FEDERAL UNIVERSITY OF ALAGOAS
INSTITUTE OF COMPUTING
GRADUATE PROGRAM IN INFORMATICS

Masters dissertation

Brazilian Data Scientists: Revealing their Challenges and Practices on Machine Learning Model Development

João Lucas Marques Correia
jlmc@ic.ufal.br

Advisor:

Baldoino Fonseca dos Santos Neto

Co-advisor:

Rafael Maiani de Mello

MACEIÓ, JUNE OF 2021

João Lucas Marques Correia

Brazilian Data Scientists: Revealing their Challenges and Practices on Machine Learning Model Development

Dissertation presented as a partial requirement for obtaining a Master's degree in the Graduate Program in Informatics at the Institute of Computing, Federal University of Alagoas.

Advisor:

Baldoino Fonseca dos Santos Neto

Co-advisor:

Rafael Maiani de Mello

Maceió, June of 2021

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

C824b Correia, João Lucas Marques.
Brazilian data scientists : revealing their challenges and practices on machine learning model development / João Lucas Marques Correia. – 2021.
60 f. : il.

Orientador: Baldoino Fonseca dos Santos Neto.
Co-orientador: Rafael Maiani de Mello.
Dissertação (mestrado em Informática) - Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2021.

Bibliografia: f. 30-31.
Anexos: f. 32-60.

1. Engenharia de *software*. 2. Aprendizagem de máquina. 3. Estudo empírico. I. Título.

CDU: 004.81:159.953.5

Agradecimentos

Acredito que a gratidão é um sentimento que todos os homens deveriam carregar em si. Por isso, gostaria de expor meus agradecimentos a pessoas importantes. Primeiro gostaria de agradecer a minha avó Maria de Lourdes Correia da Silva (*in memoriam*), que em vida iluminou a vida de tantos, inclusive a minha. O ínfimo passo que dou hoje é resultado dos seus esforços em vida.

Guardo elevada estima e agradecimento a todos os meus irmãos, Junior Correia, Lourdes Neta e Ana Paula Neves por terem me acompanhado por todos estes anos e por sempre me direcionarem pelos melhores caminhos, se hoje finalizo mais essa etapa na minha formação acadêmica, é pelos bons direcionamentos que recebi de vocês. Também sou grato aos meus cunhados Cíntia Vilela e Eliano Junior por todo o acolhimento que nunca hesitaram em me oferecer.

Gostaria de agradecer a minha mãe Ana Marques, por todo o esforço que sempre empenhou para a minha educação. Agradeço a minha namorada Ewelin Costa por ter me acompanhado durante esses anos de pós-graduação. Ao meu amigo Aldo Silva também deixo os meus agradecimentos.

Sou grato ao Prof. Dr. Baldoino Fonseca, por ter me concedido valiosas oportunidades, que me enriqueceram e contribuíram significativamente para a minha formação intelectual. Gostaria de agradecer ao Prof. Dr. Rafael Mello pela co-orientação e todos os direcionamentos científicos que me ofereceu em ótimas conversas. Sou grato ao Prof. Dr. Alessandro Garcia, por todos os direcionamentos e oportunidades que me ofereceu. Também gostaria de registrar meus agradecimentos a todos os pesquisadores que contribuíram com meu trabalho.

Finalmente, gostaria de agradecer a CAPES(117875, 175956, 88882.452283/2019-01, 88887.373933/2019-00, 373892/2019-00), FAPERJ (22520-7/2016) e CNPq (434969/2018-4, 312149/2016-6, 140185/2020-8, 426005/2018-0, 311442/2019-6, 421306/2018-1, 309844/2018-5, 427787/2018-1). Agradeço a ExACTa/PUC-Rio, LES/PUC-Rio, e IBM Research/Brasil por apoiarem os nossos estudos.

Resumo

Cientistas de dados com frequência desenvolvem modelos de aprendizagem de máquina para resolver uma variedade de problemas tanto na indústria como na academia. Para construir esses modelos, estes profissionais executam atividades que também são executadas no ciclo tradicional do desenvolvimento de software, como a elicitação e implementação de requisitos. É factivel argumentar que os cientistas de dados poderiam tirar vantagem dos métodos utilizados pela engenharia de software tradicional para construir modelos de aprendizagem de máquina. Entretanto, o desenvolvimento de código voltado para aprendizagem de máquina possui particularidades que podem levar a desafios que podem que necessitam da adoção de novas práticas de desenvolvimento. De modo que a literatura atual não caracteriza esse tipo de conhecimento do ponto de vista dos cientistas de dados. Neste trabalho, nós caracterizamos os desafios e práticas a respeito da engenharia de modelos de aprendizagem de máquina que merecem atenção da comunidade de pesquisa. Para isto, nós executamos um estudo qualitativo com oito desenvolvedores de software membros de cinco companhias distintas, com diferentes níveis de experiência no desenvolvimento de modelos de aprendizagem de máquina. Nossos achados sugerem que: (i) o processamento de dados e a engenharia de atributos são os estágios de desenvolvimento mais desafiadores durante o desenvolvimento de um modelo de aprendizagem de máquina; (ii) é essencial uma sinergia entre os cientistas de dados e especialistas no domínio da aplicação do modelo; e (iii) o desenvolvimento de modelos de aprendizagem de máquina sofre da falta de suporte de um processo de engenharia bem definido.

Palavras-chave: Engenharia de software, Aprendizagem de Máquina, Praticante, Estudo Empírico.

Abstract

Data scientists often develop machine learning models to solve a variety of problems in the industry and academy. To build these models, these professionals usually perform activities that are also performed in the traditional software development lifecycle, such as eliciting and implementing requirements. One might argue that data scientists could rely on the engineering of traditional software development to build machine learning models. However, machine learning development presents certain characteristics, which may raise challenges that lead to the need for adopting new practices. The literature lacks in characterizing this knowledge from the perspective of the data scientists. In this work, we characterize challenges and practices addressing the engineering of machine learning models that deserve attention from the research community. To this end, we performed a qualitative study with eight data scientists across five different companies having different levels of experience in developing machine learning models. Our findings suggest that: (i) data processing and feature engineering are the most challenging stages in the development of machine learning models; (ii) it is essential synergy between data scientists and domain experts in most of stages; and (iii) the development of machine learning models lacks the support of a well-engineered process.

Keywords: Software Engineering, Machine Learning, Practitioner, Empirical Study.

List of Figures

List of Tables

Contents

1 Introduction

In this Chapter, we present a summary of the research, starting with the context and problem, connecting them with the objectives and contributions of this work.

1.1 Context and Problem

The adoption of Machine learning (ML) models has been growing as the intelligence behind software systems. These models are used for solving specialized problems in several domains. In the oil and gas industry, ML models are used for mitigating environmental disasters (ONAWOLE et al., 2018; BAGHBAN et al., 2016). Governments have been using ML models for monitoring socioeconomic development (PISCOPO; SIEBES; HARDMAN, 2017). In the Ecology domain, ML models have been used to classify animal species (WÄLDCHEN; MÄDER, 2018; TABAK et al., 2019).

ML models are developed for allowing programs to learn from previous experiences (MITCHELL, 1997). In this context, professionals working with ML modeling should define appropriate resources to train their models (e.g., data samples, features, algorithms, and parameters), optimizing the learning from different perspectives, and obtaining the model that best fits the desired solution. Typically, the professionals allocated to conduct this development are the data scientists (PATIL, 2011). From these professionals, it is required a multidisciplinary knowledge, including but not limited to data management, mathematics, and software engineering. Besides, they also often need to interact with customers and domain experts to understand the scope of the problem to be solved.

Data scientists may be served from different strategies to develop an ML model depending on the practical problem to be addressed. For instance, if the model's clarity and communication are the most important, it would require data scientists to use interpretable ML models (e.g., decision tree) independent of its accuracy. Otherwise, if the model's accuracy is more important, it would require data scientists to focus on tuning model parameters.

One might argue that data scientists would benefit from adopting classical software engineering disciplines (e.g., systems design, quality assurance, and verification) to properly build their models. However, ML modeling addresses a distinguished development paradigm, requiring proper tools and techniques (MASUDA et al., 2018; WAN et al., 2019).

The lack of this support may lead data scientists, frequently having a limited background in software engineering, to improvise and frequently perform *ad hoc* activities to overcome the particular challenges involved in engineering ML models. For example, the validation and verification process of stochastic models is challenging. Current tools and engineering disciplines do not fully support the verification and validation of code with random behavior (ZHANG et al., 2020).

1.2 Objectives and Methodological Aspects

In this context, we investigate the challenges and practices that emerge from the development of ML models from the perspective of data scientists. We focus our analysis on well-know stages present in the ML workflow (Hill et al., 2016; PATEL et al., 2008; AMERSHI et al., 2019): *Model Requirement*, *Data Collection*, *Data Cleaning*, *Data Labelling*, *Feature Engineering*, *Model Training*, *Model Evaluation*, and *Model Deployment*. More specifically, we investigate which stages are considered more challenging by data scientists, and which are the common practices adopted by them to deal with the corresponding challenges.

To perform our study, we conducted individual semi-structured interviews (COHEN; CRABTREE, 2006) with data scientists from different Brazilian companies. These professionals are experienced in developing back-box and white-box ML models in three main domains: oil and gas, government, and natural resources. We transcribed the interviews and applied the open-coding methodology (STRAUSS; CORBIN, 1998). Then, we grouped the resulting codes into five categories: *actor, activity, method, limitation, and challenge*. We rely on the categories' limitations and challenges to point out problems and issues faced nowadays by data scientists that deserve attention from the research community. Then, we rely on the categories activity and method to understand common practices followed by data scientists when developing ML models.

1.3 Contributions

The findings of our study indicate that data scientists perceive the *Data Processing* and *Feature Engineering* as the more challenging stages of ML model development. However, data scientists also reported important issues addressing other stages. In general, the chal-

lenges reported indicate the need to enhance the interaction and cooperation between data scientists and the other actors involved in that stage's development, including domain experts, customers, and project managers. Besides, we also found that the practice of ML development lacks the support of a engineered process. For instance, we did not find in the interviews mentions about techniques for assuring the traceability between the features and the model requirements. Also, the validation of the ML model is often skipped due to the difficulty to test back-box ML models. These characteristics reflect on the recurrent rework and on the strong and continuous dependence of data scientists to the domain experts. Thus, we understand that the findings of our study may support future research on designing a comprehensive and engineered process for supporting data scientists on developing ML models.

1.4 Thesis Structure

The remainder of the work is organized as follows:

- Chapter 2 introduces background concepts and discusses the related work.
- Chapter 3 introduces our research questions and methodological steps.
- Chapter 4 describes the execution of our methodology.
- Chapter 5 and Chapter 6 describe our findings and discuss results, respectively.
- Chapter 7 describes threats to validity.
- Chapter 8 presents our conclusion which could lead to future contributions.

2 Related Work

Recent studies (WAN et al., 2019; AMERSHI et al., 2019; NGUYEN-DUC et al., 2020; ZHANG et al., 2019; MASUDA et al., 2018) have indicated a difference between the challenges faced by ML practitioners (in our research, data scientists) and non-practitioners.

Nguyen et al. (NGUYEN-DUC et al., 2020) reported an exploratory study across seven companies. They investigated how software engineering processes and practices can be applied to develop systems based on Artificial Intelligence (AI). Their findings revealed that particularities of AI-based applications, such as the uncertainty of predictions, hinder the adoption of traditional software engineering guidelines during the development of those applications, showing different development approaches. In this context, our study focuses on practices and challenges faced by data scientists in ML models development. Besides, we explore companies with development goals driven by business and/or research.

Amershi et al. (AMERSHI et al., 2019) reported a study involving AI professionals at Microsoft. The authors investigated scientists, researchers, managers, programmers, and other professionals in their daily activities. They observed three major challenges in building large-scale AI applications: *data management*, *reuse*, and *modularity*. Amershi et al. (AMERSHI et al., 2019) concluded that building AI systems require more effort and expertise from professionals. In this sense, our study aims to investigate in-depth the efforts of data scientists professionals in ML from five different companies and three different domains.

Wan et al. (WAN et al., 2019) investigated how the adoption of ML frameworks affects software engineering practices. The authors performed an empirical study through interviews and a survey. They showed statistically significant differences in software engineering practices and teamwork characteristics. They suggested that most of the differences come from the uncertainty present in the inherent randomness of the data and ML algorithms. Notice that the authors explore frameworks employed for ML, while our study examines the whole development process.

Nascimento et al. (NASCIMENTO et al., 2019) interviewed seven developers from three small software companies placed in Brazil working with ML-based systems development. They aimed to investigate software processes and challenges faced by developers. Their findings pointed that companies follow a four-step software process: understanding

the problem, data handling, model building, and model monitoring. In addition, the challenges faced by developers were: identifying the clients' business metrics, lack of a defined development process, and designing the database structure. In our work, we also conduct investigations inside Brazilian companies. However, we have a more specific focus since we investigate companies that work with the development of ML models, not fully ML-based systems. Additionally, we also proceed with our study from the perspective of data scientists, not developers in general.

We found recent studies (MASUDA et al., 2018; ZHANG et al., 2020) investigating and identifying state-of-the-art approaches for supporting the engineering of ML systems. Masuda et al. (MASUDA et al., 2018) conducted a survey to discover software engineering approaches to support the development and assure the quality of ML systems. They analyzed 78 research papers and pointed out that existing software engineering practices may be inappropriate to deal with ML's uncertainty. Zhang et al. (ZHANG et al., 2020) investigated 138 research papers looking for approaches to test and debug the ML code. Their findings indicated that only a few contributions focus on testing interpretability, privacy, or efficiency. Zhang et al. (ZHANG et al., 2020) focus on analyzing exclusively the stage of *Model Evaluation*, instead we target all ML stages.

All these studies (WAN et al., 2019; AMERSHI et al., 2019; NASCIMENTO et al., 2019; NGUYEN-DUC et al., 2020; ZHANG et al., 2019; MASUDA et al., 2018) highlight the need for improving the ML development process. In this context, a few recent works (AMERSHI et al., 2019; BYRNE, 2017; KIM et al., 2017; POLYZOTIS et al., 2018) have been proposing ML workflows to guide ML developers. These workflows are composed of typical stages addressing the development of ML models. Among them, the workflow proposed by Amershi et al. (AMERSHI et al., 2019) is the most recent work found in the literature, presenting the most comprehensive and up to date workflow. This workflow is composed of the following stages:

- *Model requirements.* This stage establishes the basis for an agreement between stakeholders about how the ML model should work. In this stage, stakeholders decide which data to consider and what types of ML models are most appropriate for the given problem.

- *Data processing.* This stage involves collecting all relevant data; cleaning biased and irrelevant data; and data labelling (for supervised learning).
  - *Data collection*. This stage involves collect relevant data to the problem to be addressed.
  - *Data cleaning*. In this stage, wrong or irrelevant data is removed from dataset.
  - *Data labelling*. This stage involves label the data, when working with supervised learning.
- *Feature engineering.* This stage covers the process of modifying the selected data (e.g., by encoding features and extracting new features) in order to better suit the particularities of the chosen model and improve its accuracy.
- *Model training.* In this stage, the ML model chosen is trained and tuned on the (labeled) data.
- *Model evaluation.* In this stage, metrics are used to evaluate the created model on new (non-labeled) data.
- *Model deployment*. This stage covers the deployment of the model in the customer environment and the activities performed for monitoring and maintaining the model.

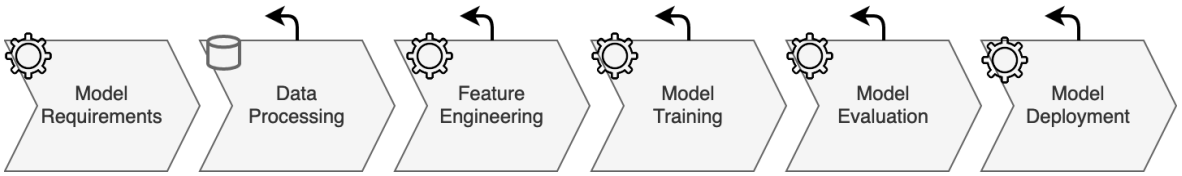

Figure 1 – Machine Learning workflow. Adapted from (AMERSHI et al., 2019).

In this work, we used the Amershi et al. workflow (Figure 1) as a common reference to understand the ML modeling process followed by the data scientists interviewed. For this propose, we introduced the workflow to the data scientist at the beginning of each interview, asking him/her for identifying to what extent the workflow addresses his/her developing practice (see Chapter 4). The arrows above each stage in Figure 1 indicates the possibility of a *feedback loop* across the stages.

As explained above, to understanding and classifying code harmfulness is essential to identify pieces of code that were really harmful to the software, helping developers prioritize them while refactoring the code. Based on that, we set the following objectives.

3 Study Design

Our study aims at characterizing the main challenges faced by data scientists on developing ML models as well as the practices adopted by them to deal with these challenges. More specifically, we address the following research questions:

RQ_1 *What are the most challenging stages faced by data scientists on developing ML models?* The motivation of this question is to comprehend the Machine Learning development process followed by data scientists, using as a starting point the workflow discussed in Chapter 2. In addition, we aim to investigate which stages and activities composing their workflow introduce more challenges, besides the aspects of challenges.

RQ_2 *What are the practices adopted by data scientists to deal with the most challenging stages?* The motivation of this question is to investigate practices that data scientists employ to deal with the most challenging stages. In addition, we aim to figure out if data scientists can overcome the challenges present in such stages. For practices, we mean methods that data scientists employ to overcome the challenges, tools they use, help of external actors, and limitations.

To answer these questions, we conducted an exploratory study based on semi-structured interviews. By challenging stages, we mean the stages with complex, time-consuming, and error-prone activities. By practice, we mean the activities performed by data scientists, as well as the tools and methods they use to perform these activities.

To address RQ_1, we started the interviews by stimulating data scientists to reflect on the stages of the ML workflow (see Figure 1). Then, we asked data scientists to objectively indicate which stages they consider more challenging and why. To address RQ_2, we applied open questions to stimulate the participant to describe in detail how they perform the more challenging stages. After coding all the answers given by the data scientists, we identified code categories and distributed the coded data among these categories.

Target population and sample. This study's target population comprises experienced data scientists in developing ML models for complex and customer-oriented solutions. These models may be supervised or not. We recruited a small but diverse sample of eight data

scientists from five different companies (see Table 1). The participants' experience with ML ranges from two to 40 years. Besides, we selected participants working in three different domains within our industrial collaboration network: three data scientists ($d1$, $d2$, $d4$) from the oil and gas domain; three data scientists ($d6$, $d7$, $d8$) of natural resources; and two ($d3$, $d5$) data scientists working for the government. Although we do not directly explore projects in the academy, six of the eight data scientists ($d3$, $d4$, $d5$, $d6$, $d7$, $d8$) are postgraduate students, having experience in using ML models for research. The frameworks for developing ML models frequently employed by the data scientists are TensorFlow [1], Keras [2], Scikit-learn [3], PyTorch [4].

Instrumentation. All the authors of this study were involved in the design and validation of the interview questions. Two authors described the interview questions and protocol, and the others validated it. At the beginning of each interview, the interviewer introduced to the data scientist the workflow that we centered our research (Figure 1). Then, we asked the participant to report any difference in this workflow with his/her practical experience. Once we focus on the high-level stages, data scientists did not have any disagreement with the workflow or with the terminologies used. Next, we applied *general questions* and *specific questions* regarding each stage of the workflow, as described in Listing 1. The interview' *general questions* aim at exploring the background of the data scientist, the role of people in his/her team, and how data scientists execute and verify activities in a given stage. Besides, we also ask the data scientists which workflow stages they perceive as most challenging. Otherwise, the *specific questions* explore aim to explore in more depth the activities performed by the development teams in the most challenging stage(s). With this, we aim to (1) optimizing the interviews' time, limited to 1 hour, and (2) gathering more reliable data for supporting our analysis.

The complete form with all the specific and general questions is available in our supplementary material (J, 2020). Notice that a few additional questions appeared according to the flow of each interview.

[1] https://www.tensorflow.org/
[2] https://keras.io/
[3] https://scikit-learn.org/
[4] https://pytorch.org/

Table 1 – Characterization of the data scientists.

Company	Projects Context	Data Scientist	Experience	Technical Background		Frameworks
$c1$	Oil and gas	$d1$	2 years	Supervised Learning	Image Detection	TensorFlow Keras
		$d2$	40 years	Supervised Learning	Image Detection	TensorFlow Keras
$c2$	Government	$d3$	12 years	Supervised Learning	Discriminative models	TensorFlow Keras Scikit-learn
$c3$	Oil and gas	$d4$	4 years	Supervised Learning	Discriminative models	Scikit-Learn
$c4$	Government	$d5$	3 years	Unsupervised Learning	Clustering	Scikit-Learn
$c5$	Natural resources	$d6$	4 years	Supervised Learning	Knowledge Representation	TensorFlow Scikit-Learn PyTorch
		$d7$	4 years	Supervised Learning	Image Detection	TensorFlow
		$d8$	12 years	Supervised Learning Unsupervised Learning	Image Detection	TensorFlow Scikit-Learn

List 1: General and Specific Questions.

