Pontifícia Universidade Católica
do Rio de Janeiro

Gabriel de Araujo Carvalho

Assessing the benefits of MLOps for Supervised Online Regression Machine Learning

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor : Prof. Markus Endler
Co-advisor: Prof. Marcos Kalinowski

Rio de Janeiro
October 2023

Pontifícia Universidade Católica
do Rio de Janeiro

Gabriel de Araujo Carvalho

Assessing the benefits of MLOps for Supervised Online Regression Machine Learning

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee:

Prof. Markus Endler
Advisor
Departamento de Informática – PUC-Rio

Prof. Marcos Kalinowski
Co-advisor
Departamento de Informática – PUC-Rio

Prof. Sérgio Colcher
PUC-Rio

Prof. Fabio Calefato
Uniba

Rio de Janeiro, October 2nd, 2023

Gabriel de Araujo Carvalho

Graduated in Information Systems at PUC-Rio in 2020. Worked as a software engineer in Brazilian and American companies and today work as a Tech Manager at Voltz.

Bibliographic data

Carvalho,Gabriel de Araujo

Assessing the benefits of MLOps for Supervised Online Regression Machine Learning / Gabriel de Araujo Carvalho; advisor: Markus Endler; co-advisor: Marcos Kalinowski. – 2023.

80 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2023.

Inclui bibliografia

1. Informática – Teses. 2. MLOps. 3. Aprendizado de Máquina. 4. Operações. 5. Focus Group. I. Endler, Markus. II. Kalinowski, Marcos. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

To my family and friends, for their support
and encouragement.

Acknowledgments

I would like to thank my advisors Markus Endler and Marcos Kalinowski, for the stimulus and partnership to carry out this work. Thanks also to Fabio Calefato and Sergio Colcher for accepting being part of my assessment board.

To my parents, Claudia and Geraldo, my sister Bruna, my aunt Tânia, my grandmother Nelly, and all the other members of my family, my eternal thanks for all the love, support, and enthusiasm over the years. To Bina, Eduardo, Márcia, and Venina, who showed me that the meaning of family doesn't need to be specifically blood-related and gave me love and care throughout my life.

I would also like to thank my friends and the people who have passed through my life during this master's degree, either at the very beginning or at the very end, who have kept me strong in my goal, each in their way. In particular, I would like to thank Rafael França and Giulia Duncan, who have been a source of support and comfort through this dissertation.

Thanks to Aladdin, my companion in days, nights, and dawns awake, in moments of relaxation, exercise, and an inexhaustible source of love and affection. During this master's degree, we went through several moments of fear and despair, whether during a pandemic, health problems, or the end of cycles. Having you by my side during all this gave life even more meaning. Forever, I am grateful for you.

To CNPq and PUC-Rio, for the aids granted, without which this work could not have been accomplished.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Carvalho,Gabriel de Araujo; Endler, Markus (Advisor); Kalinowski, Marcos (Co-Advisor). **Assessing the benefits of MLOps for Supervised Online Regression Machine Learning**. Rio de Janeiro, 2023. 80p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Context: Machine Learning Operations (MLOps) has emerged as a set of practices that combines development, testing, and operations to deploy and maintain machine learning applications. **Objective:** In this dissertation, we will assess the benefits and limitations of the use of MLOps principles in the context of online supervised models, which are widely used in applications such as weather forecasting, market trends, and risk identification. **Method:** We applied two research methods to assess the benefits of MLOps for supervised online machine learning applications: **(i)** developing a practical supervised machine learning project to deepen the understanding of the problem and of the MLOps principles usage possibilities; and **(ii)** two focus group discussions on the benefits and limitations of using the MLOps principles with six experienced machine learning developers. **Results:** The practical project implemented a supervised regression machine learning application using KNN. The application uses information on Rio de Janeiro's public bus line routes and calculates the bus trip duration based on the trip departure time of the day and trip direction. Due to the scope of the first version and given that it was not deployed into production, we didn't feel the need to use the MLOps principles we were expecting at first. Indeed, we identified the need for only one principle, the versioning principle, to align versions of the code and the data. The focus group revealed that machine learning developers believe that the benefits of using MLOps principles are many but that they do not apply to all the projects they worked on. The discussion brought up that most of the benefits are related to avoiding error-prone manual steps, enabling it to restore the application to a previous state, and having a robust continuous automated deployment pipeline. **Conclusions:** It is important to balance the trade-offs of investing time and effort in implementing the MLOps principles considering the scope and needs of the project. According to the experts, this investment tends to pay off for larger applications with continuous deployment that require well-prepared automated processes. On the other hand, for initial

versions of machine learning applications, the effort taken into implementing the principles might enlarge the scope of the project and increase the time needed to deploy a first version to production.

Keywords

MLOps; Machine Learning; Operations; Focus Group.

