEE, A. *Lean startup and machine learning*. 2018. Blog post at Towards Data Science <https://towardsdatascience.com/lean-startup-and-machine-learning-39a96efecb2a>. Accessed on Nov 2020. Code: TDS-12. Citado 6 vezes nas páginas 27, 28, 29, 30, 31 e 35.

MUKHERJEE, A. *Product definition in the age of AI*. 2018. Blog post at Towards Data Science <https://towardsdatascience.com/product-definition-in-the-age-of-ai-619a417e3415>. Accessed on Nov 2020. Code: TDS-11. Citado 7 vezes nas páginas 27, 28, 29, 30, 31, 35 e 36.

NORI, K.; SWAMINATHAN, N. A framework for software product engineering. In: IEEE. *2006 13th Asia Pacific Software Engineering Conference (APSEC'06)*. [S.l.], 2006. p. 285–292. Citado 2 vezes nas páginas 13 e 15.

PATHA, V. *A Product Manager's Guide to Machine Learning: Balanced Scorecard*. 2020. Blog post at Towards Data Science <https://towardsdatascience.com/a-product-managers-guide-to-machine-learning-balanced-scorecard-344cd37ab4a7>. Accessed on Nov 2020. Code: TDS-07. Citado 8 vezes nas páginas 27, 28, 29, 30, 31, 32, 37 e 38.

PATHAK, P. *A Product Management Framework for Machine Learning*. 2018. Blog post at Medium <https://productcoalition.com/my-framework-for-machine-learning-products-part-1-of-3-2354a0da63d2>. Accessed on Nov 2020. Code: M-06. Citado 4 vezes nas páginas 30, 31, 32 e 34.

PAULK, M. Capability maturity model for software. *Encyclopedia of Software Engineering*, Wiley Online Library, 2002. Citado na página 13.

PETTICREW, M.; ROBERTS, H. *Systematic reviews in the social sciences: A practical guide*. [S.l.]: John Wiley & Sons, 2008. Citado na página 20.

POHL, K. *Requirements engineering: fundamentals, principles, and techniques*. [S.l.]: Springer Publishing Company, Incorporated, 2010. Citado na página 38.

RAGHUWANSHI, R. *Building ML Products: Common pitfalls that product managers should avoid*. 2019. Blog post at Medium <https://medium.com/@rahulraghuwanshi_13455/building-ml-products-common-pitfalls-that-product-managers-should-avoid-14f393d1c250>. Accessed on Nov 2020. Code: M-04. Citado 2 vezes nas páginas 32 e 33.

REBITZER, G. et al. Life cycle assessment: Part 1: Framework, goal and scope definition, inventory analysis, and applications. *Environment international*, Elsevier, v. 30, n. 5, p. 701–720, 2004. Citado na página 14.

REGALADO, R. V. Building artificial intelligent (ai) products that make sense. In: *Proceedings of the 4th International Conference on Human-Computer Interaction and User Experience in Indonesia, CHIuXiD'18*. [S.l.: s.n.], 2018. p. 93–96. Code: ACM-01. Citado 6 vezes nas páginas 11, 28, 29, 30, 34 e 35.

ROSATO, M. Go small for project success. 2018. Citado na página 14.

ROSS, D. T.; GOODENOUGH, J. B.; IRVINE, C. Software engineering: Process, principles, and goals. *Computer*, IEEE, v. 8, n. 5, p. 17–27, 1975. Citado na página 13.

Samuel, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, v. 3, n. 3, p. 210–229, 1959. Citado na página 11.

SCHELTER, S. et al. On challenges in machine learning model management. 2018. Code: SB-06. Citado 12 vezes nas páginas 11, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37 e 38.

SHAMS, R. Developing machine learning products better and faster at startups. *IEEE Engineering Management Review*, IEEE, v. 46, n. 3, p. 36–39, 2018. Code: IEEE-01. Citado 12 vezes nas páginas 16, 27, 28, 30, 31, 32, 33, 34, 35, 36, 37 e 38.

SINGLA, K.; BOSE, J.; NAIK, C. Analysis of software engineering for agile machine learning projects. In: IEEE. *2018 15th IEEE India Council International Conference (INDICON)*. [S.l.], 2018. p. 1–5. Code: SB-03. Citado 10 vezes nas páginas 11, 16, 17, 28, 29, 30, 31, 33, 34 e 35.

SMITH, R. Peer review: a flawed process at the heart of science and journals. *Journal of the royal society of medicine*, SAGE Publications Sage UK: London, England, v. 99, n. 4, p. 178–182, 2006. Citado na página 25.

SYNTHESIZED. *Shipping disruptive ML/AI products at scale (not papers). The engineer's take on scaling ML development processes*. 2020. Blog post at Medium <https://medium.com/@synthesized/shipping-disruptive-ml-ai-products-at-scale-not-papers-4008c6131f12>. Accessed on Nov 2020. Code: M-07. Citado 7 vezes nas páginas 27, 29, 31, 33, 34, 35 e 36.

ULRICH, K. T. *Product design and development*. [S.l.]: Tata McGraw-Hill Education, 2003. Citado na página 14.

WHY do 87% of data science projects never make it into production? 2019. <https://venturebeat.com/2019/07/19/why-do-87-of-data-science-projects-never-make-it-into-production/>. Accessed on Oct 2020. Citado na página 11.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. [S.l.]: ACM, 2014. (EASE '14), p. 38:1–38:10. Citado na página 19.

WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. [S.l.: s.n.], 2014. p. 1–10. Citado na página 23.

ZHANG, D.; TSAI, J. J. Machine learning and software engineering. *Software Quality Journal*, Springer, v. 11, n. 2, p. 87–119, 2003. Citado na página 16.

Anexos

ANEXO A – Position paper from WAIN - ICSE21

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.*