1. **General Questions**

    1.1 Background information

    1.2 Who is involved at each stage

    1.3 Activities to accomplish a stage

    1.4 Stage(s) which the data scientist has more expertise[1]

    1.5 Most challenging stage(s)

2. **Specific Questions**

    2.1. *Model Requirements*

        2.1.1. Requirements specification

        2.1.2. Functional and nonfunctional requirements

        2.1.3. Completeness, correctness and testability

        2.1.4. Verification and migration

    2.2. *Data Processing*

        2.2.1. Data Collection

        2.2.2. Data Cleaning

        2.2.3. Data Labeling[2]

    2.3. *Feature Engineering*

        2.3.1. Feature selection and transformations

        2.3.2. Importance of an expert in this stage

    2.4. *Model Training*

[1] To measure the expertise, we consider the number of projects or the time the practitioner worked on that stage.

[2] Only for supervised learning.

4 Execution

The eight interviews were taken between November 2019 and April 2020. Each interview took, on average, 45 minutes. To be able to collect different viewpoints and perspectives, two researchers conducted each interview. One researcher played the role of the main interviewer, applying the planned questions. Another researcher predominantly played the role of observer, taking notes about the participants' behaviors and asking additional questions to gather more information.

4.1 Context of the Answers Given

All the study participants found correspondence between the workflow introduced by the interviewers (Figure 1) and their practice, confirming that their development activities follow the same flow: *definition of model requirements*, *data processing*, *feature engineering*, *model training*, *model evaluation*, and finally *model deployment*. Besides, most of the participants

argue that several iterations may happen during the development of ML models, mainly due to issues lately identified. For instance, data scientist $d4$ describes cases in which was necessary to go back in earlier stages of the workflow.

> *"Very often during model training we identify problems having to return to the stage of feature engineering. Similarly, it also occurs in the stage of model evaluation, once the resulting model presents bad accuracy we have either to reprocess the data or reengineer the features." - (Data scientist $d4$)*

Our interview was composed of general and specific questions (see Chapter 3). We applied the general questions to all stages where the data scientist was experienced, and the specific ones to the most challenging stage(s). Table 2 presents the distribution of the general and specific questions answered by the data scientists.

One can see that none of the data scientists answered general questions about *model requirements* once they are not experienced in this stage. Besides, they did not classify this stage as challenging. However, it does not mean that the participants not provided information about this stage. For instance, $d8$ described how requirements for a model arises during the model idealization.

> *"I think the way we process the data and engineer the features comes a lot from the customer (the stage of Model Requirements). What does the customer want? What task would he like to solve applying ML? What is the problem he is facing [...] I think there is a lot of customers' input here." - (Data scientist $d8$)*

Table 2 – Distribution of the data scientists' answers by stage.

Stage	**General Questions**	**Specific Questions**
Model Requirements		
Data Processing	$d1$,$d2$,$d3$,$d4$,$d8$	$d1$,$d2$,$d3$,$d4$
Feature Engineering	$d2$,$d3$,$d4$,$d6$,$d7$	$d2$,$d3$,$d4$,$d8$
Model Training	$d1$,$d5$,$d6$,$d7$	$d5$,$d6$,$d7$
Model Evaluation	$d6$,$d7$	
Model Deployment	$d5$	

4.2 Generated Codes

During data analysis, the first author performed the transcription of the recorded interviews. Then, he conducted the *open-coding* process (STRAUSS; CORBIN, 1998) of each quotation (raw transcription) to support the analysis. The open-coding process, was executed as illustrated in the example that follows:

> **Raw Transcription:** *"Geologists and geophysicists bring knowledge in geology and geophysics. They often warn that data-driven solutions, merely based on data, are minor problems once there is a whole physics background supporting this data."*

Table 3 – Example of codes from the raw transcription.

Code	Content
Code 1:	domain expert presence
Code 2:	domain expert indicates rules that must be respected
Code 3:	data-driven solutions are not always possible
Code 4:	the model must respect domain particularities

After coding the whole raw data (see example in Table 3), the first author analyzed the codes, aiming at identifying an initial set of categories for grouping these codes. After discussions and refinements on initial codes and categories, all authors reached a consensus for establishing the final set of categories. Then, the first author redistributed the codes into the new set of categories. Next, another author who did not attend the interviews, validated the codes and their corresponding categories. They disagreed in 43 of 447 codes (9.6%). Then, we allocated a third author for solving the points of disagreement. At the end of the process, it has emerged the following categories:

- *actor*: person or team involved in a stage;
- *activity*: process or task performed by an actor;
- *method*: tools or methodology used by the actor;
- *limitation*: expected behaviors or activity limitations;
- *challenge*: challenges faced by an actor.

In particular, codes from Table 3 were classified in the following categories: Code 1 $\rightarrow$ *actor*; Code 2 $\rightarrow$ *activity*; Code 3 $\rightarrow$ *limitation*; Code 4 $\rightarrow$ *limitation*.

In total, the analysis for all interview data resulted in 447 codes, distributed into the five categories: actor (33); activity (119); method (53); limitation (152); and challenge (90). With the aim to access a diverse number of interviews, in the analysis process the authors ponder three factors:

1. The codes convergence to a similar main set of categories.
2. The amount of data to answer our research questions (i.e., did all available interviews cover our research questions?)
3. The availability (1-hour interviews) and access (industry partnership) to interviewees.

To provide some rationale about the context of the codes, Table 4 shows examples of code found by category. Our analysis shows that even the answers of the most and the least experienced data scientists, respectively $d2$ and $d1$, do not diverge from the central topics of discussion addressed by other interviews. All codings and interview transcriptions are available at our supplementary material in Appendix A (only in Portuguese).

After establishing the categories and distributing the codes among them, we mapped each code with its corresponding workflow stage (Figure 1). As an example, Table 5 describes the main actors (data scientist, domain expert, customer, project manager and infrastructure team) involved in each stage. Codes without a clear and specific association with the workflow stages was classified as *general*.

Table 4 – Examples of the categories emerged.

Category	**Example of relevant codes**
Actor	Presence of the client in Model Evaluation
Activity	Data scientist define hyperparameters according to his experience
Method	Use of framework for model development
Limitation	One data scientist responsible for all stages
Challenge	Uncertainty about model quality for real scenarios

Table 5 – Participation of actors by stage from the perspective of the data scientists.

	Data scientist	Domain expert	Customer	Project manager	Infrastructure team
Model Requirements	$d2$,$d3$,$d4$,$d6$,$d7$,$d8$	$d6$	$d5$,$d4$,$d6$,$d7$,$d8$	$d5$	
Data Processing	$d1$,$d2$,$d3$,$d4$	$d1$,$d3$,$d4$,$d6$,$d7$	$d8$		
Feature Engineering	$d1$,$d2$,$d3$,$d4$,$d5$,$d6$,$d7$,$d8$	$d1$,$d3$,$d4$,$d6$,$d7$,$d8$			
Model Training	$d1$,$d2$,$d3$,$d4$,$d5$,$d6$,$d7$,$d8$				
Model Evaluation	$d1$,$d2$,$d3$,$d4$,$d5$,$d6$,$d7$,$d8$		$d2$,$d4$,$d6$,$d7$	$d5$	
Model Deployment	$d5$,$d6$				$d4$,$d6$,$d8$

5 Revealing Challenges and Practices

In this Chapter, we answer the research questions of the study, revealing the challenging stages and common practices in the ML model development.

5.1 RQ_1: Most Challenging Stages

To answer this research question, we benefit from two distinct moments First, we analyzed the number of data scientists that considered each stage as more challenging. Second, we analyzed the distribution of codes categorized as challenges in each stage during the *open coding* process. In our classification, we consider the number of data scientists indicating a stage as challenging more important than the number of challenging codes per stage, since individual interviews may unbalance the code distribution. Table 6 shows the stages analyzed in our study, the number of participants that reported the stage as challenging, and the number of codes in the category *Challenge* associated with each stage.

We found that *Data Processing* and *Feature Engineering* are the two stages more frequently perceived as challenging. Six and seven participants reported these stages as challenging, respectively. Besides, we also found that the *Data Processing* and *Feature Engineering* are the first and third stages with more challenge codes associated. While the *Data Processing* challenges are more related to the difficulties in choosing proper data to compose the dataset, the *Feature Engineering* challenges are frequently associated with the management of features for training the model.

Table 6 – Distribution of the challenges reported.

Stage	Data Scientists (Total)	Challenges Reported
Model Requirement		0
Data Processing	$d1$,$d2$,$d3$,$d4$,$d5$,$d8$ (6)	19
Feature Engineering	$d2$,$d3$,$d4$,$d5$,$d6$,$d7$,$d8$ (7)	23
Model Training	$d1$,$d5$,$d6$,$d7$,$d8$ (5)	22
Model Evaluation	$d2$,$d4$,$d5$,$d6$,$d7$ (5)	12
Model Deployment	$d5$ (1)	2

Finding 1: *Data Processing* and *Feature Engineering* are perceived as the most challenging stages.

Besides, five participants reported the *Model Training* and *Model Evaluation* as challenging. While the challenges in *Model Training* has 22 codes associated, *Model Evaluation* has only 12 codes. In *Model Training*, the challenges frequently address concerns with time-consumption, the definition of artifacts such as parameters and algorithms, and the verification of the fitness of the chosen artifacts. The *Model Evaluation* challenges predominantly address the definition of the best metrics for evaluation and the model's quality assurance.

A single data scientist reported the *Model Deployment* as a challenging stage. He reported two challenges regarding the need to master specific technologies during deployment. To overcome this challenge, we observed that an infrastructure team frequently conducts the model deployment, as reported by data scientists $d4$, $d6$, $d8$ (see Table 5). Finally, none of the participants reported the *Model Requirements* as one of the most challenging stages. It was an expected result once although most of the data scientists recognized their presence in this stage (Table 5), most of them reported low experience on this stage.

5.2 RQ_2: Practices Adopted by Data Scientists

In the previous Chapter, we identified that the *Data Processing* and *Feature Engineering* are the most challenging stages from the data scientists' perspective. Now, we identify how data scientists deal with these stages. To do so, we analyze the actors, activities, methods and limitations related to these stages.

5.2.1 Data Processing

Actors. The data processing involves three distinct actors: *data scientist*, *domain expert*, and *customer*, as described in Table 5. The data scientist is the professional involved in all the activities related to the ML model development. The customer is a representative of the company or professional aware of the company's interests. Finally, the domain expert is a professional having sufficient scientific knowledge in the field of the application. For example, we observed development teams containing electrical engineers and geologists.

Depending on the availability of data sources, there are different approaches to perform *Data Processing*. For instance, if the company has the data required to build an intelligent model, it is not necessary external sources and, therefore, data processing can be performed in-house (at the company).

The data processing in-house may be interesting, mainly in situations in which the company contains confidential content that cannot be shared, or even when the company has a large number of data stored in different ways, and not all data is relevant to build the model. In both cases, the data scientists need the support of the customer for data gathering, since they may not have access to the confidential content. Besides, in some cases, they may not be aware of the company's internal data infrastructure as well as the most pertinent data. Thus, the conduction of data processing would be very costly. In any case, the interaction and cooperation between the data scientists and customers is needed to accomplish data processing activities, as described in the quotation below:

> *"The Data Processing stage ends up requiring a lot of inputs from the customer. Although the customer give us some instructions to collect the data, often the data are not in the right format (or in a good shape) for us to proceed, thus we often need several interactions to get the desired data." - (Data scientist $d8$)*

On the other hand, external data sources are used when the customer does not have the required data to build an intelligent model. In such cases, data processing is conducted by the data scientist and domain expert. The domain expert plays a crucial role, analyzing which data is relevant to meet the requirements. As we will see forward, one of the most challenges from data scientists is the demand for domain knowledge, which is mitigated by the participation of the domain expert in the activities. For instance, the subject $d3$ report difficulties in the data processing:

> *"Very often you will have to get the data from different sources and extract the relationship behind it, thus you will need to use your own conventions for that." - (Data scientist $d3$)*

Activities. Our results indicate that the first challenge activity of this stage is the *instance labeling* by the domain expert. In supervised models, the data scientist must provide a sam-

ple of instances and their labels to the training process. In more complex domains, such as natural resources, the labeling activity is not trivial, requiring several and detailed dataset analysis. For instance, the following statement was made by a data scientist working with image classification, he recognizes that he cannot label its instances due to the lack of expertise in the domain.

> *"I always need to label new images to train the classification model, however very often I do not know what they mean, so I heavily depend on a specialist who will annotate the images for me." - (Data scientist d1)*

Another challenging activity of data processing reported by data scientists is the *data enrichment*. Data scientists tend to perform data enrichment when they recognize that the number of data available lacks size and/or quality. We observe that this scenario is common in the context of projects involving image processing. In these cases, the data scientists reported that the small datasets lead them to artificially creating new images through techniques such as rotation, contrast increasing and noise addition.

Method. The unique data processing method reported by data scientists was the use of charts, such as box plots and histograms, for supporting the verification of data quality. The motivation behind the adoption of charts addresses the need for visual tools to avoid the use of inappropriate data. For instance, if a data scientist identifies an error in the dataset only further in the *Feature Engineering* stage, he/she needs to re-execute *Data Processing*, which implies in higher development costs.

Limitation. We found one main limitation for data processing; the data scientists depend on the domain expert and the customer. As shown in our results, data scientists believe that data processing considerably affects the next workflow stages. Therefore, aspects such as gathering relevant data, label data correctly, data quantity, and quality assurance should be well performed in this stage to mitigate the impact on the following stages.

Finding 2: The cooperation between data scientists, domain experts, and customers during data processing is essential to perform data collection, data enrichment, and instance

labeling. In this stage, charts (e.g., histograms and boxplots) are commonly used for the verification of data quality.

5.2.2 Feature Engineering

Actors. In this stage, our results indicate the involvement of two actors: the data scientist and the domain expert (see Table 5). As previously reported, the data scientist is involved in almost all stages once he/she is responsible for building the entire intelligent model. The domain expert assists data scientists to better understand complex features composing the dataset.

Activity. Feature Engineering addresses preparing features and selecting those most relevant for the model training. Our results identify three activities at this stage. The first one is the *feature analysis*, activity in which data scientists explore the whole dataset aiming at characterizing feature's parameters such as value range, correlation, distributions, and independence degree.

The second activity is *data transformation*. It consists of executing operations over features for removing inadequate values, increasing representativeness, and converting types and values. For instance, neural network algorithms require numeral features with values in the range between zero and one. Thus, it may be necessary to convert the data type and value of some features. One special case of transformation is the feature combination, in which two or more features are combined into a new one more valuable for the learning process.

Data transformations are usually performed only by data scientists, except by the feature combination, typically performed by the data scientist with the assistance of the domain expert. It can be explained by the fact that identifying opportunities and performing these transformations requires understanding the semantics of features used and resulted from transformations.

The next activity is the *feature selection*. It identifies the best features from the entire dataset for training the model. Previously mentioned activities influence directly in the quality of feature selection, since observation in *feature analysis* and *data transformation* will impact the data scientists' judgment regarding the best features to train the model. The

feature selection is conducted by the data scientist and the domain expert, since each one cumulate knowledge about features. We observe in our results participants saying that in some cases, the domain expert reveals excellent features according to his/her expertise. However, data scientists noticed some cases that only their feature analysis without the support of a domain expert revealed excellent features (not noticed by domain experts):

> *"For example, although the expert claims a feature is not that important, [...] after trying out the model with the feature it may turn out to be very valuable. Also, the opposite may happen. The algorithm shows us that a specific feature is not that important, but [...] it turns out to be very important for the domain expert, and we cannot simply ignore it." - (Data scientist d4)*

Methods. Our codings revealed two main methods for performing the *Feature Engineering*: the use of *statistical methods* in *data analysis*, and the use of *automatic feature selectors* in *feature selection*.

Statistical methods were mentioned as being widely used to assist the *data analysis* process, since they provide functions and tools for helping data scientists on observing the data behavior. Data scientists did not mention specific tools for data analysis. However, they reported the use of the Python language and its libraries, e.g. pandas [3], to run statistical operations.

According to the participants, the use of automated feature selector is associated with deep learning, because algorithms for this propose automatically learns the best features to problem solution and model training, discarding the need of the data scientists for performing this activity:

> *"It is worth mentioning that in today's deep learning era, we skip this stage (Feature Engineering), once we have a model that learns the features, and also learns the task." - (Data scientist d8)*

On the other hand, when algorithms for deep learning are not used, the developer performs the feature selection manually. In this case, the role of the domain expert is essential once he/she will point out the appropriate features to train the model based on ground truth.

[3] https://pandas.pydata.org/

Although the domain expert knows the relevant features of the problem, the data scientist must execute operations such as feature scoring to ranking features based on relevance. It should happen once the building of ML models is an exploratory process, which may reveal new patterns or behaviors even not know previously by experts.

Limitations. Our findings show that limitations in this stage are related strictly to the need for understanding about the application domain, for this reason the domain expert is so essential:

> *"Feature engineering is a very complicated process. First, you have to understand the (application) domain. After you understand this domain, you should identify those aspects that will make the statistical model more efficient in understanding the data representation." - (Data scientist d6)*

When data scientists talk about the *feature selection*, they frequently report being not sure on whether selected features are the best options to train the model, mentioning that the stage that indicates this feature adequacy is the model evaluation. Therefore, it suggests that at this point, any mechanism grants confidence to the data scientist about their work on *Feature Engineering*. The next quote exposes how data scientists deal with the correctness of *Feature Engineering*:

> *"Very often, we have to turn again to the stage of Feature Engineering, especially when we are dealing with data that we do not know very well [...] We first evaluate the model. If the accuracy is not good enough, we have to make changes in how the features are being considered to improve accuracy." - (Data scientist d7)*

Finding 3: Data scientists and domain experts are the main actors involved in feature engineering, performing feature analysis, data transformation, and feature selection. Automatic feature selection is the most common method at this stage, but data scientists still need the domain expert to better understand the application domain.

6 Discussion

Table 7 describes the central topics of discussion for each stage of the workflow of Figure 1. The table also presents the frequency of the central topics in each stage grouped by data scientists' experience with ML models (low experience and high experience). We consider as having low experience those data scientists with less than five years of experience in ML development. The data scientists with high experience are those having more than five years of experience.

First, we group similar topics from all coding referring to a single stage. Then, we produced a list of the most discussed topics. Table 7 presents the top-3 hot topics (except for the stage *Model Requirements* which had an insignificant number of discussions for a single another topic). Second, for all listed topics referring to each stage, we compute how often these topics are discussed by low and high experienced data scientists. As an example, notice in the first row of the table that the central topic discussed in the stage of *Model Requirements* was the *"Presence of internal and external actors"*. We observe that 67% of the comments were about this topic. From these comments, 58% were made by low experienced data scientists, while the others were made data scientists with high experience in ML development. Considering this summary, three things are worth noticing:

(1) Presence of internal and external actors. The presence of internal and external actors is essential in most of the stages, except for *Model Training*. In *Model Training*, data scientists write code, optimize model parameters, and train the model. Thus, this stage requires skills in ML algorithms, hence only data scientists act in this stage. However, for the remaining stages, we observe the need for synergy between the data scientists and the external actors.

(2) Challenging stages. The problems of the challenging stages address the need to understand the domain area of the ML model. Both most challenging stages (*Data Processing* and *Feature Engineering*) directly depend on the quality of the requirement specification, which strongly relies on the knowledge of external actors. Therefore, data scientists need to properly extract knowledge of the domain and the external actors' needs to perform a good requirements specification. However, data scientists often associate the need for this interaction with problems. For instance, the quotations below illustrate the discussion about the stage of *Data Processing* with the data scientist $d1$:

Table 7 – Central topics of discussion per stage, with the frequencies of discussion by data scientists' experience.