Resumo

Carvalho,Gabriel de Araujo; Endler, Markus; Kalinowski, Marcos. **Avaliação dos benefícios de MLOps para Aprendizado de Máquina Supervisionada Online de Regressão**. Rio de Janeiro, 2023. 80p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Contexto: As operações de aprendizagem automática (MLOps) surgiram como um conjunto de práticas que combina desenvolvimento, testes e operações para implementar e manter aplicações de aprendizagem automática. **Objetivo:** Nesta dissertação, iremos avaliar os benefícios e limitações da utilização dos princípios de MLOps no contexto de modelos supervisionados online, que são amplamente utilizados em aplicações como a previsão meteorológica, tendências de mercado e identificação de riscos. **Método:** Aplicámos dois métodos de investigação para avaliar os benefícios dos MLOps para aplicações de aprendizagem automática online supervisionada: **(i)** desenvolvimento de um projeto prático de aprendizagem automática supervisionada para aprofundar a compreensão do problema e das possibilidades de utilização dos princípios MLOps; e **(ii)** duas discussões de grupo de foco sobre os benefícios e limitações da utilização dos princípios MLOps com seis programadores de aprendizagem automática experientes. **Resultados:** O projeto prático implementou uma aplicação de aprendizagem automática de regressão supervisionada utilizando KNN. A aplicação utiliza informações sobre as rotas das linhas de autocarros públicos do Rio de Janeiro e calcula a duração da viagem de autocarro com base na hora de partida do dia e no sentido da viagem. Devido ao âmbito da primeira versão e ao facto de não ter sido implementada em produção, não sentimos a necessidade de utilizar os princípios MLOps que esperávamos inicialmente. De facto, identificámos a necessidade de apenas um princípio, o princípio do controlo de versões, para alinhar as versões do código e dos dados. O grupo de discussão revelou que os programadores de aprendizagem automática acreditam que os benefícios da utilização dos princípios MLOps são muitos, mas que não se aplicam a todos os projectos em que trabalham. A discussão revelou que a maioria dos benefícios está relacionada com a prevenção de passos manuais propensos a erros, permitindo restaurar a aplicação para um estado anterior e ter um pipeline robusto de implementação automatizada contínua. **Conclusões:** É importante equilibrar as compensações do investimento de tempo e esforço na implementação dos princípios de MLOps, considerando o âmbito e as necessidades do projeto. De acordo com os especialistas, esse investimento tende a compensar para aplicativos maiores com implantação contínua que exigem processos automatizados bem preparados. Por outro lado, para versões iniciais de aplicações de aprendizagem automática, o esforço despendido na implementação dos princípios pode alargar o âmbito do projeto e aumentar o tempo de execução.

Palavras-chave

MLOps; Aprendizado de Máquina; Operações; Focus Group.

Table of contents

List of figures

List of tables

List of algorithms

List of codes

List of Abreviations

ML – Machine Learning

SML – Supervised Machine Learning

OSML – Online Supervised Machine Learning

MLOps – Machine Learning Operations

DevOps – Development and Operations

KNN – K-Nearest-Neighbour

SVM – Support Vector Machines

The best way to predict the future
is to invent it.

Alan Kay, .

1
Introduction

1.1
Context and Motivation

Machine Learning (ML) is a specific discipline of Artificial Intelligence (AI) that provides machines the ability to automatically learn from data and past processing experiences to identify data patterns, classify data, and predict results with minimal human intervention. ML has allowed businesses to be more innovative (AYKOL; HERRING; ANAPOLSKY, 2020), efficient (Pandey; Rautaray, 2021), and sustainable (HERAS; LUQUE-SENDRA; ZAMORA-POLO, 2020). It has enabled powerful web searches, self-driving vehicles, practical language processing and generation, and a better knowledge of our genetic underpinning.

Supervised ML (SML) is a sub-branch of ML (SHETTY et al., 2022) and generally depends on a human domain expert who "teaches" the learning scheme with required supervision. It also maps inputs to selected outputs. The SML is trained using a *training data set*, that is subject to annotation/labeling, while the non-annotated data is termed the *testing set*. SML problems are grouped into classification and regression. In classification problems, the prediction results correspond to discrete values. In regression, on the other hand, the results correspond to continuous values. In (RAY, 2019), we find a review of multiple machine learning algorithms.

Online machine learning is a method of machine learning in which data becomes available in sequential order and is used to update the best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once. The goal of online learning is to make a sequence of accurate predictions given knowledge of the correct answer to previous prediction tasks and possibly additional available information (SHALEV-SHWARTZ, 2012). Online learning is a common technique used in areas of machine learning where it is computationally infeasible to train over the entire dataset. It is also used in situations where it is necessary for the algorithm to dynamically adapt to new patterns in the data, or when the data itself is generated as a function of time, for example, stock price prediction.

Unfortunately, the success of many productive ML applications in real-world settings falls short of expectations (KOCIELNIK; AMERSHI; BEN-

NETT, 2019). A large number of ML projects fail—with many ML proofs of concept never progressing as far as production (MEULEN; MCCALL, 2018).

From a research perspective, this does not come as a surprise as the ML community has focused extensively on the building of ML models but not on (a) building production-ready ML products and (b) providing the necessary coordination of the resulting, often complex ML system components and infrastructure, including the roles required to automate and operate an ML system in a real-world setting (POSOLDOVA, 2020).

For instance, in many industrial applications, data scientists still manage ML workflows manually to a great extent, resulting in many issues during the operations of the respective ML solution (LWAKATARE et al., 2020). This is a particular problem, especially in online learning, in which data arrives in a sequential order, and the model is expected to learn and update the best predictor for future data at every step.

To help solve problems like building production-ready applications for complex systems, the technical community has started to adopt continuous software engineering practices such as Development and Operations (DevOps). DevOps can be defined as the development method with an emphasis on software delivery, automated deployment, continuous integration, and quality assurance (JABBARI et al., 2016).