Stage	Central topics (frequency of discussion)		Experience	
			Low	High
Model Requirements	Presence of internal and external actors	(67%)	58%	42%
	Others	(33%)	($d4$, $d6$, $d7$)	($d3$, $d8$)
Data Processing	Data processing as problematic	(42%)		
	Data scientist has difficulties to assess data quality	(21%)	46%	54%
	Presence of internal and external actors	(18%)	($d1$, $d4$, $d5$, $d6$, $d7$)	($d2$, $d3$, $d8$)
	Others	(19%)		
Feature Engineering	Feature engineering as problematic	(22%)		
	Use of classic techniques are not enough	(21%)	45%	55%
	Presence of internal and external actors	(20%)	($d4$, $d5$, $d6$, $d7$)	($d2$, $d3$, $d8$)
	Others	(37%)		
Model Training	Training as problematic	(26%)		
	Empirically chosen algorithms and hyperparameters	(25%)	90%	10%
	Use of framework for model development	(22%)	($d1$, $d4$, $d5$, $d6$, $d7$)	($d2$, $d3$, $d8$)
	Others	(27%)		
Model Evaluation	Process of choosing a metric for model validation	(41%)		
	External actors also evaluates the model	(19%)	76%	24%
	Model accuracy	(14%)	($d1$, $d4$, $d5$, $d6$, $d7$)	($d2$, $d3$)
	Others	(26%)		
Model Deployment	External actor presence	(41%)		
	Internal actor presence	(18%)	92%	8%
	Data Scientists has difficulties to deploy the model	(18%)	($d4$, $d5$, $d6$, $d7$)	($d8$)
	Others	(23%)		

- *"Poor data implies in non-accurate model"*
- *"Difficulty in ensuring the correctness of the dataset"*
- *"We are unable to work without the support of external actors"*

The quotations below illustrate the discussion about the stage of *Feature Engineering* with the data scientist $d4$:

- *"Difficulty in identifying suitable transformations for the data"*
- *"The stage of feature engineering impacts the other stages"*
- *"Uncertainty whether the chosen attributes are the most suitable for the model"*

Although *Model Training* does not directly depend on external actors, it was considered mainly by less experienced data scientists (90%) as problematic due to the uncertainly about the quality of the model. The quotations below illustrate the discussion about the stage of *Model Training* with the data scientist $d7$:

- *"Uncertainty whether the training data represents real data"*
- *"Uncertainty whether the model will be good in practice"*
- *"Random characteristic of the model makes its verification difficult"*

Finally, data scientists also faced problems in the stages of *Model Evaluation* and *Model Deployment*. More exactly, they reported difficulties in choosing appropriate and sufficient metrics for evaluation:

- ($d4$) *"Overfitting is often observed"*
- ($d6$) *"Classic metrics are usually not enough"*
- ($d7$) *"Uncertainty about whether the chosen metrics are the most appropriated."*

Notice that in *Model Training* and *Model Evaluation* even though data scientists mentioned problems, they discussed about the general aspects of the use of metrics and model accuracy. We describe a few problems with *Model Deployment*.

- ($d5$) *"Difficulty to deploy the model due to business rules"*
- ($d4$, $d6$) *"Data Scientists depend on the customer's infrastructure team"*

These study findings suggest that all stages are somehow problematic, mainly due to expert knowledge dependence. However, most of the problems reported regarding the last three stages may be partially explained due to lack of experience. Table 7 shows that the problems with *Model Training*, *Model Evaluation* and *Model Deployment* are more frequently reported by less experienced data scientists (76% and 92%, respectively). On the other hand, the table shows that the distribution of problems is more balanced among the groups of data scientists for the first three workflow stages (*Model Requirements*, *Data Processing* and *Feature Engineering*).

(3) Exploratory analysis. The most common problems of data scientists in both stages of *Model Training* and *Model Evaluation* addresses the diversity and complexity of alternatives for implementation. Beyond coding, there are many different frameworks and types of models available. For each model selected, there is an infinite number of parameters, metrics, and validation designs. Since the model selection is not so straightforward, data scientists usually choose over exploratory analysis. We present below a few examples of discussions around this issue:

- ($d6$, *Training*) *"Parameter optimizers are time-consuming."*
- ($d5$, *Evaluation*) *"Metrics have no defined target values"*
- ($d6$, *Evaluation*) *"Random selection of training, testing and validation sets."*

Notice that these stages are even more error-prone due to the way that data are made available in the three first stages: *Model Requirements*, *Data Processing* and *Feature Engineering*. Data scientist $d3$ argued that *"... the difficulty in improving the model accuracy may be resultant from problems in the stages of Data Processing and Feature Engineering"*. Besides, data scientist $d7$ commented that *"...the model is created in a loop involving all stages"*. Also, data scientists' answers indicate that the resulting data quality from these stages is usually not verified. For instance, data scientist $d4$ mentioned *"...the difficulty in verifying whether the chosen attributes are the most suitable for the model"*. Since these

stages are very exploratory, domain experts do not end up being heavily involved in them. On the other hand, data scientists end up being uniquely responsible for understanding the data. As a consequence, problems are only discovered late in the process. Thus, while the accuracy of the model is not satisfactory, the stages are reviewed.

Given the mentioned particularities of ML model development, our findings reveal gaps in the developing stages performed by the data scientists. The different personal processes adopted by data scientists in their companies are non-linear, requiring too much rework to satisfy customers' needs. Besides, our observations do not show activities addressing the verification and the validation of the artifacts generated during the workflow stages. Based on these findings, we understand that new and engineered eye on the development of ML models is required. We believe the software engineering values such as traceability and quality assurance should be continuously addressed by concrete, planned, and structured practices. For instance, we see that customized inspection techniques should be developed to support the verification of ML features and models. Thus, we believe that by following software engineering practices since the early ML modeling stages, companies would reduce rework and the dependence of the domain experts, also leveraging the maintainability of ML models. Consequently, they would mitigate recurrent feedback loops in the process by anticipating problems and saving resources.

7 Threats to validity

One threat of our study address the restricted sample size. For mitigating this threat, we interviewed data scientists with diverse background and working contexts. Each data scientist is from a different team, distributed among five different companies located in Brazil. These companies distinguish themselves in terms of size, project domains, and geographical locations. Although our results suggest a great variety of ideas and agreement between responses, we are aware of the chance of geographic bias among the culture of ML practice in the country.

Results in this work are limited to data scientists working with *supervised learning* and *unsupervised learning*. Although our efforts, we were not able to reach into our industrial collaboration network data scientists applying *reinforcement learning*. We believe that this fact originates from *reinforcement learning* be applied for very particular scenarios inside projects challenging to reach. We are aware that the presence of data scientists working with *reinforcement learning* could impact observations. Therefore, we recommend further researches on this topic to complement our results.

We execute the data analysis process using the *open-coding* methodology, which is a subjective task. To avoid bias, we carefully involve distinct authors in the coding, refinements, validation, and discussions since authors have a different interview perspective. Still, to further ensure the validity of our results, a fourth researcher who did not write Chapter 5, traced back the developers' quotations to their source and none traceability problem was found.

To identify possible differences between workflows followed by data scientists, we asked them to compare their workflows with the one used as a reference in this study (see Section 2). In this way, we intentionally did not present very detailed stages in the workflow of Figure 1 since they could generate considerable disagreement and difficult data grouping. In the end, we perceived that our strategy stimulated the data scientists to feel free to discuss the granularity level of the workflow stages, detailing activities at a lower level.

8 Conclusion

This work presented a study to understand how Brazilian data scientists develop ML models in practice. To accomplish our goal, we investigated which stages are considered more challenging from the data scientists' perspectives and the practices involved in the most challenging stages. To perform our investigations, we conducted semi-structured interviews with data scientists from five different Brazilian companies. Then, we applied a code technique on six hours of transcripted interviews for data analysis. As a result, we obtained 447 codes that address *actors*, *activities*, *methods*, *limitations*, and *challenges* related to the ML stages analyzed by our study.

Our findings reveal that data scientists perceive the *Data Processing* and *Feature Engineering* as the most challenging stages during the ML model development. Although, they also mention important issues on the *Model Training*, *Model Evaluation* and *Deployment*. These findings indicate a lack of support of an engineered process to the practice of developing ML models. For instance, we did not find in the interviews techniques for assuring the traceability between the features and the model requirements. Besides, unlike recommended in software engineering, ML tests are typically not planned and do not have their coverage measured.

The intended audience of this paper includes but is not limited to researchers in software engineering and companies. Researchers shall benefit from our results and insights, especially regarding developing processes adopted by companies proposing further investigations or tools to support the development. Also, companies can use results and insights to enhance their development process, for example, encouraging cooperation between actors in challenging stages, hence improving characteristics such as the delivery time.

As future work, we intend to perform interviews by considering professionals from other companies located in different countries. Moreover, we plan to perform action research to deeply understand the challenges and practices during the development of ML models. At last, we also intend to extend the ML workflow by considering the findings of our interviews and experiments.

Bibliography

AMERSHI, S. et al. Software engineering for machine learning: A case study. In: *International Conference on Software Engineering: Software Engineering in Practice*. [S.l.: s.n.], 2019. p. 291–300.

BAGHBAN, A. et al. Modelling of co2 separation from gas streams emissions in the oil and gas industries. *Petroleum Science and Technology*, Taylor & Francis, v. 34, n. 14, p. 1291–1299, 2016.

BYRNE, C. *Development Workflows for Data Scientists*. [S.l.]: O'Reilly Media, 2017.

COHEN, D.; CRABTREE, B. *Qualitative research guidelines project*. 2006.

Hill, C. et al. Trials and tribulations of developers of intelligent systems: A field study. In: *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. [S.l.: s.n.], 2016. p. 162–170. ISSN 1943-6106.

J, C. *Data Scientists: Revealing their Challenges and Practices on Machine Learning Model Development*. [S.l.]: GitHub, 2020. <https://github.com/sbqs2020/sbqs2020>.

KIM, M. et al. Data scientists in software teams: State of the art and challenges. *IEEE Transactions on Software Engineering*, IEEE, v. 44, n. 11, p. 1024–1038, 2017.

MASUDA, S. et al. A survey of software quality for machine learning applications. In: IEEE. *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*. [S.l.], 2018. p. 279–284.

MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997. ISBN 978-0-07-042807-2.

NASCIMENTO, E. d. S. et al. Understanding development process of machine learning systems: Challenges and solutions. In: *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. [S.l.: s.n.], 2019. p. 1–6.

NGUYEN-DUC, A. et al. A multiple case study of artificial intelligent system development in industry. In: *Proceedings of the Evaluation and Assessment in Software Engineering*. [S.l.: s.n.], 2020. p. 1–10.

ONAWOLE, A. T. et al. Computational screening of potential inhibitors of desulfobacter postgatei for pyrite scale prevention in oil and gas wells. *BioRxiv*, Cold Spring Harbor Laboratory, p. 327957, 2018.

PATEL, K. et al. Investigating statistical machine learning as a tool for software development. In: *SIGCHI Conference on Human Factors in Computing Systems*. [S.l.: s.n.], 2008. p. 667–676.

PATIL, D. *Building data science teams*. [S.l.]: " O'Reilly Media, Inc.", 2011.

PISCOPO, A.; SIEBES, R.; HARDMAN, L. Predicting sense of community and participation by applying machine learning to open government data. *Policy & Internet*, Wiley Online Library, v. 9, n. 1, p. 55–75, 2017.

POLYZOTIS, N. et al. Data lifecycle challenges in production machine learning: a survey. *ACM SIGMOD Record*, ACM New York, NY, USA, v. 47, n. 2, p. 17–28, 2018.

STRAUSS, A.; CORBIN, J. *Basics of qualitative research techniques*. [S.l.]: Sage publications Thousand Oaks, CA, 1998.

TABAK, M. A. et al. Machine learning to classify animal species in camera trap images: Applications in ecology. *Methods in Ecology and Evolution*, Wiley Online Library, v. 10, n. 4, p. 585–590, 2019.

WÄLDCHEN, J.; MÄDER, P. Machine learning for image based species identification. *Methods in Ecology and Evolution*, Wiley Online Library, v. 9, n. 11, p. 2216–2225, 2018.

WAN, Z. et al. How does machine learning change software development practices? *IEEE Transactions on Software Engineering*, IEEE, 2019.

ZHANG, J. M. et al. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 2020.

ZHANG, T. et al. An empirical study of common challenges in developing deep learning applications. In: IEEE. *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*. [S.l.], 2019. p. 104–115.

APPENDIX A – Supplementary Material

Trecho	Código	Categoria
basicamente seria isso mesmo, os requisitos do modelo seriam o processo que a gente iria automatizar utilizando inteligência artificial.	Requisitos do Modelo como definição do processo a ser automatizado.	Atividade [Requisitos do modelo]
O processamento de dados seria o principal	Processamento de Dados como etapa principal.	Limitação [Processamento de dados]
Os atributos a gente tem que avaliar	Atributos devem ser analisados.	Atividade [Engenharia de atributos]
porque pode ser que alguns dependendo do caso não influenciam, ou [...] podem ser muito pesados e a gente pode reduzir	Atributos podem estar não otimizados.	Limitação [Engenharia de atributos]
então valida e vê se tem que mudar alguma coisa	Incerteza no resultado de treino	Desafio [Treino do modelo]
	Avaliação do modelo indica a corretude do treino.	Limitação [Avaliação do modelo]
é como se a gente (refere-se a sua equipe de trabalho) já tivesse o objetivo, tem que fazer um classificador de...	Requisitos do Modelo como definição do modelo a ser criado.	Limitação [Requisitos do modelo]
Nós trabalhamos em módulos isolados [...] esses módulos serão usados em sistema maior que tem todo processo de negócio a ser seguido [...] então fica meio que independente.	Modelos de machine learning como módulos em um MLES.	
a gente (refere-se a empresa) tem um sistema a desenvolver, uma ferramenta, no final a gente entrega a solução inteira.	Empresa desenvolve MLES.	
eu estou envolvido em todas as etapas	Único desenvolvedor em todas as etapas.	Ator [Geral]
mas às vezes dependendo do projeto tem que ter o especialista nos dados (refere-se a um especialista no domínio da aplicação, não um especialista em data science), um pessoa que [...] consiga interpretar os dados, que tenha conhecimento sobre aquela área de negócio [...].	Presença de um especialista de domínio.	Ator [Geral]
Eu trabalho com imagem, às vezes eu preciso classificar coisas na imagem e eu não sei o que são, então eu dependo de um especialista que vai anotar as imagens pra mim [...].	Desenvolvedor incapaz de reconhecer seus dados	Desafio [Processamento de dados]
	Especialista de domínio anota os dados	Atividade [Processamento de dados]
Depende, em casos que você tem literatura você já tem um alvo a atingir, você tem que chegar naquela medida de acurácia,	Literatura define as metas de avaliação.	Limitação [Avaliação do modelo]
mas dependendo do caso, se você estiver trabalhando em um negócio novo, qualquer coisa seria melhor, acima de 50% ou 60%.	Novas soluções não precisam de ótimos resultados de avaliação.	Limitação [Avaliação do modelo]
não chegamos assim a fundo nesse ponto [...], é só mais a acurácia mesmo [...]. A depender, em na detecção de imagens também uso o recall.	Uso de metricas clássicas.	Método [Avaliação do modelo]
	Métricas clássicas utilizadas na avaliação do modelo.	Atividade [Avaliação do modelo]
o workflow se repete, pode se repetir sim [...]. Porque você não tem certeza que aquilo é o caso ótimo.	Ciclos de desenvolvimento existem devido a incerteza de corretude.	Desafio [Geral]
depende da especificidade de cada projeto, mas aparentemente o processamento de dados.	O projeto influencia na etapa que mais se repete.	Limitação [Geral]
	Processamento de dados como etapa que mais se repete.	Desafio [Processamento de dados]
Por exemplo, na detecção de imagem nós estávamos trabalhando (refere-se a uma atividade anterior) com um dataset que não era tão grande, então (a princípio) fizemos um treino com o que tinha. Na avaliação ficou mais ou menos, aí eu voltei para o processamento de dados, gerei mais imagens: girei, inseri ruído [...].	Avaliação da indicios da corretude do modelo.	Limitação [Avaliação do modelo]
	Enriquecimento de dados.	Método [Processamento de dados]
	Dados em quantidade ou qualidade inferior são enriquecidos.	Atividade [Processamento de dados]
Processamento de dados, sem dúvidas, porque tudo depende disso, se os dados estiverem errados, o modelo vai sair errado [...].	Processamento de dados como etapa mais problemática	Desafio [Processamento de dados]
	Dados errados implicam em modelo errado.	Limitação [Processamento de dados]
Não só a quantidade dos dados, mas também a qualidade, pra balancear [...]. Então é o principal a definir para fazer qualquer coisa.	Quantidade dos dados como aspecto importante	Limitação [Processamento de dados]
	Qualidade dos dados como aspecto importante.	Limitação [Processamento de dados]
esse é o principal problema que eu vejo em Machine Learning, montar o dataset [...], ter certeza que aquilo tá certo (refere-se aos dados).	Dificuldade em concretizar o dataset	Desafio [Processamento de dados]
	Dificuldade em garantir a corretude do dataset.	Desafio [Processamento de dados]
depende de cada caso, no meu caso em específico é a dependência que tenho do especialista (refere-se ao especialista no domínio da aplicação)	O projeto influencia nos problemas a serem enfrentados	Limitação [Desenvolvimento]
	Desenvolvedor incapaz de reconhecer seus dados	Desafio [Processamento de dados]

no meu caso há todo um processo de aquisição das imagens, então eu identifico visualmente	Identificação visual de problemas em imagens	Atividade [Processamento de dados]
em alguns projetos (refere-se aos de não detecção de imagens) são utilizadas ferramentas para gerar gráficos dos dados como histogramas e boxplots, geralmente com o auxílio do especialista.	Gráficos utilizados para identificação dos problemas	Atividade [Processamento de dados]
	Uso de gráficos para a visualização de dados: boxplot, histogramas.	Método [Proessamento de dados]
	Epecialista de domínio para a identificação de problemas.	Atividade [Processamento de dados]
	Presença de um especialista de domínio.	Ator [Processamento de dados]
eu vejo sempre na literatura (se alguma ferramenta é indicada), mas nenhuma em específico.	Literatura define novas ferramentas a serem incorporadas	Limitação [Geral]
sabendo que os dados estão corretos, se o especialista me diz que naquela região X tem aquilo [...]. Eu só tenho que ter amostras e testar arquiteturas.	Especialista de domínio anota os dados	Atividade [Processamento de dados]
	Desenvolvedor confia no especialista de domínio.	Limitação [Processamento de dados]
isso depende de análise específica do modelo que você está utilizando, mas é mais empírico sim, analisar a saída camada a camada e testar [...].	Definição empírica de hiperparâmetros para os modelos	Atividade [Treino do modelo]
a gente já tem bibliotecas que tem a arquitetura implementada, até porque as arquiteturas para redes convolucionais são bem grandes, então fazer do zero …	Uso de frameworks para criação de modelos.	Método [Treino do modelo]
no meu caso, eu não sei. Porque os desafios que nós temos hoje [...] são de deteção de objetos do mundo real [...]. Eu estou em um outro contexto que não tem muito a ver com isso, então eu utilizo pesos pré-treinados. [...] Pode ser que seja melhor fazer do zero ou não.	Incerteza sobre a corretude dos modelos.	Desafio [Treino do modelo]

Trecho	Código	Categoria
Nessa engenharia de atributos é que você tem toda essa parte de estatística, conhecimento dos seus dados, redução dos dados [...].	Engenharia de atributos como a etapa de análise dos dados.	Atividade [Engenharia de atributos]
O que é importante para Machine Learning? Grande volume de dados, se você não tiver, você não pode utilizar [...].	Volume de dados como aspecto crucial para uso de ML.	Limitação [Geral]
Em uma área exploratória não funciona (refere-se a Machine Learning) porque eu não tenho dados, entendeu? Eu consigo isso se eu tenho muitos dados na área.	Áreas exploratórias dependem da quantidade dos dados.	Limitação [Geral]
bom, no nosso caso nós conseguimos uma forma de programação que a gente pode distribuir (desenvolvedores) para cada uma (refere-se às etapas) [...], mas todo mundo sabe entrar em cada módulo desse.	Desenvolvedores diferentes para cada uma das etapas como alternativa	Limitação [Geral]
	Presença do desenvolvedor	Ator [Geral]
	Cada desenvolvedor conhece todas as etapas.	Limitação [Geral]
Nós fazemos nessa sequência, mas aí um pode ficar na estatística, outro pode ficar na seleção, outro pode fazer os modelos lá de classificação, mas [...] eles podem compartilhar com todo mundo.	Desenvolvedores diferentes para cada uma das etapas	Limitação [Geral]
	Cada desenvolvedor deve se comunicar com todos os demais.	Limitação [Geral]
sim, no nosso caso como nós trabalhamos com a <<compahia>>, tem alguém de lá que vai analisar	Cliente analisa a qualidade do modelo gerado.	Atividade [Avaliação do modelo]
	Presença do cliente	Ator [Avaliação do modelo]
a gente roda o modelo discute com eles e então eles vão analisar tudo.	Modelo posto em execução para análise do cliente.	Atividade [Avaliação do modelo]
Eles vão dizer: olhe podemos utilizar essa variável, podemos retirar essa variável, pela experiência deles.	De acordo com sua experiência o cliente pode indicar melhorias em relação aos atributos do modelo.	Atividade [Avaliação do modelo]
bem, nessa área hoje estamos Eu e o <<cientista de dados>>, a gente discute [...].	Internamente modelo avaliado por todos os desenvolvedores.	Atividade [Avaliação do modelo]
sim, as etapas se repetem. Você volta, muda as variáveis ou o modelo que pode usar, e testar esse conjunto de dados. [...]	Ciclos de desenvolvimento existem devido a incerteza de corretude.	Desafio [Geral]
Posso fazer uma outra redução, um PCA, um discriminate [...]. Nessa parte isso é bem cíclico.	Uso de métodos estatísticos na Engenharia de Atributos.	Atividade [Engenharia de atributos]
Isso aqui ó, [...] aqui sim você fica trabalhando: nos atributos, treino e avaliação do modelo, nestes três a gente fica trabalhando aqui.	Avaliação da indícios da corretude do modelo	Limitação [Avaliação do modelo]
[...] (devido a) quantidade de dados, qualidade dos dados, redundância dos dados, e o que a gente chama de [...] correlações entre os dados [...].	Quantidade dos dados como aspecto importante	Limitação [Processamento de dados]
	Qualidade dos dados como aspecto importante	Limitação [Processamento de dados]
	Correlação entre os dados como aspecto importante.	Limitação [Engenharia de atributos]
Porque a gente pensa que pega os dados seleciona os atributos e joga, gerou o modelo [...]. Se você achou que tá um modelo bom pra esse conjunto de dados, você vai aplicar. Depois quando você vai aplicar em outros dados, você ver que seu modelo não está ajustado, porque você não tratou a casualidade, redução dos dados [...].	Overfitting causado pela falta de cuidado no tratamento dos dados.	Desafio [Engenharia de atributos]
Eu acho que essa é a parte mais importante e mais trabalhosa, processamento de dados e seleção de atributos	Processamento de dados é a etapa mais trabalhosa	Desafio [Processamento de dados]
	Engenharia de atributos é a etapa mais trabalhosa.	Desafio [Engenharia de atributos]
Porque o resto você fica mudando parâmetro, usa essa rede, usa essa.	Definição empírica de hiperparâmetros para os modelos	Atividade [Treino do modelo]
	Definição empírica de arquiteturas para os modelos	Atividade [Treino do modelo]
ruídos, [...] falta de dados, [...] erros nos dados, muitos erros. Variáveis que não tem causalidade, que não tem relação nenhuma.	Problemas nos dados relacionados a erros	Desafio [Processamento de dados]
	Problemas nos dados relacionados a ruídos	Desafio [Processamento de dados]
	Problemas nos dados relacionados a variáveis pouco representativas.	Desafio [Engenharia de atributos]
estatística multivariada, PCA, análise de discriminantes. Tudo da estatística multivariada, [...]	Métodos estatísticos utilizados para identificar problemas nos dados.	Método [Engenharia de atributos]
eu conhecendo o meu dado eu posso identificar os melhores atributos que eu vou utilizar.	Conhecer o dado auxilia a identificar os melhores atributos.	Limitação [Engenharia de atributos]
gente usa precisão, recall, analisa a matriz de confusão, gera vários modelos e usa a Curva ROC.	Métricas clássicas são utilizadas na avaliação do modelo	Atividade [Avaliação do modelo]
	Uso de métricas clássicas na avaliação.	Método [Avaliação do modelo]
a gente utiliza tudo que tem no Python, tudo Python aqui. [...] O Python já nos disponibiliza tudo.	Uso de python e frameworks para a criação do modelo.	Método [Treino do modelo]
Eu uso o Python, como eu tô com meus dados no Pandas eu consigo usar só as minhas variáveis (refere-se às variáveis de interesse do modelo) [...], eu discrimino, eu nomeio as variáveis de interesse.	Alteração de dados para corrigir problemas.	Atividade [Engenharia de atributos]
com os dois, nós trabalhamos com classificação, e muita coisa é regressão.	Uso de modelos para atividades de classificação e regressão.	Método [Geral]