The practice of continuous delivery of ML solutions is called Machine Learning Operations (MLOps), which mimics DevOps practices but introduces additional actions that are specific to ML (MÄKINEN et al., 2021). MLOps is a paradigm, including aspects like best practices, sets of concepts, as well as a development culture when it comes to the end-to-end conceptualization, implementation, monitoring, deployment, and scalability of machine learning products (KREUZBERGER; KüHL; HIRSCHL, 2023). Most of all, MLOps represents the alignment of the construction of machine learning models with software development and operation (KALINOWSKI et al., 2023).

MLOps aims to bridge the gap between development and operations for ML-enabled systems. Essentially, it seeks to facilitate the creation of machine learning products by leveraging the following principles: versioning, testing, automation, reproducibility, and monitoring (VISENGERIYEVA et al., 2023).

The academic space has focused intensively on machine learning model building and bench-marking but too little on operating complex machine learning systems in real-world scenarios. In the real world, we observe data scientists still managing ML workflows manually to a great extent (KREUZBERGER; KüHL; HIRSCHL, 2023). Furthermore, it can be observed that the use of software engineering best practices and MLOps is still relatively unexplored (KREUZBERGER; KüHL; HIRSCHL, 2023).

1.2 Objectives

The goal of this dissertation is to assess the benefits and limitations of the use of MLOps principles in the context of online supervised learning. Our research method to address this goal is twofold. First, we developed a practical supervised machine learning project to deepen the understanding of the problem and of the MLOps principles usage possibilities. Second, we conducted two focus group sessions among experienced machine learning developers to understand the benefits and limitations of using the MLOps principles.

1.3 Dissertation Outline

The remainder of this dissertation is organized as follows. In Chapter 2, we present the background on machine learning and MLOps. In Chapter 3, we describe the practical project and the results of the reflection on MLOps principles usage possibilities. In Chapter 4, we provide the details on how the focus group sessions were conducted and the outcomes on benefits and limitations of using MLOps principles. Finally, Chapter 5 contains our concluding remarks.

2
Machine Learning and MLOps

2.1
Introduction

In this chapter, we briefly describe the background on machine learning (Section 2.2), supervised machine learning (Section 2.3), and MLOps (Section 2.4), which are the building blocks for understanding this dissertation.

2.2
Machine Learning

Machine learning (ML) is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment. Techniques based on machine learning have been applied successfully in diverse fields ranging from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, and computational biology to biomedical and medical applications (NAQA; MURPHY, 2015).

ML has allowed businesses to be more innovative (AYKOL; HERRING; ANAPOLSKY, 2020), efficient (Pandey; Rautaray, 2021), and sustainable (HERAS; LUQUE-SENDRA; ZAMORA-POLO, 2020). Real-world problems have high complexity, making them excellent candidates for the application of ML. Machine learning can be applied to various areas of computing to design and program explicit algorithms with high-performance output, for example, email spam filtering, online stock trading, and others(ALZUBI; NAYYAR; KUMAR, 2018). The self-driving Google cars, Netflix showcasing the movies and shows a person might like, online recommendation engines like friend suggestions on Facebook, and credit card fraud detection are all real-world examples of the application of machine learning.

Compared to traditional software-engineered system development workflow, machine learning applications development workflow arises two main differences. **(i)** For an ML-enabled system, the final output of the development workflow is the model's outcome rather than the code itself (EPPERSON et al., 2022). And **(ii)** in contrast to common problem-solving in software applications, machine learning applications workflows require a continuous feedback loop due to the frequent iteration involving model selection, hyper-parameters tuning, and dataset refinement (AMERSHI et al., 2019).

Based on the nine ML life cycle stages presented by (AMERSHI et al., 2019), Calefato *et al.* (2023) adapted Figure 2.1 grouping these nine stages into three groups with respective roles: **(i)** Preparation - Data engineer; **(ii)** Analysis: Data scientist, **(iii)** Dissemination - ML engineer/MLOps. Following, we present a brief explanation of the nine steps presented in the image, grouped by their activities and roles, based on the definitions of the article:

- **Preparation - Data engineer**
  - **Data collection** - The first step of preparation is where data engineers retrieve data from multiple data sources and integrate them into a dataset.
  - **Data cleaning** - The cleaning consists of removing inaccurate, duplicated, or irrelevant data.
  - **Data labeling** - The last step involves assigning ground truth labels to each record in the dataset. This step can be done both manually or by ML-assisted.
- **Analysis - Data scientist**
  - **Feature engineering** - The first step of analysis consists of all the activities related to extracting and selecting from the dataset the most informative features for building ML models.
  - **Model training** - Model training refers to the step of training the chosen method using the selected features.
  - **Model evolution** - The last step evaluates the model trained in the previous step by using a predefined performance metric. Some examples are log loss, accuracy, and AUC. This step typically involves extensive human evaluation to ensure the model output meets the required performance criteria.
- **Dissemination - MLOps**
  - **Model deployment** - This is the step where the evaluated machine learning model is integrated into a target production environment to make practical predictions or business decisions based on new data.
  - **Model Monitoring** - The last step consists of monitoring the deployed models for possible errors or decreases in performance (e.g., model drift).

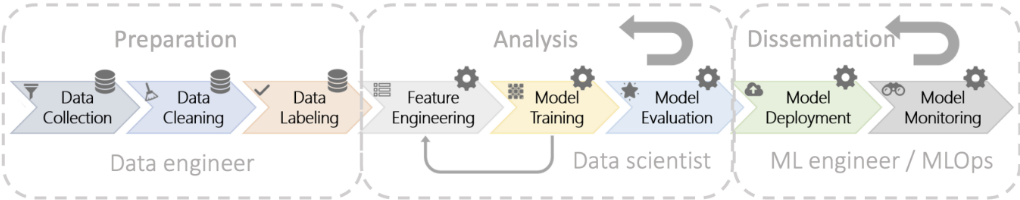

Figure 2.1: ML workflow with activities and roles (adapted by (CALEFATO et al., 2023) from (AMERSHI et al., 2019))

2.3 Supervised Learning

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning. According to IBM, Supervised Learning is defined by its use of labeled datasets to train algorithms that classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately, which occurs as part of the cross-validation process (IBM, 2023a).