Trecho	Código	Categoria
(a etapa de processamento de dados seria) falar, "ah esses atributos talvez sejam interessantes para mim, então deixa eu pegar esses dados", entendi.	Processamento de dados como etapa de coleta de dados relevantes.	Atividade [Processamento de dados]
Então fazer outras coisas com os atributos (refere-se a engenharia de atributos).	Engenharia de atributos como a etapa de manipulação do dado coletado.	Atividade [Engenharia de atributos]
Engenharia, treino, avaliação, volto para a engenharia.	Ciclo de desenvolvimento existe devido a incerteza da corretude.	Desafio [Geral]
As vezes eu pego mais dados (refere-se ao processamento de dados), vejo que não tenho uma coluna que poderia ter.	Processamento de dados reexecutado caso dados não sejam suficientes.	Atividade [Processamento de dados]
Eu não sei se só de treinar eu já voltaria para a engenharia de atributos, porque eu precisaria de avaliar o que estou fazendo para voltar para alguma etapa.	Avaliação da indicios da corretude do modelo.	Limitação [Avaliação do modelo]
	Resultado da avaliação da indicios da etapa que gerou problemas.	Limitação [Avaliação do modelo]
Ficaria nesse ciclo aqui (engenharia de atributos, treino do modelo, avaliação do modelo), às vezes voltando nesse aqui (refere-se ao processamento de dados). Esse daqui (refere-se aos requisitos do modelo), só se a pessoa me voltar com outra especificação.	Cilco de desenvolvimento existe devido a incerteza da corretude.	Desafio [Geral]
	Etapa de requisitos do modelo só é revisitada caso surjam novos requisitos	Apontamento [Requisitos do Modelo]
o mesmo desenvolvedor em todas as etapas. Assim, mas partindo do pressuposto que esses requisitos é algum usuário do sistema que está querendo aquilo, você está partindo de uma necessidade externa, não é você que está definindo a necessidade.	Único desenvolvedor em todas as etapas.	Limitação [Geral]
	Presença do desenvolvedor	Limitação [Geral]
não.	Metodologias de desenvolvimento não são seguidas.	Limitação [Geral]
eu tenho um padrão que eu sigo, mas que não necessariamente é o padrão dos desenvolvedores.	Padrões de código não são seguidos.	Limitação [Geral]
pode ser que eu chegue aqui na avaliação e perceba, poxa [...] eu não consigo melhorar a qualidade do meu modelo de jeito nenhum. Volto nos dados, dou uma olhada, "será que eu estou perdendo alguma coisa?".	Avaliação da indicios da corretude do modelo.	Limitação [Avaliação do modelo]
	Dificuldade em melhorar resultados de avaliação pode indicar problemas nos dados.	Limitação [Avaliação do modelo]
As vezes tem alguma conversa com o especialista do domínio pra entender alguma coisa. Aí ele fala, "tem alguma feature que pode ser interessante pra você"	Especialista do domínio indica atributos relevantes.	Atividade [Engenharia de atributos]
	Presença do especialista de domínio	Ator [Engenharia de atributos]
associado ao cliente, é a pessoa que entende dessa tarefa (domínio da aplicação) [...].	Especialista do domínio associado ao cliente.	Ator [Geral]
eu acho que existem diversas realidades, quando você vai trabalhar em projetos maiores, faz todo o sentido você ter um time misto. Por exemplo, tem uma pessoa que é especialista em visualização, uma pessoa que é especialista em estatística, um outro que é especialista em algoritmo de Machine Learning.	Projetos maiores requerem especialistas em cada uma das etapas.	Limitação [Geral]
Eu acho que se talvez eu fosse ser encaixado em uma delas eu acredito que estaria mais para o lado de Machine Learning.	Entrevistado se considera mais experiente no Treino do Modelo.	Limitação [Treino do modelo]
isso daí a pessoa responsável por isso daqui (requisitos do modelo) está sempre tentando melhorar a qualidade.	Cliente avalia a qualidade do modelo	Atividade [Avaliação do modelo]
	Cliente define os requisitos dos modelo.	Atividade [Requisitos dos modelo]
	Presença do cliente	Ator [Avaliação do modelo]
	Presença do cliente	Ator [Requisitos dos modelo]
eu vou considerar a mais trabalhosa. É a etapa de obtenção dos dados (processamento de dados) e a de trabalhar os dados (engenharia de atributos).	Processamento de dados como etapa mais trabalhosa	Desafio [Processamento de dados]
	Engenharia de atributos como etapa mais problemática	Desafio [Engenharia de atributos]

muitas vezes você não sabe quais dados você vai ter que pegar, você vai ter que obter os dados de bases diferentes, você vai ter que juntar esses dados, às vezes você vai ter que usar algumas convenções próprias de por exemplo: "o que eu faço com dados inexistentes?".	Incerteza sobre os melhores dados para a coleta	Desafio [Processamento de dados]
	Dados de bases diferentes dificultam a obtenção	Desafio [Processamento de dados]
	Dificuldade de definir convenções próprias para dados faltantes.	Desafio [Engenharia de dados]
	Tratar dados faltantes	Atividade [Engenharia de dados]
Para a engenharia de atributos: "como eu vou transformar uma coluna [...]?". Uma data por exemplo pode não fazer sentido para mim, e sim uma diferença de datas [...]. Então como fazer essas transformações.	Dificuldade em identificar transformações adequadas para os dados	Desafio [Engenharia de atributos]
	Transformação dos dados	Atividade [Engenharia de atributos]
Nessa etapa um dos maiores problemas é o tempo que você gasta.	Processamento de dados demanda muito tempo	Desafio [Processamento de dados]
	Engenharia de atributos demanda muito tempo	Desafio [Engenharia de atributos]
testando em uma base pequena pra ver se aquilo realmente está funcionando da maneira esperada [...]. Eu construo uma mini base de teste pra ver se o pré-processamento está funcionando da maneira correta [...].	Scripts para manipulação dos dados são testados em bases menores antes da execução.	Atividade [Processamento de dados]
	Scripts de manipulação de atributos são testados em bases menores antes da execução.	Atividade [Engenharia de atributos]
Na obtenção de dados, se eu estou pegando de bases diferentes, eu faço uma verificação para ver se o dado está vindo correto.	Os dados são verificados durante a coleta	Atividade[Processamento de dados]
A base de testes é uma base pequena que eu posso facilmente verificar no olho [...]. Eu executo essa parte do pré-processamento e da engenharia de atributos nessa base para verificar se o negócio está sem erros [...]. Para tentar identificar algum bug no código [...] do processamento de dados e da engenharia de atributos.	Problemas nos scripts são verificados visualmente nas bases	Limitação [Processamento de dados]
Eu tenho uma base de testes, que eu nunca olho pra ela, nunca olho mesmo. Faço os experimentos usando cross-validation, mas dependendo do tamanho da base não tem como fazer [...]. Depois que eu chego em um ponto que eu não tenho mais como melhorar a qualidade do modelo e esse modelo vai ser o final, [...] eu pego o modelo [...], venho nessa base separada e executo uma vez e verifico o overfitting.	Uso da base de validação para verificar overfitting	Atividade [Avaliação do modelo]
eu já participei da criação de bases, que é uma etapa muito difícil. Mas o que nós fizemos …, não, não verifica fairness.	Fairness não verificado	Limitação [Avaliação do modelo]
vou no código, modifico e testo de novo.	Problemas são corrigidos através de modificação e reexecução	Atividade [Geral]
escolhendo especificamente redes neurais, por exemplo: tem a loss, que é o que ele tenta reduzir para melhorar a qualidade da rede neural. A cada época que passa, você vê a loss [...]. A sua loss no conjunto de treino ela vai sempre decrescendo, no conjunto de validação ela decresce até certo ponto que começa a subir. Aí em um five fold cross validation [...], eu vou verificar em qual época eu tive a menor loss em cada um dos folds, pego cada um deles e faço a média. Quando eu vou criar meu modelo final [...] eu vou rodar ele tantas épocas quanto a média de onde ele encontrou o menor loss no conjunto de validação.	A correção do overfitting é feita através da análise da curva loss e restrição das épocas de treino.	Atividade [Treino do modelo]
Mas com Naive Bayes eu nunca consegui um resultado (bom), mas você acaba colocando para fazer um comparativo de modelos. Por exemplo eu quero resolver uma tarefa de classificação [...], ao invés de você testar só um algoritmo e ficar melhorando as features dele, você seleciona alguns algoritmos [...] e compara os resultados entre eles.	Modelos em desenvolvimento são comparados com baselines.	Atividade [Avaliação do modelo]
depende do tipo de tarefa. Em geral se é uma tarefa conhecida, comum na literatura, eu olho as métricas que são usadas em outros trabalhos	Literatura define as metas de avaliação.	Limitação [Avaliação do modelo]
Se é um trabalho que não tem um dataset comum, eu vejo o que faz sentido naquele caso, geralmente o F1 é uma medida que eu uso. Mas tem casos que faz sentido a pessoa maximizar o recall.	Novas soluções requerem a definição da métrica mais adequada para avalia-la.	Limitação [Avaliação do modelo]

fazendo Machine Learning eu não estou propondo nenhum algoritmo [...], então essa parte (implementação dos algoritmos) [...] eu trato como uma caixa preta, porque eu sei que pode haver bugs, mas tem pessoas focadas em resolver esses bugs.	Uso de frameworks para a criação de modelos.	Método [Treino do modelo]
	Confiança na corretude da implementação de frameworks.	Limitação [Treino do modelo]
Eu estou aqui no meu fluxo, tentando melhorar pré-processamento, garantir que o pré-processamento será bem feito, [...] e tentando melhorar a qualidade (do modelo).	Entrevistado considera apenas a corretude do seu trabalho.	Limitação [Treino do modelo]

Trecho	Código	Categoria
sim, seguimos isso mesmo (sequência de etapas). De repente tem mais granularidade, mas de maneira geral é isso mesmo [...]. Por exemplo, o processamento de dados tem um conjunto de atividades [...], engenharia de atributos também, mas beleza.	Atribuíria mais granularidade ao workflow	Limitação [Geral]
não, não. Eu participava de todas as etapas mesmo. Desde o processamento dos dados, até a implantação do modelo. Às vezes a implantação não muito, mais o cara da infraestrutura. Então seria mais daqui (refere-se ao processamento de dados) até a avaliação do modelo. Em todos os projetos que eu vi, só tinha um cara pra fazer tudo isso.	Único desenvolvedor em todas as etapas.	Limitação [Geral]
	Presença do desenvolvedor	Ator [Geral]
	A implantação do modelo é auxiliada por uma equipe de infraestrutura.	Limitação [Implantação do modelo]
	Presença da equipe de infraestrutura	Ator [Implantação do modelo]
essas daqui: processamento de dados, engenharia de atributos, treinamento do modelo e avaliação. [...] Eu passava todos os pacotes para o cara (da infraestrutura) e ele implantava lá.	O modelo gerado é repassado para a equipe de infraestrutura para implantação	Atividade [Implantação do modelo]
eu participava da parte dos requisitos, porém mais para o entendimento. Aí tinham mais pessoas envolvidas no projeto, como o gerente	Desenvolvedor participa da elicitação de requisitos	Atividade [Requisitos do modelo]
	Gerente participa da elicitação dos requisitos	Atividade [Requisitos do modelo]
	Presença do gerente	Ator [Requisitos do modelo]
	Cliente participa da elicitação dos requisitos	Atividade [Requisitos do modelo]
	Presença do cliente	Ator [Requisitos do modelo]
cara, tipo assim, uma metodologia em si não. Era algo meio empírico, sabe?	Metodologias de desenvolvimento não são seguidas.	Atividade [Geral]
não, padrão de código não. [...] nós trabalhávamos com coisas mais ou menos conhecidas, como comentar o código [...], como era python toda essa questão da indexação do código tem que estar certinha [...], é mais isso.	Padrões de código não são seguidos.	Limitação [Geral]
Muitas vezes já aconteceu de nós estarmos treinando o modelo e a gente encontrar problemas, tendo que voltar para a engenharia de atributos. Isso também ocorreu na avaliação, pegar muita coisa que não está certa e voltar para a engenharia de atributos. É assim, é um vai e volta.	Ciclos de desenvolvimento existem devido a incerteza de corretude	Desafio [Geral]
	Problemas na avaliação indicam problemas na engenharia de atributos.	Limitação [Avaliação do modelo]
a avaliação. Se você não chega a um resultado bom, você tem que voltar, trocar modelo, ou mexer mais com os dados.	Avaliação do modelo como etapa que mais se repete.	Desafio [Avaliação do modelo]
no meu caso sim. Como a gente trabalhava com uma metodologia ágil, então temos o product owner. Então esse cara que fala se temos bons resultados ou não. É porque o cara conhece mais do negócio, entendeu? Aí o cara tá muito envolvido nos requisitos, e ele tem uma boa perspectiva do que o cliente quer.	Empresa desenvolve MLES.	
	O desenvolvimento do MLES segue uma metodologia ágil.	
	O cliente avalia a qualidade do modelo.	Atividade [Avaliação do modelo]
	Presença do cliente	Ator [Avaliação do modelo]
Por exemplo, a gente chegava a uma acurácia de 80%, mas pra o cara [...] isso não era suficiente. Porque o cara queria mais entender das causas do que a acurácia do modelo.	Métricas clássicas não suficientes para avaliar o modelo	Desafio [Avaliação do modelo]
	Interpretabilidade mais importante do que métricas.	Limitação [Avaliação do modelo]
cara, a engenharia de atributos. Porque deixar tudo certinho, fazer toda a limpeza, escolher as features importantes. Porque o treino do modelo é só rodar um algoritmo e pronto.	Processamento de dados como etapa mais problemática	Desafio [Processamento de dados]
	Engenharia de atributos como etapa mais problemática.	Desafio [Engenharia de atributos]
eu considero essa etapa mais problemática pelo esforço.	Esforço como palavra chave para definir etapa problemática.	Limitação [Geral]
na minha opinião é pegar um monte de atributos que não são relevantes para o modelo, que vão gerar muito ruído [...].	Problemas nos dados relacionados a ruídos	Desafio [Processamento de dados]
	Problemas nos dados relacionados a variáveis não representativas.	Desafio [Engenharia de atributos]

Então eu acho que isso é uma coisa que impacta demais no treinamento [...], na acurácia [...] e na avaliação.	Processamento de dados impacta nas demais etapas.	Limitação [Processamento de dados]
	Engenharia de atributos impacta nas demais etapas.	Limitação [Engenharia de atributos]
com certeza, aqui na engenharia de atributos. Se você tiver um monte de variáveis que fazem muito ruído é muito mais provável que teremos um overfitting no treino do modelo [...]. Nos olhamos overfitting também na árvore, na primeira versão que a gente rodou, a gente chegou em uma árvore muito desbalanceada, então isso [...] (indica) um overfitting muito grande, um ramo era muito grande [...].	Overfitting observado no balanceamento da árvore	Atividade [Avaliação do modelo]
	Origem do overfitting no processamento de dados.	Limitação [Processamento de dados]
nós fazemos análise de correlação e covariância e por conta disso a gente parte da ideia que ficou certo [...].	Uso de métodos estatísticos na engenharia de atributos	Método [Engenharia de atributos]
	Métodos estatísticos aplicados aos dados para evitar a existência de overfitting	Atividade [Engenharia de atributos]
Nós também fazemos uma validação cruzada com o cliente. [...] Porque nós não deixamos só que o algoritmo decida o que é ou não importante para o modelo. Já aconteceu do algoritmo falar para a gente que X variável não é importante, mas o cara lá do cliente fala, "essa variável é importante, você não pode tirar esse cara". Também acontece o contrário. Por exemplo, ele fala que algo (refere-se a um atributo) não é importante, mas [...] roda o modelo sem a variável e ela é importante.	Especialista do domínio indica atributos relevantes.	Atividade [Engenharia de atributos]
	Presença do especialista de domínio	Ator [Engenharia de atributos]
não, nesse caso não [...]. Só o Jupyter, Python.	Uso de frameworks e bibliotecas para a criação de modelos: Jupyter, Python	Métodos [Treino do modelo]
manipulando de novo né. Adicionando atributos, removendo atributos, normalizando os dados [...].	Problemas são corrigidos através de modificação e reexecução	Limitação [Geral]
isso, a gente chegou nessa conclusão. Se o cara quer entender das causas [...], nós não podíamos usar uma rede neural, porque nós chegaríamos a um resultado, mas a interpretação desse cara aí seria complicada [...]. Então por conta disso, a gente escolheu árvore de decisão.	Árvore de decisão necessária devido ao requisito de interpretabilidade	Limitação [Geral]
	Redes Neurais não provêem inpterpretabilidade.	Limitação [Geral]
nesse caso o product owner que fala se está bom ou não, de acordo com o que ele conhece do cliente.	O cliente define os melhores valores para as métricas de avaliação.	Atividade [Avaliação do modelo]
então, nessa primeira entrega, nós chegamos a um modelo não tão bom [...].	Modelo não satisfatório em termos de regressão, utilizando árvore de decisão	Limitação [Geral]
Como eu estava falando, naquela questão das causas, nós temos um modelo que é bom, ele explica bem o que está acontecendo, mas a previsão no modelo de regressão é ruim. Nesse caso nós entregamos essa v zero que atende a explicação das causa, mas a predição e a regressão não foi boa. Então estamos vendo se mudamos de algoritmo, como pegar uma rede neural [...].	Modelo com boa interpretabilidade	Limitação [Avaliação do modelo]
	Modelo com baixa precisão de regressão	Limitação [Avaliação do modelo]
	Rede Neural como Alternativa para regressão.	Limitação [Avaliação do modelo]
Na próximas etapas, vamos ver se exploramos mais a engenharia de atributos, dependendo do que nós encontrarmos ali, a gente continua, ou muda de algoritmo [...].	Incerteza sobre a qualidade da engenharia de atributos.	Desafio [Engenharia de atributos]
árvore de decisão é muito mais simples de interpretar do que outros algoritmos, mesmo random forest tendo estrutura de árvore ele cria várias, então a interpretação não é tão simples assim, como tendo só uma árvore [...].	Árvore de decisão como a melhor alternativa para interpretabilidade.	Limitação [Requisitos do modelo]

Trecho	Código	Categoria
assim, da forma como está eu realizo (segue o workflow), no geral nessa ordem,	O desenvolvimento segue o workflow.	Limitação [Geral]
eu ia comentar apenas da questão do ciclo, porque é quando você vai avançando (nas etapas) que você vê a necessidade de voltar [...]	Ciclos de desenvolvimento existem devido a incerteza de corretude	Desafio [Geral]
De forma geral é isso mesmo. E eu acho que passo mais um tempo tratando os dados, atributos.	Processamento de dados como etapa que toma mais tempo	Desafio [Processamento de dados]
	Engenharia de atributos como etapa que toma mais tempo	Desafio [Engenharia de atributos]
No projeto que eu trabalho, os requisitos já estão prontos, são entregues para mim.	Desenvolvedor não participa da especificação de requisitos.	Limitação [Requisitos do modelo]
Sobre a implantação, eu encontrei mais dificuldade, uma coisa é você trabalhar com arquivo csv e gerar um modelo, testar e tudo, outra coisa é implantar ele em um sistema que já existe e que muitas coisas não podem ser alteradas.	A empresa não desenvolve MLES	
	Implantação do modelo em um sistema pré-existente	Limitação [Implantação do modelo]
	Dificuldade de implantar o modelo devido a regras de negócio do sistema pré existente.	Desafio [Implantação do modelo]
Uma outra coisa sobre a implementação é que eu vinha trabalhando com sklearn, e eu sinto que é uma biblioteca mais para você estudar, desenvolver modelo do que para aplicar no mundo real (aplicações reais). Eu vi que existiam outras alternativas mais profissionais, mas tive que me resguardar a esta. [...] Outra alternativa seria o Spark, algumas outras alternativas que são mais escaláveis.	Uso de Frameworks para criação de modelos	Método [Treino do modelo]
	Frameworks utilizados são pouco escaláveis para aplicações reais	Desafio [Treino do modelo]
	Frameworks mais profisisonais existem.	Limitação [Treino do modelo]
apenas um desenvolvedor mesmo, só eu que trabalhava com isso.	Único desenvolvedor em todas as etapas.	Limitação [Geral]
	Presença do desenvolvedor	Ator [Geral]
não, o projeto era bem livre, não tinha apego a nenhum tipo de metodologia, acho que poderia até ajudar, mas ...	Metodologias de desenvolvimento não são seguidas.	Limitação [Geral]
não, também não. Tem que usar o bom senso aí.	Padrões de código não são seguidos.	Limitação [Geral]
essas etapas de treino e avaliação tem que se repetir né? você tá o tempo todo tentando tunar o seu modelo. [...] Isso é o principal.	Ciclos de desenvolvimento existem devido a incerteza de corretude	Desafio [Geral]
nessas quatro do meio (processamento de dados, engenharia de atributos, treino do modelo e avaliação do modelo).	Entrevistado considera ter experiência nas etapas de construção do modelo.	Limitação [Geral]
pessoa? Tem alguém acima de mim (gerente) que avaliava o modelo, os problemas que eu tinha.	Gerente de projeto avalia a qualidade do modelo de acordo com as necessidades do cliente.	Atividade [Avaliação do modelo]
	Presença do gerente	Ator [Avaliação do modelo]
esse modelo ele é não supervisionado, modelo de cluster então eu usava umas métricas de homogeneidade, métricas de completude e teve um momento que eu avaliei o silhouette. São métricas um pouco diferente da aprendizagem supervisionada.	Uso de aprendizagem não supervisionada	Atividade [Treino do modelo]
	Criação de modelos para clusterização.	Atividade [Treino do modelo]
	Métricas clássicas para avaliar o modelo.	Atividade [Avaliação do modelo]
assim, eu não tinha a ideia de valor satisfatório, o que eu tentava fazer era maximizar essas métricas, até porque a interpretação delas não é muito simples.	Métricas não possuem valor alvo definido influenciado pelo tipo de aprendizagem.	Limitação [Avaliação do modelo]
	Busca pela maximização das métricas.	Atividade [Avaliação do modelo]
	Métricas para modelos não supervisionados não possuem interpretação simples.	Desafio [Avaliação do modelo]
Outra coisa que fiz foi gerar gráficos, que ilustram um aspecto importante do nosso resultado.	Uso de gráficos para a avaliação do modelo.	Atividade [Avaliação do modelo]

eu colocaria a etapa de treino do modelo, porque muitas vezes para você desenvolver e avaliar um modelo é complicado você testar o que você tá fazendo.	Treino do Modelo como etapa mais problemática.	Desafio [Treino do modelo]
Primeiro porque tem pouco material sobre o desenvolvimento de modelos assim para fins práticos,	Falta de material relacionado a modelos não supervisionado com fins práticos.	Desafio [Geral]
outra coisa é porque é difícil de você testar o código porque muitos modelos por exemplo tem características aleatórias. Então o desenvolvimento não é sentar e fazer.	Característica aleatória do modelo dificulta a verificação.	Desafio [Treino do modelo]
Muitas vezes você também demora para ter seu resultado, então você quer avaliar o seu modelo e leva alguns minutos, isso tudo pode quebrar muito o fluxo do seu trabalho.	Demora para o treino do modelo quebra no ritmo de trabalho.	Desafio [Treino do modelo]
Eu também tive problema na implantação, porque eu tava mais habituado com o treinamento do modelo, avaliação e na hora de implantar eu tive que para um pouco para aprender outras tecnologias, como integrar no sistema em si.	Dificuldade na implantação do modelo devido a necessidade de dominar outras tecnologias.	Desafio [Implantação do modelo]
primeiro o tempo, as vezes demora muito para treinar.	Demora para o treino do modelo.	Desafio [Treino do modelo]
Às vezes você tem algum tipo de problema, e ou a documentação é escassa ou a comunidade não resolveu aquele problema.	Frameworks para aprendizagem não supervisionada com documentação escassa.	Desafio [Treino do modelo]
	Frameworks para aprendizagem não supervisionada com problemas não solucionados	Desafio [Treino do modelo]
Eu diria que (a aprendizagem não supervisionada) não está tão madura assim.	Comunidade não supervisionada está pouco madura.	Desafio [Geral]
dos problemas que eu citei, a questão do tempo você vê enquanto está usando [...]. Os erros que falei, por muitas vezes são erros na execução do algoritmo, então é erro de código mesmo, que dá alguma saída no terminal e você procura [...].	Problemas no treino do modelo identificado através da análise de log.	Atividade [Treino do modelo]
o meu caso é um pouco mais complicado, porque a aprendizagem é não supervisionada, então não tenho respostas na hora [...]. Eu acho que essa área de aprendizagem não supervisionada está bem atrás da supervisionada.	Overfitting não explorado devido a característica da aprendizagem não supervisionada.	Desafio [Avaliação do modelo]
então, eu uso: Python, Spider, Jupyter, e as bibliotecas como numpy, pandas e o Sklern.	Uso de python, jupyter e frameworks para criação de modelos.	Atividade [Treino do modelo]
como eu te disse, a maioria dos problemas são de implementação, então corrijo buscando documentação, StackOverflow e vou corrigindo.	Problemas de implementação solucionados em foruns.	Atividade [Treino do modelo]
	Problemas de implementação solucionados através da documentacão.	Atividade [Treino do modelo]
logo quando eu entrei no projeto, tinha uma planilha gigante, que já tinha os dados no formato que eu usava. Só que aí, quando foi mais tarde com a necessidade de implantar, eu tive que consumir os dados direto do banco, então não tinha eles mais nesse formato certinho.	Dados foram recebidos processados	Limitação [Processamento de dados]
	A implantação do modelo faz necessário modificar o formato de dados recebidos pelo modelo.	Limitação [Avaliação do modelo]
no meu caso [...] teve muita coisa que eu tive que adaptar, tipo, tem o algoritmo do K-Means, só que o K-Means não se dá muito bem com dados faltantes, então eu tive que adaptar para fazer uma variação, só que aí eu aproveitei o sklearn [...].	Modificação de algoritmos de frameworks para novos tipos de problema.	Atividade [Treino do modelo]

Trecho	Código	Categoria
Bom. A gente trabalha na <<compahia>>, a gente trabalha com equipes bem interdisciplinares, né? E a gente trabalha - pelo menos no laboratório do Rio, hoje - com problemas muito aplicados a diferentes indústrias.	Equipes multidisciplinares	
	Problemas de diferentes indústrias.	
Então principalmente nessa parte, quando você tá trabalhando com ciência aplicada, você tá trabalhando em cima de um domínio - no geral que você não conhece necessariamente esse domínio.	Em geral desenvolvedores tem pouca experiência no domínio	Desafio [Geral]
Por exemplo: você pode estar trabalhando no domínio de meteorologia, no domínio de bioinformática, no domínio de petróleo e gás - que gente acaba necessitando muito do especialista do domínio.	Dependência do desenvolvedor com o especialista do domínio	Desafio [Geral]
	Presença do especialista do domínio	Ator [Geral]
	Presença do desenvolvedor.	Ator [Geral]
Então, por exemplo, a parte de Model Requirements, acaba sendo uma decisão colegiada né? Assim, juntam-se todos os especialistas em machine learning, especialistas de domínio, e no geral a gente vai fazer um modelo requerido pelos cientistas de domínio, então provavelmente os requisitos dos problemas serão trazidos por eles. Mas enfim, de quem partiu o problema vai ser levantado isso [os requisitos], e isso acaba tornando - essa parte vermelha [Model Requirement] - uma decisão mais colegiada.	Todos os envolvidos elicitam os requisitos do modelo em uma decisão colegiada	Atividade [Requisitos do modelo]
	Requisitos coletados de quem originou o problema.	Método [Requisitos do modelo]
Essa parte de Data Collection, Cleaning e Labelling, geralmente fica [com] o especialista de domínio. Porque enfim, ele que conhece o domínio do dado, ele que geralmente traz o dado, então geralmente fica com o cientista de domínio.	Processamento de dados executado pelo especialista do domínio	Atividadade [Processamento de dados]
A parte de Feature Engineering, ela é meio complicado hoje, né? Porque a gente tá trabalhando, grande parte a gente pode trabalhar com… A gente pode fazer propriamente a Feature Engineering do dado, extrair os dados ou a gente pode aprender representações em cima desses dados né? Que entra nesse contexto de Deep Learning	Atributos podem ser coletados diretamente dos dados	Método [Engenharia de atributos]
	Atributos podem ter representações aprendidas nos dados.	Método [Engenharia de atributos]
No geral os modelos mais novos fazem a parte de Feature Engineering junto com Model Training, a gente monta lá o nosso modelo, a arquitetura do nosso modelo e faz tudo junto. Então seria uma coisa mais do machine learning engineer ou do data scientist, não sei qual o termo melhor para você usar, mas aqui ficaria mais o machine learning engineer.	Modelos mais novos fazem a engenharia de atributos e o treino de forma única	Método [Engenharia de atributos]
	Engenharia de atributos de modelos mais novos é feita pelo desenvolvedor.	Atividade [Engenharia de atributos]
No contexto de métodos tradicionais de machine learning, onde a gente tem que pegar feature mesmo, fazer a feature, aí eu acho que já é uma questão de colaboração. Ou essa feature vem inteira junto com o dado ou é uma feature combinada, do tipo, o cientista de domínio ele me aponta o que é importante e eu com os conhecimentos de machine learning vou obter aquela informação, seja ela uma imagem, fazendo, mudando, extraindo características da imagem, seja lá o que for. Então tem essas duas abordagens.	Modelos mais tradicionais não fazem a engenharia de atributos	Método [Engenharia de atributos]
Se for uma abordagem mais de apreender a representação do dado, então fica só para mim. Se for uma questão de ter que escolher uma feature fica com os dois, tanto o machine learning engineer quanto o especialista do domínio.	Utilizando modelos novos a engenharia de atributos é feita pelo desenvolvedor.	Atividade [Engenharia de atributos]
	Utilizando modelos tradicionais a engenharia de atributos é feita pelo desenvolvedor e o especialista de domínio.	Atividade [Engenharia de atributos]
A parte do [Model] Training e Evaluation ela é senão totalmente, em grande parte feita	Treino e avaliação do modelo é feita na maioria dos casos pelo desenvolvedor	Atividade [Treino do modelo]

A parte do [Model] Training e Evaluation ela é senão totalmente, em grande parte feita só pelo machine learning engineer. Pode acontecer em situações específicas por a gente tá trabalhando com workflows complexos, workflows no sentido do problemas, né? Problemas complexos, que a gente possa necessitar do especialista do domínio para entender um certo comportamento que esteja acontecendo no [Model] Training. Ou tentar visualizar uma, uma, uma, uma circunstância que está acontecendo no [Model] Evaluation e para isso a gente pode recorrer a um especialista domínio, mas no geral essa é uma parte que fica para o machine learning engineer.	Especialista do domínio pode participar do Treino e Avaliação do modelo para opinar sobre algum comportamento que está ocorrendo.	Atividade [Treino do modelo]
Modelos mais tradicionais não fazem a engenharia de atributos	Deployment do modelos em soluções mais complexas é feita por equipes de infraestrutura.	Atividade [Implantação do modelo]
Mas também essa parte de deployment, dependendo da sua infraestrutura, se a gente pegar por exemplo o Watson Studio, ela é invisível, porque aí o machine learning engineer vai desenvolver a aplicação dele, o notebook dele, e vai carregar para o Watson Studio e o Watson Studio se encarrega do Deployment. Então se esse Deployment foi feito através de uma ferramenta como essa, fica só o machine learning engineer.	Deployment em plataformas para machine learnig é feita pelo desenvolvedor.	Atividade [Implementação do modelo]
Agora se você vai fazer um Deployment no cliente, em um cluster, em uma coisa mais com o arcabouço de produto, aí você vai precisar de um cara que entenda mais dessa questão de Kubernetes, Docker, etc. Que é esse pessoal de Devops.	Presença da equipe de infraestrutura.	Ator [Implantação do modelo]
	Equipes de infraestrutura precisam dominar tecnologias específicas.	Limitação [Implantação do modelo]
	Equipe de infraestrutura para implantação do modelo.	Atividade [Implantação do modelo]
Olha acaba sendo em duas vias, porque depende muito do problema que a gente tá trabalhando. Se eu tô trabalhando em problemas com ML, que são problemas que enfim, a gente não tá interessado em resolver "um problema real", a gente tá interessado em fazer pesquisa em desenvolver um novo método, novo modelo, a gente acaba trabalhando em todo processo.	Único desenvolvedor em todas as etapas para soluções cinetíficas	Limitação [Geral]
Entretanto, [para soluções reais] por exemplo, a parte vermelha [Model Requirements] e a parte de dados [Data Collection, Cleaning and Labelling] já são dadas. Você tá trabalhando em um problema científico, onde já tenho dataset pronto, já tem o problema muito claro, você só tem que ir lá e fazer o modelo. Existem algumas questões que você precisa mudar um pouquinho, mudar alguma coisa no dataset, ou fazer uma variação nesse dataset, daí você acaba pintando um pouquinho no amarelo [Data Collection, Cleaning and Labelling]. No geral fica concentrado no azul [Feature Engineering], verde [Model Training] e roxo [Model Evaluation]. E na prática, se a gente tá trabalhando em um grupo plural como é esse da <<compahia>>, aí de fato a gente majoritariamente trabalha mais na, no azul [Feature Engineering], verde [Model Training] e roxo [Model Evaluation].	Desenvolvedor atuanas apenas na construção do modelo em casos de aplicações reais.	Limitação [Geral]
Por que as outras duas partes terão outros cientistas responsáveis por isso.	Para de aplicações reais etapas de dados e implantação são executadas por outras equipes.	Limitação [Geral]
O Model Training. Acaba sendo muito aquilo que eu falei, né? Se a gente tá trabalhando com aprendizagem de representação, o Model Training é o design do nosso modelo e acaba entrando nessa nessa caixinha. Então acho que é mais ou menos isso.	Entrevistado considera ter mais experiência no treino do modelo.	Limitação [Treino do modelo]
a gente desenvolve o código do modelo e utiliza a própria parte de treinamento dos frameworkso o .train do TensorFlow, PyTorch. E aí você faz um cenário pequeno na sua máquina, você coloca só um teste, né? Acaba fazendo assim só para ver se tá tudo funcionando e depois você transfere provavelmente esse modelo para um ambiente capaz de rodar isso, pra um cluster de GPU e etc. Aí você configura os hiperparâmetros propriamente para o seu modelo, e você segue o mesmo processo lá de executar o comando de treinar e deixar ele rodando lá até treinar o modelo	Uso de frameworks para a criação de modelos: TensorFlow, PyTorch.	Método [Treino do modelo]
	Cenários menores para teste do treino	Método [Treino do modelo]
	Transferência do código de treino para infraestruturas maiores, capazes de executar.	Método [Treino do modelo]

Exatamente através da etapa seguinte, do Model Evaluation [...]. Aí você avalia com base em uma métrica, uma acurácia, uma precisão, um recall, depende do que foi definido para o problema.	Avaliação dá indicios da corretude do modelo	Limitação [Avaliação do modelo]
	Métricas clássicas utilizadas para avaliar o modelo.	Atividade [Avaliação do modelo]
	Uso de métricas clássicas.	Método [Avaliação do modelo]
Caso seja um problema complexo, novamente, um problema do mundo real, às vezes uma métrica, um precision, um recall não é suficiente. Você também vai precisar pegar esse resultado, compilar esse resultado e mostrar para cientista de domínio, para ele avaliar aquele resultado. Às vezes a precisão e o recall estão bons, mas tem outro aspecto do dado que o produto não foi interessante e só quem vai saber te dizer isso é o especialista de domínio.	Métricas classicas podem ser insuficientes para avaliar aplicações reais	Limitação [Avaliação do modelo]
	Definir métricas adequadas para avaliar o problema.	Desafio [Avaliação do modelo]
	Algumas características do modelo podem só ser observadas pelo especialista.	Limitação [Avaliação do modelo]
	Especialista de domínio avalia o modelo.	Atividade [Avaliação do modelo]
é a parametrização desse modelo, a parametrização eu acho que é mais difícil depois do design da arquitetura, né?	Dentro a definição de hiperparâmetros é a etapa mais desafiadora.	Desafio [Treino do modelo]
Porque enfim, é extremamente empírico, você não tem não tem como medir. [Você pensa] "Ah eu vou botar aqui 0.5, porque ou é o que todo mundo usa ou é porque a sua experiência te disse que é assim". Mas no fim você acaba tendo que fazer uma variação de determinado conjunto de parâmetros, para achar o modelo [leia-se valores de hiperparâmetros] que melhor se adéqua ao seu dado e você ter um produto melhor, claro, tomando cuidado para não ter overfitting e esse tipo de coisa.	A definição de hiperparâmetros é um processo empírico.	Atividade [Treino do modelo]
depende do experimento que você está executando, né? No geral você faz um cross-validation, um team cross-validation, alguma coisa assim, se você tá fazendo um experimento mais aprofundado, onde você quer resultados mais confiáveis.	Cross-validation para resultados mais confiáveis.	Método [Avaliação do modelo]
Ou você faz só um split aleatório dos seus dados, divide lá em 60% treino, 20% teste, 20% avaliação. Você faz alguma métrica [leia-se divisão] simples dessa, mas sempre aleatório.	Separação da base de treino, teste e validação em proporções tradicionais para soluções gerais.	Método [Avaliação do modelo]
	Separar os dados em treino, teste e validação.	Atividade [Treino do modelo]
essa [divisão em] 60%, 20% e 20% é uma das mais clássicas, que todo mundo usa. [Sendo que] a seleção aleatória é fundamental, a seleção inicial aleatória no conjunto total de quais serão os 60% tem que ser aleatória, entendeu? Quem serão os 60%? Quem serão os 20? E quem serão os outros 20%? [Entretanto] Essa proporção varia muito do dado, a clássica é 60%, 20% e 20%.	Seleção aleatória de treino, teste e validação é fundamental.	Método [Avaliação do modelo]
[Essas abordagens] ficam para soluções mais científicas, se a gente tá em um teste inicial, no primeiro momento você faz a mais simples, mas depois "terminamos, vamos escrever o paper", aí você faz um cross validation.	Cross-validation apenas ao fim para dar mais confiabilidade.	Limitação [Avaliação do modelo]
geralmente o dado que você usa para treino, ele vêm do cenário real, né? Ou algo muito próximo do cenário real. O que você pode fazer no processo de Data collection é fazer avaliações estatísticas em cima do seu dado, né? Você analisa a distribuição dos dados e etc. No treino do modelo você já não tem muito mais o que fazer, o dado já tá pronto, você só vai dividir ele.	Dados de treino oriundos do mundo real	Método [Processamento de dados]
	Estatística usada para verificar a representatividade do dado.	Método [Engenharia de dados]
	Uso de estatítica na engenharia de atributos.	Atividade [Engenharia de atributos]