A systematic literature review of supervised learning (DRIDI, 2021) describes supervised learning entails learning a mapping and labeling between a set of input variables X and an output variable Y and applying this mapping to predict the outputs for unlabeled data. Its fundamental architecture begins with the dataset collection; the dataset is then partitioned into testing and training data, and then the data is preprocessed. Extracted features are fed into an algorithm, and the model is then trained to learn the features associated with each label. Finally, the model is supplied with the test data, and said model makes predictions on the test data by providing the expected labels, as illustrated in the figure:

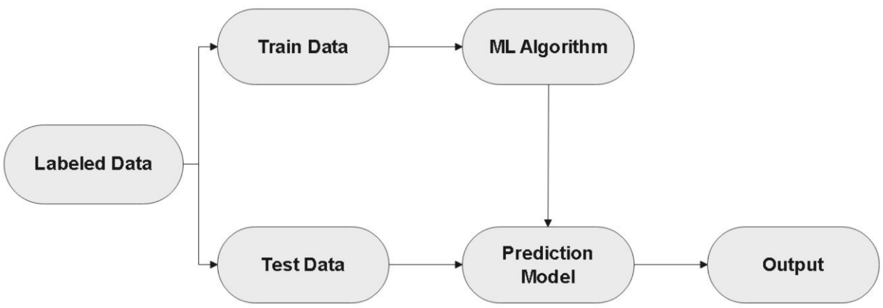

Figure 2.2: Supervised Learning Architecture

Supervised learning algorithms consist of performing supervised learning from historical data. Supervised algorithms work by taking a given set of n patterns, where each is composed of information from independent explanatory variables and a continuous or discrete response variable (dependent), and aims to build a model that, given a new pattern, estimates the expected value for the response variable (KALINOWSKI et al., 2023).

Examples of commonly used supervised algorithms include (ESCOVEDO; KOSHIYAMA, 2020):

1. Linear Regression.

2. Decision Tree.

3. K-Nearest-Neighborhood (KNN).

4. Support Vector Machines (SVM).

While descriptions of these algorithms can be found elsewhere (SHETTY et al., 2022; ESCOVEDO; KOSHIYAMA, 2020), hereafter, we provide a brief overview of KNN, given that our practical example described later on uses this algorithm.

2.3.1 KNN

KNN stands for K Nearest Neighbor classifications, identifying new records by a combination of K's most recent historical records. This method can be used for Classification or Regression problems. For classification problems, a class label is assigned on the basis of a majority vote—i.e., the label most frequently represented around a given data point is used as the chosen label for that specific data point. Regression problems use a similar concept as a classification problem, but in this case, the average of the k nearest neighbors is taken to make a prediction about a classification. The main distinction here is that classification is used for discrete values, whereas regression is used with continuous ones (IBM, 2023b).

To determine the distance between the neighbors of a query point before assigning a class label to it, we need to determine the distance metrics; examples are (IBM, 2023b):

1. Euclidean distance: This is the most commonly used distance measure and is limited to real-valued vectors. It measures the distance as a straight line between the target point and the point being measured.

2. Minkowski distance: This distance metric is also to be used with real-valued vectors. It considers the largest difference between all vector dimensions as the distance.

3. Manhattan distance: This is another popular distance metric measuring the absolute value between two points. It is also referred to as taxicab distance or city block distance as it is commonly visualized with a grid, illustrating how one might navigate from one address to another via city streets.

4. Hamming distance: This technique is typically used with Boolean or string vectors, identifying the points where the vectors do not match. As a result, it has also been referred to as the overlap metric.

(RAY, 2019) presents a short review of multiple machine learning algorithms and describes KNN advantages as the following:

– It is a simple technique to be implemented.

– Building the model is *cheap*.

– It is an flexible classification scheme and suited for Multi-modal classes.

– It can sometimes be the best method. KNN outperformed Support Vector Regression for protein function prediction using expression profiles.

On the other side, it also lists the following disadvantages:

– Classifying unknown records is relatively expensive.

– It requires distance computation of k-nearest neighbors.

– With the growth in training set size, the algorithm gets computationally intensive, and noisy/irrelevant features will result in degradation of accuracy.

2.4 MLOps

Machine Learning Operations (MLOps) is a core function of machine learning engineering, focused on streamlining the process of deploying machine learning models to production and then maintaining, scaling, and monitoring them. MLOps is a collaborative function, often comprising data scientists, DevOps engineers, and IT. An optimal MLOps experience is one where machine learning assets are treated consistently with all other software assets within a CI/CD environment (VISENGERIYEVA et al., 2023; KALINOWSKI et al., 2023). *I.e.*, machine learning models can be deployed alongside the services that wrap them and the services that consume them as part of a unified release process(VISENGERIYEVA et al., 2023).

As machine learning and AI are increasingly pervasive in software products and services, we need to establish best practices and tools to test, deploy, manage, and monitor ML models in real-world production. In short, with MLOps, we strive to avoid "technical debt" in machine learning applications. With that, (VISENGERIYEVA et al., 2023) describes a summary of MLOps principles and best practices that we will be using in the thesis. The following list summarizes the MLOps principles for building ML-based software:

– Automation.

– Monitoring.

– Versioning.

– Reproducibility.

– Test.

– Deployment.

To reduce the scope of this research, we will focus on assessing three of these principles more directly: Versioning, Automation, and Monitoring. These principles were chosen because they are closely related to the final stages of the process, where the product is delivered to the final application. Indeed, we indirectly also address the other principles, given that Reproducibility, Testing, and Deployment are strongly related to the Automation principle.

We present next a brief background explanation of the six principles.

2.4.1 Automation

The objective of an MLOps team is to automate the deployment of ML models into the core software system or as a service component. This means automating the complete ML workflow steps without any manual intervention. Triggers for automated model training and deployment can be calendar events, messaging, monitoring events, as well as changes in data, model training code, and application code.

Automating the ML workflow can be divided into stages and areas such as tests, deployment, and monitoring. This is because the ML pipeline process is responsible for many stages, and each has its responsibility and output. The following image by (VISENGERIYEVA et al., 2023) shows the stages that reflect the process of ML pipeline automation:

MLOps Stage	Output of the Stage Execution
Development & Experimentation (ML algorithms, new ML models)	Source code for pipelines: Data extraction, validation, preparation, model training, model evaluation, model testing
Pipeline Continuous Integration (Build source code and run tests)	Pipeline components to be deployed: packages and executables.
Pipeline Continuous Delivery (Deploy pipelines to the target environment)	Deployed pipeline with new implementation of the model.
Automated Triggering (Pipeline is automatically executed in production. Schedule or trigger are used)	Trained model that is stored in the model registry.
Model Continuous Delivery (Model serving for prediction)	Deployed model prediction service (e.g. model exposed as REST API)
Monitoring (Collecting data about the model performance on live data)	Trigger to execute the pipeline or to start a new experiment cycle.

Figure 2.3: MLOps pipeline automation stages (VISENGERIYEVA et al., 2023).

Analyzing the MLOps stages, we might notice that the MLOps setup requires several components to be prepared:

- Source control: versioning the code, data, and model - Every deployment knows exactly which code, data, and model it is using. With that, it creates control over the combination of code, data, and model.
- Test and Build Service: Using CI for quality assurance and to build executables for pipelines - This guarantees that, for every deployment, the CI will automatically ensure that the test and build services will run and automatically block any deployment that fails the created tests and build.

– Deployment Services: Using the CD tools to deploy pipeline to a target environment - This ensures that the developer knows exactly which environment the code is being deployed to. For example, if the application has a staging and production environment, you can use the CD to control the deployment into the correct environment.

– Model Registry: A registry for storing already trained ML models - Guaranteeing the storage of a previously trained ML model creates the possibility of rolling back faster to a previously trained working model.

– Feature Store: Preprocessing input data as features to be consumed by the model - The model will be consuming already preprocessed input that will facilitate its training.

– ML Pipeline Orchestrator: Automating the steps of the ML experiments - Automating the steps reduces human errors that might happen when orchestrating a manual deployment.

To summarize, implementing ML in a production environment means deploying an ML pipeline that can automate the retraining and deployment of new models. Setting up a CI/CD system enables the automatic testing and deployment of new pipeline implementations.

The decision to adopt MLOps doesn't mean one should immediately move all processes to an automated level. The process of implementing MLOps and, principally, its automation might take some time. To understand the maturity level of the application MLOps automation (GOOGLE, 2023) set three levels of automation:

1. Manual process: This is a typical data science process performed at the beginning of implementing ML. This level has an experimental and iterative nature. Every step in each pipeline, such as data preparation and validation, model training, and testing, is executed manually.

2. ML pipeline automation: The next level includes the execution of model training automatically. Whenever new data is available, the process of model retraining is triggered. This level of automation also includes data and model validation steps.

3. CI/CD pipeline automation: In the final stage, a CI/CD system is used to perform fast and reliable ML model deployments in production. The core difference from the previous step is that we now automatically build, test, and deploy the Data, ML Model, and ML training pipeline components.

2.4.2 Monitoring

As part of the dissemination step, after the ML model has been deployed, the model monitoring step aims to monitor it to ensure that the ML model performs as expected. Monitoring an ML application is important for being able to understand problems with the data, model, and application. (BRECK et al., 2017b) creates a test score by presenting 28 specific tests and monitoring needs for model monitoring activities in production. Following, we list examples of monitoring:

– Monitor data invariant in training and serving inputs: Alert if data does not match the schema, which has been specified in the training step.
– Monitor the numerical stability of the ML model.
– Monitor computational performance of an ML system: Both dramatic and slow-leak regression in computational performance should be notified.
– Monitor degradation of the predictive quality of the ML model on served data.

Monitoring could help detect issues within a model before they become inconsistent and even be retrained if needed. For ML applications, it can be useful to help improve the data training set, model stability and quality, and the predictive quality of the application on serving data.

2.4.3 Versioning

Besides the common code versioning, the MLOps versioning principle consists of organizing and versioning the datasets and models by their sets. In a machine learning project, data scientists continuously work on developing new models. This process relies on trying different combinations of data, parameters, and algorithms. During this process, as described in section 2.2, new versions of a dataset can be created during the preparation and analysis phases of a machine learning project. The concept of data versioning control describes the practice of storing, tracking, and managing the changes in a dataset.