Você tem que assumir que o dado já segue uma distribuição uniforme, porque você vai fazer uma seleção aleatória, agora se você sabe de antemão que o seu dado não segue uma distribuição aleatória que, você tá no momento de treino e ainda tem que fazer alguma coisa então, você faz um data augmentation, um sampling do seu dado, pra você tentar mitigar [a possibilidade de] que o seu modelo gere algum bias em relação ao mundo real, entendeu? Então você vai deixar o seu dado o mais uniforme possível e você assume que ele vai refletir o mundo real.	No treino se assume que o dado é representativo para o mundo real	Limitação [Treino do modelo]
	Uniformizar o dado de treino de modo que se assuma que ele reflete o mundo real.	Método [Engenharia de dados]
então, o algoritmo eu acho que …, eu acho que ambos são muito na intuição, né? Você tem um problema, você estuda o seu problema, você vê o que a maioria das pessoas estão desenvolvendo e você desenvolve em cima de um algoritmo similar ou alguma ideia próxima. Se ficar no contexto de pesquisa, beleza, é o que a imaginação permitir.	Literatura define pontos de partida para algoritmos e hiperparâmetros	Limitação [Treino do modelo]
A questão de hiperparâmetros, ela é muito como eu falei antes, né? Ela é muito experimental, não tem muito como você definir. Você sabe um conjunto básico de parâmetros, você sabe que uma determinada distribuição de dados [leia-se parâmetros] funciona por exemplo, no contexto de embeddings [onde] a gente a gente usa uma distribuição de dados um pouquinho diferente da uniforme, porque alguém fez um teste um dia e viu que ela é melhor do que criar todos os vetores aleatoriamente. A gente utiliza essas coisas porque enfim, acelera o processo. Mas é extremamente experimental, pode acontecer de amanhã eu achar uma outra distribuição inicial que enfim, ajuda a treinar mais rápido meu modelo, que ajuda ele a convergir mais rápido e que foi descoberta a partir de pura experimentação.	A definição de hiperparâmetros é um processo empírico.	Limitação [Treino do modelo]
sim, a gente sempre compara com baseline sim. Aí no contexto de ciência tem os baselines típicos, né? Que são os estados da arte, se você está construíndo um novo modelo, você compara com esses baselines, tentando bater o estado da arte.	A literatura define os baselines para soluções científicas.	Limitação [Treino do modelo]
Em um domínio de mundo real já é mais complicado, porque não necessariamente alguém já trabalhou naquele problema. Se você tá na indústria no geral, ninguém fez aquela solução e aí você tem os próprios baselines, você pega soluções que você sabe que são soluções tradicionais para aquele problema, implementa, toma isso como base e aí desenvolve novas abordagens, comparando com esses baselines.	Aplicações reais possuem próprios baselines que devem ser buscados para comparação.	Método [Treino do modelo]
já surgiu oportunidade mas nunca usei não, porque demora muito.	Otimizadores de hiperparâmetros consomem muito tempo.	Desafio [Treino do modelo]
assim, não necessariamente manualmente, porque a gente faz a experimentação, por exemplo, define lá um range específico. Mas aí o range quem define é a gente, então essa parte é manual. Varia em cima de tudo aquilo [range] mas é manual.	Variação de hiperparâmetros é feita dentro de uma extensão de valores.	Limitação [Treino do modelo]

enfim, eu nunca trabalhei com problemas que eu tive isso como resultado, então eu nunca resolvi um problema desse. Mas no geral quando você tem problemas que vê que são da natureza do dado, você tem que mudar o seu conjunto de dados.	Problemas da natureza dos dados são resolvidos reexecutando o processamento de dados para obter um novo conjunto	Método [Processamento de dados]
[Supondo que] Você fez um modelo para uma série temporal e essa série temporal muda de tempos em tempos, por exemplo: nenhum modelo temporal vai funcionar agora durante a época do coronavírus, porque enfim, não se teve quarentena desde a gripe espanhola. Em um caso desses, ou você sabe que o seu modelo tem essa natureza, que é um modelo temporal, então você vai tomar abordagens de aprendizagem online, aprendizagem por reforço, etc.	Soluções com caracteristicas temporais requerem tipos específicos de aprendizagem.	Limitação [Treino do modelo]
[...] Ou você assume que seu modelo - que aí já aconteceu comigo na prática em um problema de indústria - é temporal mas pode ser estacionário no sentido de que, eu posso atualizar esse modelo anualmente e não vai ferir nada. Então [...] é interessante você treinar o seu modelo em batches, de tempos em tempos. Você treina o modelo, e faz o deployment do novo modelo, que é uma solução um pouco mais custosa.	Modelos estacionarios e temporais devem são retreinados com certa frequencia.	Restrições [Treino do modelo]
Você pode utilizar uma metodologia mais técnica. "Ah eu sei que meu modelo gira em torno de uma determinada [...] métrica", e ele consistentemente foge [dos valores esperados], aí você pode fazer uma distribuição normal [...] se você vê que a pontuação do seu modelo naquela métrica muda, isso pode indicar [...] uma necessidade de retreinar.	Métricas podem ser utilizadas para monitorar um modelo implantado	Método [Implantação do modelo]
Mas no geral, na prática você tá muito orientado ao domínio, você tem um domínio onde ele claramente te permite marcar um batch, ou seja, "eu vou fazer isso de seis em seis meses, eu vou fazer isso anualmente". [Isso pode ser causado] até mesmo por necessidade [leia-se restrições] no deployment. Se pessoas tiram férias em dezembro e o cliente precisa do modelo funcionando sempre, não tem como eu ficar fazendo deployment. Então eu chego lá com o papai noel, treino o modelo e faço o deployment. Ele [o cliente] não vai sentir nada. Aí você tem essas decisões de negócio.	O periodo para retreinar o modelo está associado ao domínio.	Restrições [Treino do modelo]
via experimentação, você compara os seus algoritmos com múltiplos algoritmos que você esteja desenvolvendo e também com as baselines, para ver se ele melhora as baselines, se o algoritmo é de fato melhor.	Corretude de algoritmos é verificada através da comparação com baselines	Método [Avaliação do modelo]
	Comparar algoritmos com baselines	Atividade [Avaliação do modelo]
A parametrização é a comparação dele com ele mesmo com diversos tipos de parâmetros, né? Então você roda em diferentes parametrizações, o que tiver a melhor métrica, você adota.	A corretude de hiperparâmetros é verificada através da variação de hiperparâmetros e comparação de resultados de avaliação	Método [Avaliação do modelo]
depende muito do contexto, no geral você vai utilizar um framework, um Tensorflow, um Sklearn, um ..., qualquer coisa assim. Aí se você vai usar frameworks, que são desenvolvidos em cima desses frameworks, como é o caso de …. Sei lá, você está trabalhando no domínio de aprendizagem de representação em grafos de conhecimento, tem algumas bibliotecas específicas, algumas implementações que o	Uso de frameworks para a criação de modelos: TensorFlow, Scikit-learn, PyTorch.	Método [Treino do modelo]

pessoal já desenvolve em cima, aí você adota uma dessas que já facilite o seu processo, e você treina em cima delas. Mas no geral, se você tá fazendo algo muito novo, você vai desenvolver a sua metodologia, você que vai codar [desenvolver o código para] o treino e a validação.	Soluções novas requerem desenvolvimento do zero.	Limitação [Treino do modelo]
: assim, você testa. Você pega um TensorFlow e um PyTorch, é o que a comunidade usa, você assume que eles estão certos, enfim. Você assume que está certo e é isso. Dificilmente você vai fazer casos de teste para a biblioteca, para ver se ela tá realmente bem implementada, ou implementada de modo correto. O máximo que você pode fazer na prática é olhar lá o fórum de discussão, olhar as issues e vê onde tem problema que a comunidade já detectou. E você tá desenvolvendo, detectou um problema, vai lá e reporta.	Confiança na corretude da implementacão de frameworks.	Limitação [Treino do modelo]
você olha pela métrica que você definiu, não tem … É quase impossível você fazer um debug em um modelo que treina, porque enfim, a inicialização desses modelos por vezes é aleatória, então você não tem nenhuma garantia de que você vai conseguir enxergar o mesmo cenário todas as vezes, é tudo estocástico, então você não tem como debugar.	Característica aleatória do modelo dificulta a verificação.	Desafio [Treino do modelo]
O que você faz é, você roda o modelo e vê o resultado dele, se o resultado estiver bom, você provavelmente acertou, se não estiver, ou seja, ele para de funcionar em alguma parte, você evidentemente sabe que tem um erro. Mas senão se ele estiver ruim, ou você acha que ele tem um erro ou ele só é ruim mesmo. É muito do feeling do desenvolvedor, não tem forma de fazer isso.	A indicação de corretude de um modelo é observada na avaliação.	Limitação [Avaliação do modelo]
executo através dos frameworks de machine learning, não é muito diferente disso não.	Uso de framework para avaliação do modelo.	Método [Avaliação do modelo]
você pode aplicar em uma outra parte do dado, se você tiver uma parte não observada do dado, você pode fazer uma anotação em cima desse dado, e executar o seu modelo em cima desse dado.Aí você garante empiricamente que o seu modelo funciona.	Uso da base de validação para verificação final do modelo.	Atividade [Avaliação do modelo]
Ou você faz análises estatísticas, você tira diferentes métricas em cima do seu modelo, faz uma análise sobre a questão de assertividade do seu modelo com base em determinada característica ou determinada label do dado, etc. Você tem essas duas formas.	Uso de métodos estatísticos para avaliação do modelo.	Atividade [Avaliação do modelo]
dependendo do seu domínio, se você tá trabalhando em um modelo científico, quem vai definir essa [medida] são os baselines. Você tem os baselines lá e eles vão te dizer qual é o [valor] suficiente.	Literatura define alvos de métricas para modelos científicos.	Limitação [Avaliação do modelo]
Quando você tá trabalhando em um cenário real, você não tem esse [valor] dado. Você vai ter que ..., por exemplo, você vai ter uma acurácia de 70% às vezes, e vai estar imaginando que só 99% que serve. Mas os cientistas de domínio [...] vão olhar aquele produto e vão ver que aquilo é bom, muito melhor do que existia antes. Às	Soluções reais não possuem alvos para as métricas	Desafio [Avaliação do modelo]

vezes eles tinham que fazer na mão, passavam dias, e ele aceita perder esses 30% de acurácia e não ter que fazer nada, talvez fazer um fine tuning manual depois. Então tem essas duas óticas no geral. No prática você vai precisar do dono do dado, o cientista do domínio, e no caso científico você tem as baselines lá.	A avaliação do especialista do domíno pode ser mais importante do que os valores de métricas.	Limitação [Avaliação do modelo]
fazer justamente isso que você perguntou. Garantir que ele é um bom modelo para um resultado prático. Garantir que ele não tem nenhum bias e etc.	Garantir que o modelo é bom para fins práticos é o maior desafio da avaliação do modelo.	Desafio [Avaliação do modelo]
ele é muito auxiliado, quer dizer, ele sempre vai ser auxiliado dos cientistas de domínio, entendeu? Pelo menos de alguma pessoa que entenda de domínio.	Especialista de domínio auxilia a engenharia de atributos.	Atividade [Engenharia de atributos]
No geral eu sigo a primeiro modo as técnicas clássicas de fazer Feature engineering. Por exemplo, trabalhando com imagens eu vou tentar usar a primeira coisa que é um shift, se não foi suficiente, eu pego essa mesma imagem eu combino com outras características que eu extraío dela. E aí você vai compondo lá o seu vetor de features, né? Da forma que mais lhe agrade e que melhor se adéqua o problema.	Extração de atributos se inicia de acordo com técnincas clássicas	Método [Engenharia de atributos]
Não tem…, é muito na experimentação e no que já existe sobre o tema. Ou [então] o cientista de domínio ele fala, "Ah não, para esse dado aqui só interessa você olhar a intensidade da cor vermelha", beleza. Você faz lá o filtro em cima da imagem, que extrai a cor vermelha e pronto, acabou. Então tem essas duas abordagens.	Desenvolvedor executa a engenharia de atributos.	Atividade [Engenharia de atributos]
	Desenvolvedor não conhecimento específico sobre o dado.	Desafio [Engenharia de atributos]
	Especialista de domínio auxilia a engenharia de atributos.	Atividade [Engenharia de atributos]
quando é avaliado. Você só vai saber se o seu input é suficiente, se o seu modelo ele consegue aprender em cima daquele dado e você só sabe se o seu modelo aprendeu algo sobre aquele dado, quando ele te dá uma métrica.	Incerteza sobre a corretude da engenharia de atributos	Desafio [Engenharia de atributos]
	A avaliação da indicio sobre a corretude da engenharia de atributos.	Limitação [Engenharia de atributos]
Feature engineering é um processo bem complicado, primeiro que você tem que entender o domínio e depois de você entender o domínio, você tem que identificar quais são os pontos que farão o modelo estatístico - que no caso é nosso modelo de machine learning - ter melhor eficiência para entender a representação daquele dado.	Necessidade de compreender o domínio	Desafio [Engenharia de atributos]
A Feature Engineer faz justamente o processo de representar o dado. A gente tá transformando uma imagem em um vetor de características, um vetor de pontos flutuantes, e o seu modelo vai ter que entender que aqueles pontos flutuantes refletem determinado símbolo. Você fazer esse mapeamento manualmente é extremamente complexo, você tem que conhecer o domínio, você tem que conhecer técnicas para fazer isso. Em geral é isso, essa é maior dificuldade.	Dificuldade em garantir que as features são relevantes para o modelo aprender a representação desejada.	Desafio [Engenharia de atributos]

Trecho	Código	Categoria
é... bom, no caso dos grupos que eu trabalhei, as pessoas são relativamente similares em cada uma dessas partes, em termos de background. Todo mundo trabalha um pouco na parte de coleção de dados, né? Digo, tanto para limpar ou adequar o dado, isso realmente é... acho que todo mundo tem que trabalhar de certa forma com isso.	Todos participam da etapa de Processamento de dados.	Atividade [Processamento de dados]
	Todos participam da Engenharia de atributos.	Atividade [Engenharia de atributos]
Na parte de Feature Engineering - eu particularmente - na minha equipe estamos trabalhando mais com imagem, então normalmente tem uma galera para tratar os dados, [estes] têm um background de processamento de imagens, gente que estudou isso mais a fundo.	Presença de especialista em processamento de imagem	Ator [Engeharia de atributos]
	Engenharia de atributos auxiliada pelo especialista em processamento de imagem.	Atividade [Engenharia de atributos]
Na parte de modelos, criação de modelos e treinamento de modelos [Model Training] - normalmente mais redes neurais - aí também é a galera que - vamos dizer assim - está mais acostumada com… a mexer com o TensorFlow, mexer com o Keras. Eu normalmente estou nessa parte sempre e acho que seria isso. No meu grupo, acho que seria difícil separar isso assim em pessoas. Acho que essa é a melhor separação que eu consigo fazer.	Presença do desenvolvedor	Ator [Treino do modelo]
	Uso de Frameworks para a criação do modelo: TensorFlow, Keras.	Método [Treino do modelo]
	Desenvolvedores atuam com frequência no Treino do modelo	Atividade [Treino do modelo]
sim, normalmente sim. Tem sempre uma... O grupo como um todo pode até se dividir razoavelmente, mas tipo assim... é possível que uma pessoa só trabalhe do início ao fim em todas essas etapas - ou praticamente do início ao fim - ou o grupo todo se divida, é... para fazer cada um uma parte desse processo, mas no geral, pelo menos todos têm experiência em cada uma das partes desse processo todo.	Unico desenvolvedor participa de todas as etapas.	Limitação [Geral]
	Desenvolvedores diferentes para cada uma das etapas.	Limitação [Geral]
eu acho que mais no Model Training, Model Evaluation e Data Collection, Cleaning e Labelling. Não exatamente o collection [Data Collection, Cleaning e Labelling], né? Mas vamos dizer assim, é... talvez seja... acho que não no collection, vou botar no engineering [Feature Engineering], porque o dado já foi coletado, mas a gente tem que trabalhar no... na limpeza dele, mas em mais alto nível, então vou colocar no Feature Engineering, Model Training e Model Evaluation.	Desenvolvedor atua apenas na construção do modelo.	Limitação [Geral]
é... eu diria que talvez... é complicado... pelo menos para mim. Dificilmente eu pego um modelo com dado pronto, dificilmente eu mexo no dado e não treino o modelo, eu diria que é bem similar mas talvez com um pouco mais de... da parte de modelo, da criação de redes e treinamento de redes.	O desenvolvedor raramente se detem apenas a uma etapa.	Limitação [Geral]
	Entrevistado considera ter mais experiência no Treino do Modelo.	Limitação [Treino do modelo]
bom, eu tô incluindo nesse... não sei se deveria, mas eu tô incluindo nesse treinamento a parte de... eu não trabalho muito com modelos de ML em geral, eu trabalho com rede neural, então tem a parte de criação do modelo em si, e o treinamento dele.	O Treino do Modelo também inclui a criação dele.	Atividade [Treino do modelo]
Basicamente, eu começo vendo qual o tamanho do problema, qual tipo de rede eu vou ter que usar, mais ou menos o tamanho dela, se eu vou precisar... se eu consigo usar uma rede que já existe, isso é a primeira coisa para testar, fazer os testes iniciais.	A primeira atividade é analisar o problema e estimar a rede	Atividade [Treino do modelo]
	A princípio tem que verificar se existe alguma rede que possa ser reusada.	Atividade [Treino do modelo]
Se não rola realmente de cara, parte para tentar desenvolver o modelo que consiga resolver o meu problema. Aí eu diria que existe um loopzinho, design e treinamento, design e treinamento, pra tentar chegar no modelo que resolva o meu problema.	O modelo é criado em um loop design-treino.	Atividade [Treino do modelo]
Normalmente - nas experiências que eu tive, nos projetos que eu trabalhei - dificilmente eu consigo resolver o problema com algo que já esteja pronto e que eu não precise fazer algum tipo de modificação.	Dificilmente algum modelo poderá ser reusado sem adaptações.	Limitação [Treino do modelo]

Dado os requisitos do problema, as vezes a gente acha que pode usar um modelo, e o resultado não é tão bom, tem que melhorar o resultado, então a gente tem que melhorar esse modelo, então eu diria que sempre começa do mais simples e vai evoluindo conforme os requisitos, mas ficaria nesse loop de: [desenvolvimento do] modelo , treino [do modelo] e volta. É basicamente isso.	O reuso pode parecer plausível em um primeiro momento dados os requisitos mas em seguida se verificar que não.	Limitação [Treino do modelo]
é... bom.... depende - vamos dizer assim - do que eu tô trabalhando, mas se existe uma base de teste clara, digo assim, que já é pronta - vamos dizer assim - existe uma coleção de dados de teste que não fui eu que preparei, eu simplesmente rodo o modelo na base de teste e descubro se aquilo é o suficiente.	Bases de validação utilizadas para testar o modelo	Atividade [Avaliação do modelo]
Tem outros problemas que eu já... estive e estou, em vários projetos no qual a própria avaliação é difícil, ela é uma coisa mais indireta, não está relacionada por exemplo a acurácia do modelo, ou alguma coisa assim, é uma coisa mais indireta, então depende de outras coisas e aí é um processo mais difícil de saber se a rede responde ou não ao que era esperado.	Dependendo do projeto a avaliação do modelo é complexa	Desafio [Avaliação do modelo]
	Métricas clássicas insuficientes para valiar o modelo	Limitação [Avaliação do modelo]
	Dificuldade em garantir a corretude do modelo através de métricas.	Desafio [Avaliação do modelo]
A avaliação normalmente é a parte mais difícil nesses projetos, porque justamente ela as vezes é até um pouco subjetiva, e a gente não tem como treinar o modelo subjetivamente. A gente treina ele [o modelo] da maneira que a gente consegue treinar, de alguma maneira supervisionada ou não supervisionada que a gente conheça, e aplica as métricas de requisitos do projeto. Isso seria - vamos dizer assim - o caso mais real, que eu estou assim... mais acostumada a lidar, mas existem casos que a base de teste é bem definida e você consegue diretamente avaliar o seu modelo.	A avaliação do modelo pode ser afetada pela da subjetividade da análise.	Desafio [Avaliação do modelo]
	Métricas especiais para a avaliação do modelo podem ser especificadas junto com os requisitos.	Limitação [Avaliação do modelo]
é... os maiores desafios... por mais que nós tenhamos avanços nas GPUs, elas são cada vez mais poderosas, muitas vezes... tem que se desdobrar para treinar determinado modelo em determinado dado. Eu já tive que rodar modelos em 20 máquinas diferentes, para conseguir rodar em determinado dado, ou mais: 20, 40. Acho que o máximo foi 120 máquinas. Isso é um desafio, um desafio não só da infraestrutura em si, pode ser muito complicado, mas também de botar isso em pé, né?	Alguns modelos precisam de infraestrutura massiva para serem treinados.	Limitação [Treino do modelo]
	Gerenciar uma grande infraestrutura para treino de um modelo é um desafio.	Desafio [Treino do modelo]
Por mais que a gente tenha biblioteca que facilite bastante isso, se a pessoa não tem um background, ou pelo menos vontade de aprender a distribuir essas coisas... enfim, é um problema. Isso é um desafio bem grande.	Uso de bibliotecas para configurar a infraestrutura de treino para o modelo.	Métodos [Treino do modelo]
Outro desafio que eu enxergo é que... às vezes se gasta muito tempo tentando desenvolver o modelo especificamente para um determinado problema, e isso pode demandar muito tempo e aí se esse dado muda um pouquinho, a gente tem que ter esse trabalho todo de novo. Isso ainda é uma coisa problemática e ainda não está super bem resolvida, e... atrapalha a gente a manter esses modelos.	Alto custo em termos de tempo para a criação de um modelo	Desafio [Treino do modelo]
	Data drift torna necessário retreinar todo um modelo.	Atividade [Treino do modelo]
	Retreinar um modelo com data drift implica em um alto custo de manutenção.	Desafio [Treino do modelo]
então isso depende muito do problema, depende mesmo.	Limitação [Avaliação do modelo]	Limitação [Treino do modelo]
Normalmente, quando a gente tá lidando com coisas que são razoavelmente bem comportadas, é possível fazer uma seleção aleatória, coisa assim e tal, mas por exemplo eu tenho que trabalhar...	Seleção aleatório é utilizada na escolha das bases de treino, teste e validação para problemas comuns	Método [Avaliação do modelo]