Similarly to how data versioning indicates the version control for data in machine learning applications, model version control or model versioning indicates the version control of ML models. It describes the practice of storing, tracking, and managing the changes in a machine learning model. Examples of reasons why an ML model might change are: models can be retrained based on new training data, new training approaches, or the necessity to quickly roll back to a previous serving version. Hence, it's possible to need access to all versions of the productionized ML model beyond other reasons (VISENGERIYEVA et al., 2023). Thus, creating an environment where it's possible to go back and forth on older or new experiments is positive.

Therefore, the MLOps principles reinforce the necessity of versioning code, data, and models. Nowadays, few data and model versioning control tools have been developed to support the process of versioning models and datasets; examples are MLFlow (PROJECTS, 2023), DVC(ITERATIVE.AI, 2023), Sacred(GREFF et al., 2017).

2.4.4 Reproducibility

The reproducibility principle is described as the process of repeatedly running a machine learning application on certain datasets and obtaining the same or similar results. The reproducibility principles ensure that researchers can reproduce the accuracy of reported results and detect biases in the models. Analyses and models that are reproducible by third parties can be examined in depth and, ultimately, become worthy of trust. To that end, we believe the life sciences community should adopt norms and standards that underlie reproducible machine learning research (STODDEN; BORWEIN; BAILEY, 2013).

Reproducibility might help developers reduce errors and ambiguity when the projects move from development to production by ensuring data consistency. Following, we present examples of challenges and what the reproducibility principle addresses to help reduce these challenges during machine learning phases (VISENGERIYEVA et al., 2023):

Data collecting - **Challange**: reproducing the generation of the training data. **Reproducibility principle**: the principle introduces the practice of implementing data versioning, snapshots of the dataset, and adding timestamps to data sources so that a view of the data can be retrieved at any point.

Feature engineering - **Challange**: during the extracting and selecting from step from the dataset, **(i)** missing value can be imputed with random or mean values, **(ii)** labels based on the percentage of observation might be removed. **Reproducibility principle**: practices such as adding version control under the feature generation and being able to reproduce the data collecting as solutions that might avoid the challenge.

Model deployment - **Challange**: the input data required by the ML model is missing in the production environment. **Reproducibility principle**: through an automated deployment pipeline (Section 2.4.1), the reproducibility principle consists that the software versions and dependencies should match the production environment.

2.4.5 Testing

Testing and monitoring are important strategies for improving reliability, reducing technical debt, and lowering long-term maintenance costs; however, as suggested by Figure 2.4, ML system testing is also a more complex challenge than manually coded systems because ML system behavior depends strongly on data and models that cannot be strongly specified (BRECK et al., 2017a).

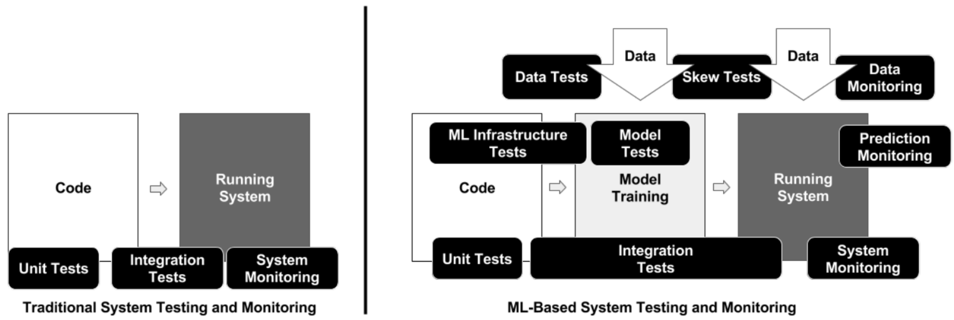

Figure 2.4: Unlike a manually coded system (left), ML-based system behavior is not easily specified in advance. This behavior depends on the dynamic qualities of the data and on various model configuration choices (BRECK et al., 2017a)

The MLOps testing principle introduces tests for features and data, model development, and ML infrastructure (VISENGERIYEVA et al., 2023) as part of ensuring data and model quality. A few examples of how to produce these tests are:

- **Features and data test**
  - Feature creation should be tested by unit tests.
  - Measure data dependencies, inference latency, and RAM usage for each new feature. Compare it with the predictive power of the newly added features.
- **Tests for reliable model deployment**

    - Validate the performance of a model by using an additional test set, which is disjoint from the training and validation sets.
    - Test the model fairness and bias by collecting more data that includes potentially under-represented categories and examine input features to see if they correlate with protected user categories.

- **ML infrastructure test**

    - Crash tests for model training. The ML model should be restored from a checkpoint after a mid-training crash.
    - The full ML pipeline should be integration tested by creating a fully automated test that validates that the data and code successfully finish each stage of training and the resulting ML model performs as expected.
    - Setting a threshold and testing for slow degradation in model quality over many versions on a validation set.

2.4.6 Deployment

The deployment of machine learning models or pipelines is the process of making models available in a production environment so applications and APIs can consume the trained model, provide new data points, and get predictions. In short, deployment in machine learning is the method by which you integrate a machine learning model from a development environment into an existing production environment to make practical business decisions based on data (model deployment).

The MLOps principle deployment consists of containerizing the ML stack and providing access to the production deployed application to be accessed by software applications.

3
Practical Project: A Study of Rio de Janeiro's Public Bus Lines Routes

3.1
Introduction

In this chapter, we describe the practical supervised machine learning project implemented to deepen the understanding of the problem. The application uses information on Rio de Janeiro's public bus line routes and calculates the bus trip duration based on the trip departure time of the day and trip direction. We reflect on MLOps principles usage possibilities within this scenario.

3.2
Practical Project Description

Seeking to analyze and understand how MLOps principles affect machine learning applications, we created a supervised online machine learning application. Our goal is to predict the bus trip duration based on the trip departure time of the day and trip direction. The project will give us the opportunity to develop a real project with a defined scope, minimal viable product, and goal in one month. The time decided for this implementation will match the time we have available for conducting this dissertation, and it will also simulate how a real-life project with a deadline would be. The main goal is to understand how we can fit the MLOps principles and where we will need to use them.

The theme chosen for the project was the study of bus line routes in Rio de Janeiro - Brazil. This project uses data collected through smart sensors scattered around the city, which, when in contact with sensors installed in the buses, register the date and time that the specific public vehicle passed in a given location.

This data has been made publicly available by the Rio de Janeiro City Hall and contains information about the trajectory of several bus lines during 2018. For study purposes in this project, we will focus on using data from the "415" bus line that has 2 routes and runs every day between 04:00 am and 11:00 pm, with an average of 65 stops. For clarity of understanding, we will use the concept of **trip** when the bus travels a route, in a complete way or not, going from a starting point A to the endpoint B, previously foreseen in its itinerary.

The project tends to use the information collected by these intelligent sensors scattered in the city with the intent of applying machine learning algorithms in order to detect various aspects and anomalies during a route, among them:

- Beginning and end point of a trip.
- Total duration of a trip.
- Average speed during a trip.
- If the trip was completed or not.
- Whether the bus was returning to or leaving starting point A.
- Whether the bus is making a circuit.
- Whether the bus went out of its route to "cut across".
- Whether the bus, during its trip, wandered on an unexpected route.

Following, we demonstrate the results obtained during the creation of this project and the needs and effects of using MLOps principles in the creation of this practical project.

3.2.1 Preparation

3.2.1.1 Data Collection

The original data set is available in the format of a group of files containing a structure compatible with the JSON format. It contains 5766639 entries for 5683 unique buses of 436 different lines. Each entry on the data set is a sample of a bus state at a given moment containing the following information:

1. timestamp: date and time of when the bus state was sampled.
2. id: bus serial number.
3. line: bus line.
4. latitude: latitude coordinate component of the bus position.
5. longitude: longitude coordinate component of the bus position.
6. speed: bus speed.

3.2.1.2 Exploratory Analysis

As mentioned in section 3.2.1, the bus line "415" was picked as a case study. The reason behind this decision is based on domain knowledge and information available for that bus line. Other bus lines on the data set fit the same criteria; the decision for the "415" specifically was arbitrary.

Our first analysis was plotting the coordinates of each sample of the line "415" to check if they formed the expected pattern. As shown in Figure 3.1, most of the data forms routes. However, we also have two other types of outliers. The first type is coordinate points far away from the clusters. The second are clusters with a reasonable amount of points but with a density far below the other clusters.

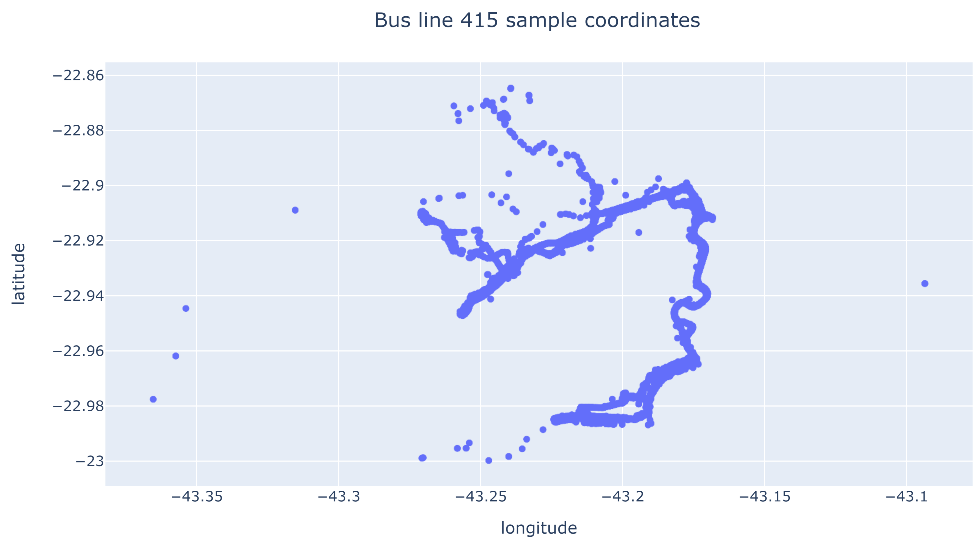

Figure 3.1: Bus line 415 sample coordinate analysis.

To understand the nature of the wandering points, we referred to the bus company definitions of the bus line's route (MOOVIT, 2023). Based on Figure 3.2, it is possible to identify that the data set has two well-defined wandering routes on the northern part of the route.