Normalmente eu tenho que trabalhar com dados espaciais, então eu não posso escolher qualquer ponto aleatoriamente, tem que ter algum... tem que garantir por exemplo que o teste esteja fisicamente longe do treino, senão eu não estou testando razoavelmente o meu modelo. Ou garantir que o, se eu tenho um dado espacial que eu, na hora que eu fui selecionar o meu dado de treino eu consegui cobrir boa parte do meu dado, então normalmente eu tento aplicar... enfim, seleção aleatória dentro das restrições que o dado me coloca, né? No meu caso essas restrições têm a ver com a relação espacial das amostras, ou coisa assim.	Regras de domínio devem ser consideradas para a seleção das bases de treino, teste e validação	Limitação [Treino do modelo]
: olha... normalmente eu uso... isso é um pouco difícil de dizer... mas normalmente eu costumo trabalhar com 20% do que eu tenho disponível para treinamento, para validação.	Separação da base de treino, teste e validação em proporções tradicionais para soluções gerais.	Método [Treino do modelo]
	Seleção das bases de treino, teste e validação	Atividade [Treino do modelo]
No caso do teste é um pouco mais complicado, nos trabalhos que eu tenho que desenvolver, nos projetos, né? O teste não é uma base que você separa... enfim, te dão um monte de coisa para você testar, então normalmente a base de teste que eu tenho que rodar o meu modelo é muito maior do que a base de treino ou a de validação. Não sou eu que defino isso, né? Normalmente o teste é uma coisa que... vamos dizer assim, você tem que rodar esse seu modelo para esse determinado dado, mas em um ambiente controlado, eu diria que costumo separar para teste pelo menos uns 20% ou 30%.	Dados de teste e treino são repassados de maneira independente pelo cliente	Limitação [Processamento de dados]
	O modelo é testado em cerca de 20% ou 30% dos dados recebidos.	Atividade [Avaliação do modelo]
como eu falei depende muito do problema, mas sim já usei bastante cross-validation, já usei simplesmente seleção aleatória, mas como eu falei, nos projetos que eu trabalho eu não posso fazer isso tão diretamente	Cross-folder para validar o modelo.	Método [Treino do modelo]
	Uso do cross-folder validation.	Atividade [Treino do modelo]
eu preciso respeitar algumas condições do dado, normalmente condições espaciais, então não é tão direto por exemplo, fazer um cross-validation.	Soluções reais podem ter regras que não permitem o uso de cross-folder.	Limitação [Treino do modelo]
é, garantir, garantir, ninguém super garante, mas a gente tenta... enfim, tenta pelo menos na hora que a gente usa... vai selecionar o... os dados de treinamento, que eles cubram boa parte do que se espera - vamos dizer assim - de a onde ele funciona.	Incerteza sobre se os dados de treino representam os reais.	Desafio [Treino do modelo]
	Regras do domínio devem ser respeitadas para aumentar a generalização do modelo.	Limitação [Treino do modelo]
que também no caso dos projetos que eu trabalho, o pré-processamento costuma ser muito importante, para remover problemas nos dados - vamos dizer assim - que dificultariam você usar, o mesmo modelo em outros dados, ou dados novos.	Processamento de dados importante para a generalização do modelo	Limitação [Processamento de dados]
	Engeharia de atributos importante para a generalização do modelo.	Limitação [Engenharia de atributos]
Muitas vezes, a gente... - como eu falei no caso dos meus projetos - isso não é possível, mesmo que eu quisesse treinar modelo e... não é qualquer dado... não é qualquer dado novo que eu vou conseguir um bom resultado. Existem - vamos dizer assim - restrições, muitas vezes do tipo de dado que vão me deixar ou não usar o modelo em outro lugar, mas a garantia, o que a gente pode garantir é o pré-processamento e tentar espalhar bem as... no caso desses problemas mais espaciais, as amostras de treinamento pelo dado.	Incerteza sobre se o modelo será bom para o cenário real.	Desafio [Treino do modelo]
então... por acaso meu tema de tese de doutorado foi Neural Architecture Search, então eu trabalhava com a seleção de arquiteturas automaticamente, também um pouco de seleção de hiperparâmetros, mas no dia-a-dia eu sempre começo com o default, de no caso o TensorFlow que é o que eu sou, pra ver se funciona. Se não funciona a gente pode partir para um método mais... é... de busca de parâmetros, alguma coisa assim, como grid search e tal. Mas em termos de arquiteturas, eu... o tema da minha pesquisa é justamente identificar essas arquiteturas automaticamente.	Processo automatizado para a seleção de arquiteturas	Método [Treino do modelo]
	Processo automatizado para seleção de hiperparâmetros	Método [Treino do modelo]
	A primeira opção em relação a hiperparâmentros e arquiteturas é testar alternativas padrão.	Atividade [Treino do modelo]
ntão... já cheguei a usar o meu próprio, normalmente o que eu mesmo trabalhei, mas no geral eu consigo fixar os parâmetros em um range razoável, não me vi na situação que eu precisasse fazer uma busca exaustiva de hiperparâmetros sem contar a arquitetura que eu tô falando.	Mesmo a seleção automática de hiperparâmetros possui auxílio de conhecimento empírico.	Limitação [Treino do modelo]

então... normalmente, pelo menos da maneira como eu normalmente trabalho a pessoa de domínio - o cliente - ele não entende nada de ML, então o expert do hiperparâmetro sou eu mesmo. Eu é que setto os parâmetros com base na minha experiência, e o cliente ele pode me dar dicas, de - vamos dizer assim - o que ele espera de resultado, e o que ele [espera]... , "olha, essa imagem que eu te dei era para sair de tal jeito".	Desenvolvedor define hiperparâmetros de acordo com sua experiência	Atividade [Treino do modelo]
	Presença do cliente.	Ator [Avaliação do modelo]
	Cliente auxilia na avaliação do modelo.	Atividade [Avaliação do modelo]
Aí eu tenho que avaliar isso e saber como eu devo trabalhar no meu modelo para poder chegar nesse resultado que ele espera, então normalmente a iteração que eu tenho com o usuário final é... são essas pessoas, pessoas que não trabalham... ou simplesmente usam ML em um nível... alto nível e não entendem muito bem desses hiperparâmetros.	Cliente não opina em características como arquitetura ou hiperparâmetros	Limitação [Treino do modelo]
é... normalmente eu não tive que lidar muito com esse tipo de problema, mas ele é um problema existente. Já aconteceu de a gente conseguir aplicar um fine tuning de um modelo e conseguir usar em outro dado, mas eu não... na minha experiência mesmo eu não tive que lidar com isso, eu diretamente.	Refinar um modelo pode fazê-lo superar um data drift.	Atividade [Treino do modelo]
é, normalmente não sou eu que monitoro, né? O cliente que tem o modelo, que usa, e que retreina se precisar. Então eu não tenho visão de como está o modelo atualmente, tipo isso.	Após a entrega o cliente é responsável pelo modelo	Limitação [Implantação do modelo]
	O cliente monitora o modelo	Atividade [Implantação do modelo]
	O cliente retreina o modelo caso necessário.	Atividade [Implantação do modelo]
normalmente eu não... não... não garanto isso não, infelizmente a vida de trabalhar com projeto é "o resultado é satisfatório? Sim ou não". Aí você tenta buscar, os... enfim, rodar mais parâmetros, busca de parâmetros, ou não mas normalmente eu não tenho essa garantia.	Incerteza sobre se a arquitetura escolhida é a melhor	Desafio [Treino do modelo]
	Incerteza sobre se os hiperparâmetros escolhidos são os melhores.	Desafio [Treino do modelo]
No caso, mais voltado para a minha pesquisa, a busca de parâmetros, ela está teoricamente tentando encontrar um bom modelo automaticamente, é... para determinada tarefa, mas ela também não garante que é o melhor modelo, ela só garante que é melhor do que muitos outros que você possa tentar usar.	Seleção automática de hiperparâmetros não garante que os parâmetros são os melhores	Desafio [Treino do modelo]
	Seleção automática de hiperparâmetros tem vantagem pois garante que os parâmetros escolhidos são melhores do que vários outros.	Limitação [Treino do modelo]
é, eu uso... costumo usar o TensorFlow e normalmente - vamos dizer assim - a casca do código é reprodutível, né? Uma vez que você tem mais ou menos o código de treinamento, você consegue usá-lo em vários lugares, mudando coisas, né?	Uso de Frameworks para criação de modelos	Método [Treino do modelo]
Mas sim, eu uso TensorFlow e uma vez que você desenvolve um código alí, você consegue reusar várias partes dele, da para realimentar essa coisa aí.	Reuso de partes do código de treino criado sobre frameworks.	Atividade [Treino do modelo]
já usei o PyTorch, não gosto e prefiro o TensorFlow.	Uso de Frameworks para criação de modelos: PyTorch, TensorFlow	Método [Desenvolvimento]
não.. nunca executei nenhum tipo de teste, dessas bibliotecas mais... Assim, alguns testes vagabundos eu até já fiz, mas assim, debugar para saber se está tudo perfeitamente implementado, nunca fiz isso,	Incerteza sobre a corretude da implementação do framework.	Desafio [Treino do modelo]
mas eu confio no tamanho da comunidade do TensorFlow, isso é uma das coisas que me faz gostar do TensorFlow, é que existe muita gente mexendo, e muita gente vendo problemas. Isso para quem está usando - pelo menos na minha opinião - é muito bom.	Confiança na corretude da implementacão de frameworks.	Limitação [Treino do modelo]
eu costumo fazer teste nos meus códigos - digo assim - existem funções de teste, mas. assim... elas garantem o básico. Agora eu assumo... que por exemplo, enfim, que a função de gradiente do TensorFlow ela funciona, eu não vou testar esse tipo de coisa, mas geralmente eu tenho sim funções de teste para as coisas que eu desenvolvo em cima da biblioteca dos outros, então digo assim, eu não vou checar se o TensorFlow tá fazendo, vamos dizer assim, a coisa certa lá por debaixo dos panos dele, mas onde eu tô mexendo, eu costumo fazer funções de teste sim.	Funções do framework não são testadas.	Limitação [Treino do modelo]
não, seriam testes de software mesmo, para saber se o código está correto, normalmente não envolve diretamente alguma avaliação do modelo não.	Testes de software executados para avaliar o código desenvolvido.	Atividade [Avaliação do modelo]

é... normalmente - como eu falei, nos projetos que eu trabalho - o dado cru, ele é algo que a gente normalmente não entende logo de cara, não é tão óbvio como uma base de reconhecimento de imagem, que você já sabe o que precisa fazer.	Dados são de dificil compreensão no primeiro momento.	Desafio [Engenharia de atributos]
Normalmente nessa parte tem uma grande interação com o cliente, ou quem vai consumir o modelo, para entender o dado e saber que tipo de processamento que a gente pode fazer com ele.	Presença do especialista de domínio.	Ator [Engenharia de atributos]
	Iteração com o especialista de domínio/cliente para entender o dado	Atividade [Engenharia de atributos]
a grosso modo eu não garanto, eu vou com o pipeline até o fim, a gente avalia para ver se está tudo dentro dos requisitos e fica nesse loop.	Incerteza sobre a corretude da engenharia de atributos	Limitação [Engenharia de atributos]
	A avaliação dá indicio sobre a corretude da engenharia de atributos	Limitação [Engenharia de atributos]
Isso volta sempre, o tempo inteiro, ainda mais quando a gente está lidando com dados que a gente não conhece muito bem - vamos dizer assim - não são coisas tão triviais, a gente tem que sempre avaliar o resultado e voltar, caso necessário para mexer nas features, isso realmente não - pelo menos na minha experiência - tem garantia não.	Processo iterativo entre engenharia de atributos e avaliação.	Atividade [Engenharia de atributos]
desafio... eu acho que assim... essa dificuldade de a gente ter que entender o domínio nessa hora. Muitas vezes a gente não entende com muita facilidade... enfim, porque a gente trabalha com a parte de desenvolvimento de modelo, mas constantemente a gente precisa aprender sobre o domínio para poder saber o que fazer com os dados.	Desafio da Engenharia de Atributos é entender o domínio.	Desafio [Engenharia de atributos]
Não é... as transformações que a gente tem que fazer nos dados não são triviais, a gente tem que aprender algumas vezes algumas transformações do próprio domínio, entender como o dado é processado e isso é um grande desafio, sempre.	Transformações dos dados podem ser oriundas do próprio domínio.	Limitação [Engenharia de atributos]
como eu falei - normalmente nos projetos onde eu trabalho - a avaliação é uma coisa definida... é um pouco mais difícil de definir, não é tão direto, enfim. [...] normalmente está ligado a uma métrica que é discutida profundamente com o cliente, sobre o que é que o modelo melhorou nos dados, se ele conseguiu chegar ou não. Normalmente isso é uma etapa de discussão, assim, acorda-se uma métricas e aí sim a gente avalia o modelo.	Métrica de avaliação definida em discussão com o cliente	Atividade [Requisitos do modelo]
	Métrica relacionada ao domínio da aplicação.	Limitação [Requisitos do modelo]
	presença do cliente	ator[avaliação do modelo]
isso é um ponto muito difícil nos projetos que a gente trabalha, porque normalmente isso não está claro, isso é justamente tema de pesquisa, é esse o ponto, a gente escreve paper sobre as avaliações que a gente está criando.	Métricas não triviais pra avaliação do modelo	Desafio [Avaliação do modelo]
	Métrica geralmente são relacionadas ao domínio da aplicação.	Limitação [Avaliação do modelo]
Assim, não é nem um pouco trivial garantir isso, porque justamente esses projetos que a gente trabalha envolvem discussões - vamos dizer assim - de tentar fazer uma avaliação que satisfaça o que o cliente acredita como razoável. Normalmente [essas avaliações] envolvem coisas do próprio domínio, algumas coisas que a gente tem que criar, mas não é... não tem garantia, é pesquisa mesmo... no nível mais básico dela.	Incerteza sobre se as métricas escolhidas para avaliação são as melhores.	Desafio [Avaliação do modelo]

Trecho	Código	Categoria
é… vamos lá né… Nesse workflow, a gente tá falando mais especificamente de ML, né? Então, no final das contas, é… esse workflow representa um subconjunto dos nossos workflows, a gente tem uma equipe ali desenvolvimento de software que não faz somente ML, então acho que dentro desse subconjunto.	Empresa desenvolve MLES.	
	Modelos de ML são módulos do MLES.	
Começando ali pelo primeiro, a parte de requisitos [Model Requirements], eu acho que vem muito do cliente, né? O que o cliente quer, qual a tarefa que ele gostaria que fosse atacada usando ML, qual é o problema que ele está enfrentando e que a gente definiu internamente conversando que ML é a melhor ferramenta para atacar aquele problema e etc, né? Acho que tem muito do cliente, do input né? Do... do… de qual é o problema, o que é que a gente quer melhorar e etc, né? E aí eu acho que com base nisso, já... já entra ali o pessoal né?	Requisitos do modelo como a definição das tarefas que devem ser resolvidas com ML.	
	Requisitos do modelo especificados pelo cliente.	Atividade [Requisitos do modelo]
	Presença do cliente	Ator [Requisitos do modelo]
	Presença do desenvolvedor	Ator [Requisitos do modelo]
A gente não tem ali papéis bem definidos, como uma software house teria, por ser um grupo que você sabe, né? Acho que os outros que você entrevistou também são do grupo ali, é um grupo de pesquisa aplicada, né? Tem muito desse componente de computação mas também tem um outro componente de pesquisa, não é uma software house por assim dizer, né? Então esses papéis se confundem um pouco.	Empresa de pesquisa e desenvolvimento	
	Desenvolvedores não possuem papeis bem específicos relacionados a ML	Limitação [Geral]
A gente tem... pessoas ali - que vamos dizer assim - tem esse interesse, essa área de atuação mais voltada para ML e qualquer uma dessas pessoas - respondendo a sua pergunta - atuam na perte de entender os requisitos e pensar quais modelos de ML são os mais adequados e etc, né? Então qualquer uma dessas pessoas, seja com um nível de mestrado ou doutorado podem vir a atacar esse problema e começar ali a estudar como propor um modelo de ML, para resolver o problema apresentado pelo cliente.	Desenvolvedores com interesse em ML trabalham na criação dos modelos.	Limitação [Geral]
Eu falei muito dos requisitos, né? Mas os outros são a mesma coisa, a gente tem esse grupo de pessoas, que tem essa - vou dizer aqui, interesse, área de atuação, que faz pesquisa - que sabe o que é ML, que entende como funciona essa tecnologia e tal. Sendo que elas atacam muitas vezes esse problema, muitas vezes pode ser um grupo de pessoas que sentam para discutir, às vezes uma única pessoas.	Único desenvolvedor participa de todo o workflow como alternativa	Limitação [Geral]
	Vários desenvolvedores participam de todo o worflow como alternativa.	Limitação [Geral]
A coleta de dados liga muito ao cliente, né? A gente volta para o cliente e pede, né? "Ah, a gente precisa de um dado assim, de um dado assado, com tal característica para testarmos tal hipótese e etc". Muitas vezes a rotulação dos dados ela é feita pelo cliente, acho que é uma característica também que vale a pena mencionar, porque é … se você fala de aplicações de ML mais gerais, que você tem, vamos lá, aqueles exemplos bem clássicos da imagem do gato, imagem do cachorro o carro e etc. Qualquer pessoa pode rotular, mas se você tem um negócio mais de domínio, que tem uma expertise específica, você vai precisar que o cliente ou especialista do domínio te ajude a rotular, nós mesmos não temos a capacidade de rotular.	Processamento de dados feito pelo cliente	Atividade [Processamento de dados]
	Fornecimento dos dados feito pelo cliente	Atividade [Processamento de dados]
	Cliente/Especialista dodomínio rotula os dados em domínios específicos.	Atividade [Processamento de dados]
	Presença do cliente	Ator [Processamento de dados]
	Presença do especialista de domínio	Ator [Engenharia de domínio]
A parte do model training e model evaluation é a mesma coisa que eu falei da parte de requisitos.	Único desenvolvedor participa de todo o workflow como alternativa	Limitação [Geral]
	Vários desenvolvedores participam de todo o worflow como alternativa.	Limitação [Geral]
E a feature engineering, também acho que vale a pena ressaltar, que hoje em dia a gente fala muito em deep learning e quando você fala de deep learning você pula um pouco essa etapa, né? Você tem ali um modelo que aprende as features, e também aprende a tarefa, seja ela uma tarefa de classificação, seja ela uma tarefa de é… segmentação, regressão e etc. A gente nem sempre passa por essa etapa de Feature Engineering, ela pode ser pulada.	Deep learning diminui a necessidade de engenharia de atributos	Limitação [Engenharia de atributos]
	Modelos de deep learning aprende as features e as tarefas	Limitação [Treino do modelo]
	Deep learning para engeharia de dados e treino do modelo de maneira única.	Método [Geral]
	Uso de deep learning para criação de modelos	Atividade [Treino do modelo]
como eu te falei, em geral, se eu pego um problema de ML para tratar, eu lido com todas essas etapas.	Único desenvolvedor em todas as etapas do workflow como alternativa.	Apontamento [Desenvolvimento]

A <<compahia>> é uma empresa grande, né? E a gente tem uma infraestrutura em nuvem, tem toda essa complexidade para você fazer o deployment de um modelo. A gente tem pessoas lá no grupo que… mais técnicas, que ajudam a gente, entende como é que essa infraestrutura funciona, como é que faz para colocar um modelo disponível em um serviço na nuvem e etc. Nessa etapa tem um pessoal mais especializado, mas até a parte de avaliação… desde o requisito até a parte de avaliação é a mesma pessoa, ou grupo de pessoas, caso seja um problema que tem mais de uma pessoa tratando dele.	Equipe especializada implanta modelo em núvem	Limitação [Implantação do modelo]
	Presença da equipe de infraestrutura	Ator [Implantação do modelo]
dessas etapas, eu acho que... vamos lá, deixa eu pensar um pouquinho aqui… Eu acho que a maioria das pessoas vai te falar que Data Cleaning é um dos principais problemas de ML e eu concordo, tá?	Processamento de dados como etapa mais problemática.	Desafio [Processamento de dados]
Porém como eu falei, os problemas que - eu pelo menos - tive a oportunidade de trabalhar na <<compahia>> a gente recebe muita coisa já pronta, ou trabalhada pelo cliente, então eu não vejo essa etapa - pelo menos do ponto de vista de ML - como uma das que mais tomam tempo.	Cliente executando o processamento de dados, diminui a complexidade da tarefa	Limitação [Processamento de dados]
Então eu acho que eu escolheria aí o Feature Engineering, eu passei bastante tempo pesquisando features e trabalhando com features novas, combinações de features e avaliando quais delas iriam produzir a melhor acurácia, então seria a caixinha aí de Feature Engineering.	Entrevistado considera ter mais experiência na Engenharia de atributos.	Apontamento [Desenvolvimento]
a gente… bem, depende de como você vê o Feature Engineering, né? Acho que um extremo seria você propor a sua própria feature, né? Propor uma forma de extrair uma característica do dado que vai trazer para você uma possibilidade de discriminação maior, né?	Atributos são aprendidos nos dados	Atividade [Engenharia de atributos]
	Atributos são coletados dos dados	Atividade [Engenharia de atributos]
A gente não chega nesse ponto, então a gente faz muito, é... pegar um conjunto de features… tem uma etapa de pesquisa, né? Quais são as features que a literatura aponta como boas para essa tarefa, dessas features que a literatura aponta a gente faz experimentos com elas individualmente avaliando a acurácia, e muitas vezes também combinando, né?	Busca na literatura de atributos conhecidos como relevantes para o problema.	Atividade [Engenharia de atributos]
Para ver se combinando a gente também produz uma acurácia maior. Uma outra possibilidade é a gente usar um seletor de features - por assim dizer - que aponta ali as features que tem a maior importância para aquela tarefa, então geralmente essa etapa caminha muito nessa direção de avaliar e combinar essas features e selecionar as melhores para aquela tarefa.	Uso de seletor de atributos para identificar atributos mais relevantes.	Atividade [Engenharia de atributos]
	Seletor de features para a escolha de atributos.	Método [Engenharia de atributos]
é… tem duas formas que eu consigo pensar rapidamente. A primeira - acho que é a mais fácil - é aquela que eu já falei que você tem algum tipo de ground truth que a gente chama de padrão ouro, ou verdades de campo e a gente utiliza essa informação como entrada para um seletor de features, e esse seletor de features ele já dá para você, com base nessa verdade de campo quais são as features que parecem ser mais importante para aquela determinada tarefa, por exemplo, um problema de classificação, e aí a gente caminha nessa direção.	Seletor de atributos aprimorados com verdades de campo	Método [Engenharia de atributos]
Então vê se eu enderecei direito a sua pergunta. Então esse cara, essa resposta desse seletor de features já é uma forma de avaliação, tá? Da importância daquela features.	Seletor de atributos aprimorado com verdades de campo como verificador dos melhores atributos para treinar o modelo.	Atividade [Engenharia de atributos]

Uma outra forma é, eu abro mão do seletor de features e faço na mão então eu vou selecionando ali subconjuntos de todas as features que eu quero avaliar, treino o modelo e na avaliação do modelo eu avalio se aquele conjunto de features produziu uma acurácia maior ou menor do que os outros conjuntos testados	Seleção manual de atributos	Atividade [Engenharia de atributos]
No final das contas é meio que uma força bruta, né? Você vai chegar em um ponto que você vai tem ali uma ideia de quais foram as features que produziram os melhores modelos.	Seleção manual de atributos acontece por força bruta	Limitação [Engenharia de atributos]
eu acho que… bom tem alguns, né? O primeiro acho que é essa pesquisa que tem que ser feita, se você for do domínio… se você já é do domínio… eu trabalho muito com análise de imagens, e já tenho uma boa noção que funcionam bem para determinadas tarefas, né? Mas se você tá pegando um tipo de dado novo, por exemplo, vou trabalhar com áudio, vou trabalhar com processamento de texto, com vídeo. Factualmente eu já não vou ter essa ideia, né? De quais são as melhores features, então vai ter uma etapa de pesquisa, geralmente indo a livros a papers e etc, para você fazer essa seleção, essa pré seleção de features. Então acho que esse é o primeiro desafio se você não é do domínio, se você é do domínio, você já tem uma boa ideia.	Pertencer ou não ao domínio do projeto é um primeiro desafio enfrentado pelo desenvolvedor.	Desafio [Engenharia de atributos]
	Necessidade de estudar novos temas para lidar com alguns projetos.	Desafio [Engenharia de atributos]
O outro desafio é fazer esses experimentos, muitas vezes - como eu falei - é força bruta mesmo, você ir testando várias possibilidades, é… você nunca sabe qual é a melhor feature para aquela tarefa, a não ser que você teste exatamente para aquela tarefa.	Identificar as melhores atributos para treinar o modelo como um desafio	Desafio [Engenharia de atributos]
Você pode ter feito um conjunto de features que foi muito bem em uma outra tarefa até similar a ela, mas uma pequena alteração na característica do dado ou na tarefa que você quer executar, pode é… representar algum tipo de mudança no conjunto de features que vai gerar o melhor resultado naquele momento, então acho que é essa…	O mesmo conjunto de atributos podem não funcionar para problemas similares	Limitação [Engenharia de atributos]
	Características dos dados influencias nos atributos.	Limitação [Engenharia de atributos]
Então o primeiro desafio acho que é você descobrir quais são as features, o outro é desse subconjunto que você descobriu que atendem aquele tipo de dado, qual o que de fato vai ter os melhores resultados para os seus dados.	Encontrar um subconjunto dentro dos atributos relevantes que realmente são os melhores para o problema.	Desafio [Engenharia de atributos]
Uma outra questão é que geralmente é um processo lento, em última instância você está reduzindo a dimensionalidade do seu dado, vamos trabalhar com imagens ou vídeo por exemplo, você tem um dado n-dimensional, duas dimensões, três dimensões, dependendo e você quer reduzir para um único descritorzinho que vai ser potencialmente discriminate para você separar as classes de interesse, né? Então esse processo de computação, em termos de performance computacional também é um processo caro, a gente tenta paralelizar e etc para reduzir um pouco esse esforço.	A transformação de atributos é um processo caro computacionalmente.	Desafio [Engenharia de atributos]
	Uso de transformação de atributos.	Método [Engenharia de atributos]
	Tranformação de atributos para reduzir os atributos	Atividade [Engenharia de atributos]

tá… então… acho que o processo é o mesmo que a gente discutiu, se você tá usando um seletor de features por exemplo, ele vai… você pode avaliar a relevância de uma feature de acordo com a importância que aquele seletor te dá, geralmente isso é apresentado em forma de uma nota, de um score, né? Você tem ali o maior score, aquela feature é mais importante, mais relevante, menor score, aquela feature é menos relevante, então usando algum desses métodos, você já tem diretamente aquela medida de relevância.	O seletor de atributos indica atributos relevantes e irrelevantes	Limitação [Engenharia de atributos]
No ponto em que você faz isso mais manualmente, como eu estava falando, fica muito a critério de como você faz o design, o planejamento do seu experimento, né? Mas você vai fazendo ali combinações de features e vai avaliando a acurácia, a qualidade do resultado usando aquelas features específicas e depois de alguns experimentos você vai ver que tem um conjunto de features que quando elas são selecionadas para serem utilizadas, a acurácia é menor, né? Então você tem ali um indicativo que aquelas features são menos relevantes, ou menos discriminantes para aquele problema específico.	Durante a seleção manual de atributos a relevância é indicada através de treino e avaliação.	Limitação [Engenharia de atributos]
eu acho que geralmente a gente vai fazendo uma coisa que meio, por demanda né? Você faz o primeiro teste, se você ver que o resultado ainda não é bom, você vai atacando os possíveis problemas.	O processo de melhoria dos atributos é iterativo.	Limitação [Engenharia de atributos]
Um deles é a normalização das features, né? As vezes você tem que fazer algum tipo de normalização, entre zero e um, ou aquela padronização do dado onde você subtrai a média e divide pelo desvio padrão. Dependendo da tarefa um tipo de normalização é melhor que outro, ou até mesmo tem normalizações que pioram o resultado, então geralmente é uma questão de tentativa e erro. A gente faz muito isso de acordo com o que os experimentos vão dizendo para a gente, agora existe também uma questão que eu acho importante nesse ponto que você levantou, que é...	Normalizações utilizadas para ajustar os atributos	Atividade [Engenharia de atributos]
	Uso de técnincas de normalização dos atributos	Método [Engenharia de atributos]
.. o conjunto de features muitas vezes ele é grande, não dá para a gente é… sair usando todas as featues que a gente tem disponível para aquele problema, então muitas vezes a gente usa um redutor de dimensionalidade, como a gente diz, com o PCA por exemplo, que ele já faz um pouco dessa limpeza dos dados. Ele vai deixar… ele aplica uma transformação nas features que projeta essas features em um espaço de dimensão menor do que o espaço original, onde essas featuers são as mais relevantes, ou melhor, explicam melhor a variância apresentada nos dados. Então com isso em teoria você já está tirando features que são irrelevantes, né? Que são correlacionadas.	Dados com alta quantidade de atributos são transformados através de redutores de dimensionalidade.	Atividade [Engenharia de atributos]
	Uso de PCA para transformação nos atributos.	Método [Engenharia de atributos]
	Redutores de dimensionalidade removem dependência nos dados.	Limitação [Engenharia de atributos]
essa é uma parte… eu sei que isso existe, na minha educação de Machine Learning eu trabalhei com isso, mas na prática eu não… eu particularmente não usei muito. Tipo, pegar uma feature e transform-lá.	Transformação de atributos mais complexas não são aplicadas	Limitação [Engenharia de atributos]
Agora uma coisa que a gente faz, claro, você trabalha com dados nulos, então você tem dados faltantes, você tem que aplicar algumas transformações para que você possa garantir que você tenha dados válidos. Uma das transformações que são aplicadas é… sei lá, usar o valor médio daquela feature ou a mediana e por aí vai. Tem algumas funções possíveis,	Transformações aplicadas em dados nulos ou faltantes	Atividade [Engenharia de atributos]

mas uma vez que você tem todos os valores válidos ali é… a gente não trabalha muito em cima das features, talvez porque eu trabalhe muito com imagem, então na verdade essa… essa brincadeira com transformações ela acaba acontecendo antes - muitas vezes - das features serem extraídas. A gente aplica na própria imagem, a gente aplica um realce na imagem, a gente aplica uma transformação na imagem que aumenta o contraste da imagem, entendeu? Então a gente tem algumas transformações no dado em si e não necessariamente nas featues que são extraídas daqueles dados.	Em imagens transformações são feitas nas próprias, não em atributos extraídos.	Atividade [Engenharia de atributos]
	Enriquecimento de dados.	Método [Processamento de dados]
essa pergunta ela é interessante, porque eu acho que é um profissional que está com os dias contados. No sentido de que eu falei anteriormente do deep learning por exemplo, né? Para muitas aplicações, não todas, por isso que eu falei que é uma resposta difícil de dar, mas a tendência é que esse profissional ele seja cada vez menos requisitados, porque nós temos algoritmos que conseguem não somente aprender a resolver a tarefa - como eu tava comentando - mas também conseguem aprender quais são as melhores features para resolver aquela tarefa, então você meio que abriu mão completamente desse profissional.	Especialistas de domínio são menos necessários em modelos de deep learning	Limitação [Engenharia de dados]
	Presença do Especialista do domíno	Ator [Engenharia de dados]
	Deep learning aprende as features e a resolução da tarefa.	Método [Engenharia de atributos]
Por outro lado, a gente sabe que deep learning e essas extração automática de features, ela não resolve o problemas do mundo, então ainda há espaço para esse profissional,	Deep learning não resolve todos os problemas, isso faz necessário um especialista de domínio.	Limitação [Engenharia de atributos]
mas acho que cada vez em nichos mais específicos, em problemas mais específicos, principalmente aqueles problemas de domínio que eu comecei falando, em que você ter uma expertise que não é facilmente aprendida, né? Por um profissional de TI, por um data scientist, por um machine learning engineer e etc, eu acho que esse profissional que entende os dados, né? Que entende as limitações do domínio, vamos supor por exemplo que é um domínio que também além dos dados em si claro, né? Você tem leis físicas que regem aquele domínio de dados ali e que essas leis físicas precisam ser respeitadas nessas features, né? Então seria acho que uma área que esse profissional seria requisitado ou até mesmo necessário.	Domínio específicos fazem necessário o especialista de domínio.	Limitação [Engenharia de atributos]
sim, a gente trabalha em geral com o domínio de recursos naturais no grupo, né? E a gente trabalha em geral com geólogos e geofísicos.	Geologos como especialistas de domínio	Ator [Engenharia de atributos]
	Geofísico como especialista de domínio	Ator [Engenharia de atributos]
Esses geólogos e geofísicos trazem o conhecimento da geologia da geofísica, e eles muitas vezes, eles nos alertam, "olha uma solução data driven, simplesmente baseada em dados nesse caso aí é um pouco problemático porque a gente tem toda uma física que explica esses dados que precisa ser respeitada".	Questões do domínio precisam ser respeitadas pelo modelo	Limitação [Treino do modelo]
	Soluções totalmente baseadas em dados podem não ser ideais	Limitação [Treino do modelo]
	Especialista nos dados indicam regras que precisam ser respeitadas	Atividade [Treino do modelo]
	Soluções data-driven nem sempre são possíveis.	
Até hoje tem um movimento aí que o nosso grupo também trabalha - talvez fosse até legal você entrevistar alguém que tá vendo isso, o <<cientista de dados>> é uma referência, mas a <<cientista de dados>> pode te dar o ponteiro para ele - que está trabalhando com modelos de ML physically informed, né? Então você usa a física, leis físicas para informar e reger por assim dizer o modelo de ML, para que ele sempre respeite essas leis físicas, então já tem uma linha de pesquisa voltada para isso.	Modelos Physically informed são regidos por regras físicas.	

é… como eu falei, acho que… pelo menos eu na minha experiência, tem isso né? A gente pode usar um modelo que já aprende as features, a gente pula completamente essa etapa. Tem a possibilidade de pegar as features e usar um seletor de features, como a gente chama, né? Que vai te dizer ali quais são as features mais relevantes para aquele problema e tem a força bruta como eu falei, que é você extrair as features, treinar o modelo, avaliar o modelo e baseado naquilo avaliação você determina se aquela feature foi boa ou não. Acho que é mais ou menos isso.	Incerteza sobre se os atributos escolhidos são os melhores para o modelo final	Desafio [Engenha de atributos]
essa é uma pergunta interessante, porque… talvez por eu não ser um engenheiro de software por assim dizer de profissão, minha posição é de pesquisador e o software acaba sendo uma ferramenta, a programação acaba sendo uma ferramenta para alcançar os resultados de pesquisa desejados, né?	Entrevistado não é da área de computação	Apontamento [Engenharia de atributos]
	Uso de ML como ferramenta para solução de problemas	
Então a gente acaba se baseando muito em bibliotecas disponíveis, né? Como scikit-learn, até na parte de imagem tem o scikit-image, o numpy, o scipy e etc, então assim eu não me vejo muito olhando para esses códigos diretamente, né? Eu programando esses códigos, então geralmente eu leio sobre o que o método faz, entendo o que o método faz ou lendo a própria documentação dessas bibliotecas, ou lendo os artigos que deram origem aos métodos e uma vez que eu entendi, eu utilizo se já estiver disponível. Acaba que programar essas coisas na mão fica cada vez menos comum para mim pelo menos.	Uso de frameworks para a criação de modelos.	Método [Geral]
	Incerteza sobre a corretude da implementação do framework.	Desafio [Treino do modelo]
então, na nossa experiência isso vai muito do cliente, como eu falei.	Primeiro responsável pelo processamento de dados é o cliente.	Limitação [Processamento de dados]
Os dados que a gente coleta estão intimamente relacionados ao problema que o cliente apresenta, não é? Baseado neste problema a gente verifica quais são os dados que melhor iriam ajudar a gente a ter uma primeira ideia de qual é o problema, como atacar o problema, as melhores features, modelos e etc.	A primeira análise dos dados é o que define como ele será abordado	Limitação [Processamento de dados]
Acho que aí é uma colaboração muito próxima com o cliente, que na verdade é o expert naquele domínio e conhece o dado melhor do que ninguém.	Cliente do domínio é quem melhor entende o dado.	Limitação [Processamento de dados]
Na parte de limpeza e labeling, como eu falei, ela acaba vindo muito do cliente, não é? A gente tem exemplos em que a gente recebeu dados e a gente rodou um modelo de ML, alguma coisa e percebeu que o resultado não estava interessante, voltou para o cliente e ele falou "ah, realmente gerei o dado de maneira errada ou esqueci de avisar tal coisa".	O cliente executa as etapas de coleta, limpeza e rotulação dos dados	Atividade [Processamento de dados]
	Cliente pode cometer erros no processamento de dados.	Limitação [Processamento de dados]
Na nossa vivência - no laboratório com os clientes que nós temos tido - isso acaba ficando por conta deles,	É característia dos projetos sempre receber os dados processados do cliente.	Limitação [Processamento de dados]
essa etapa, porém eu mencionei anteriormente que muitos que trabalha com ML veem essa etapa de limpeza, coleta de dados e rotulação como a mais trabalhosa e mais importante, porque nem sempre você tem o cliente ou o especialista no dado a disposição para você.	Processamento de dados como etapa problemática se você não tem auxílio do cliente.	Desafio [Processamento de dados]
Você precisa fazer isso manualmente, então… mas assim, que eu tente puxar aqui pela memória rapidamente, a gente não trabalhou muito nessa direção de limpeza, coleta e rotulação de dados, a gente diretamente.	Na empresa pouco se trabalhou sem o cliente no processamento de dados.	Limitação [Processamento de dados]

é, essa pergunta é interessante também, no sentido de que se você já é do domínio - é a segunda vez que eu volto a esse ponto - se você conhece qual é a característica esperada daquele dado, se você já trabalhou em algo parecido antes, tudo fica mais fácil.	Ter experiência no domínio é de grande utilidade no processamento de dados	Limitação [Processamento de dados]
Por exemplo, hoje a gente tem trabalhado com imagens sísmicas, ao olhar para uma imagem hoje eu já sei se ela tem a qualidade necessária para gerar um bom resultado, se eu vou conseguir tirar um bom resultado dela ou se não, se eu vou ter que voltar para o cliente e dizer, "olha não tem como processar esse dado? Não tem como entregar um dado com maior qualidade?".	Entrevistado diz ter experiência na área que atua.	
	Desenvolvedor consegue perceber a qualidade de imagens úteis para a área que ele conhece	Limitação [Processamento de dados]
	Melhorias de qualidade nos dados são solicitadas ao cliente.	Atividade [Processamento de dados]
Agora se você não é do domínio essa avaliação é muito difícil, você não sabe qual é a característica de um dado bom ou um dado ruim, então nesse caso você não tem muito o que fazer.	Desenvolvedores que não são do domínio dificilmente percebem a qualidade nos dados que tem.	Limitação [Processamento de dados]
Acho que um caminho possível é tentar rodar o modelo, rodar todo o workflow, essa pipeline aí de aprendizagem de máquina e ver se consegue um bom resultado, se você conseguir um bom resultado talvez isso indique que o dado é adequado. Se você não conseguir um bom resultado, acho que é o momento de voltar para quem te gerou o dado, né? Que geralmente no nosso caso é o cliente, e perguntar se não tem como... se ele não tem algum problema no dado ou se ele não consegue gerar um dado com melhor qualidade. Nesse caso aí é tentativa e erro, não tem jeito.	Desenvolvedores que não são do domínio tem que avaliar a qulidade do dados executando o workflow.	Atividade [Processamento de dados]
eu acho que é isso, né? Você criar essa… essa experiência com o dado a ponto de você começar a ter um sentimento do que é um dado com boa qualidade, o que é um dado com baixa qualidade, é… qual é o resultado esperado.	Um desafio do processamento de dados é adquirir essa capacidade de perceber a qualidade do dado de certo domínio.	Desafio [Processamento de dados]
Por exemplo, tem problemas de ML que são tão complicados que mesmo com dados de boa qualidade, você não consegue uma acurácia tão alta, porque o problema é complicado, então uma vez que você já tenha essa noção a priori, você já sabe o que esperar.	Alguns problemas de ML são complicados por natureza e nem dados de boa qualidade rendem bons resultados	Limitação [Geral]
Se você não tem, e é a primeira vez que você está trabalhando com aquele dado, se você tem uma acurácia baixa já acende uma alerta de que o dado pode estar com problema, ou tem um problema no seu modelo. Mas na verdade não é, é o melhor que se consegue porque o problema é complicado, né? Acho que a experiência da pessoa trabalhando naquele dado, naquele projeto, com aquele tipo de problema é fundamental para que você consiga enxergar essas coisas de maneira mais fácil.	Trabalhar com problemas complicados pode gerar a falsa impressão que algo de errado existe no workflow.	Limitação [Geral]
Caso você não tenha essa experiência, é tentativa e erro mesmo, não tem jeito. Ou claro, você pode acionar um colega, caso você tenha alguém disponível que tenha essa experiência, ou até mesmo, como eu já falei voltar ao cliente, tentar entender um pouco melhor, ou até mesmo para artigos e livros que possam ajudar nesse processo.	Se não há experiência no domínio o trabalho é de tentativa e erro	Limitação [Geral]
	Artigos e livros são úteis para aprender sobre um domínio.	