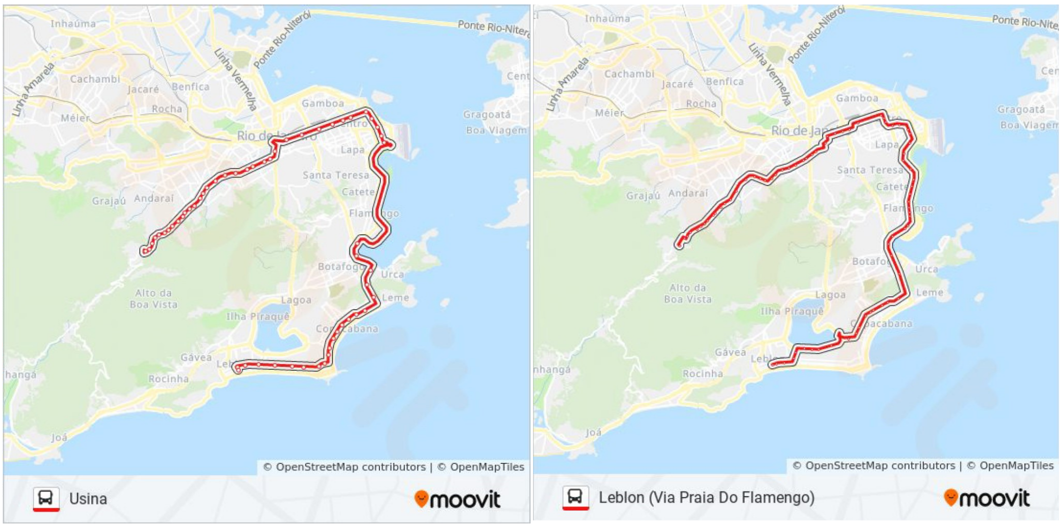

Figure 3.2: Bus line 415 official route definitions.

To better understand our data set, we added more dimensions to the scatter plot. In Figures 3.3, 3.4, and 3.5, we have plotted each point colored by the bus id, hour of the day, and speed respectively.

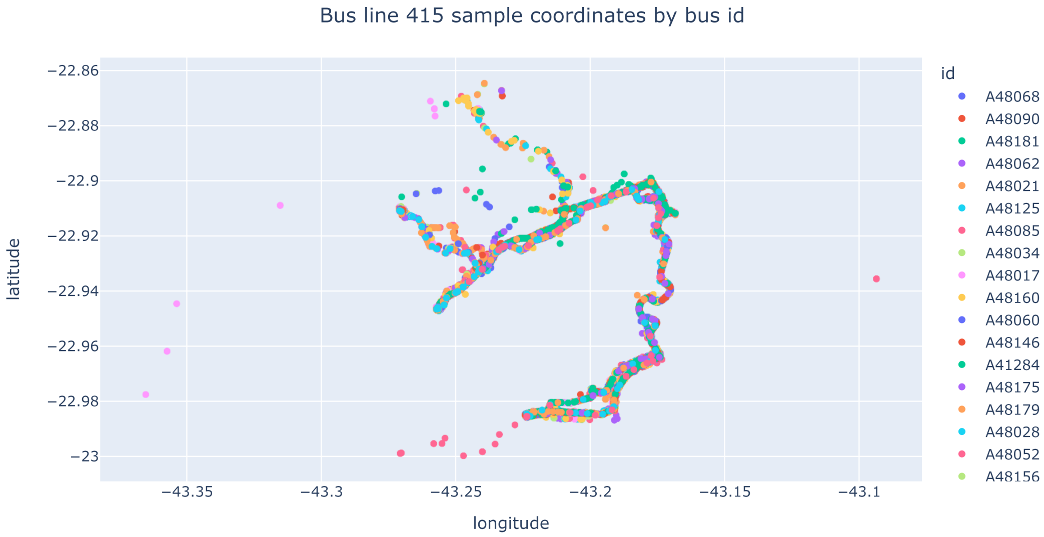

Figure 3.3: Bus line 415 sample points by bus id.

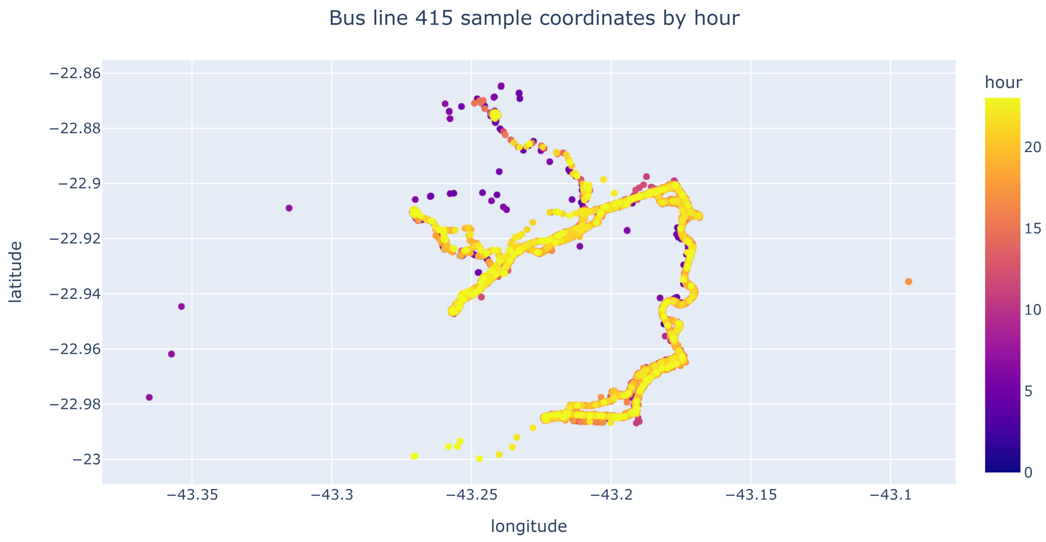

Figure 3.4: Bus line 415 sample points by bus hour of the day.

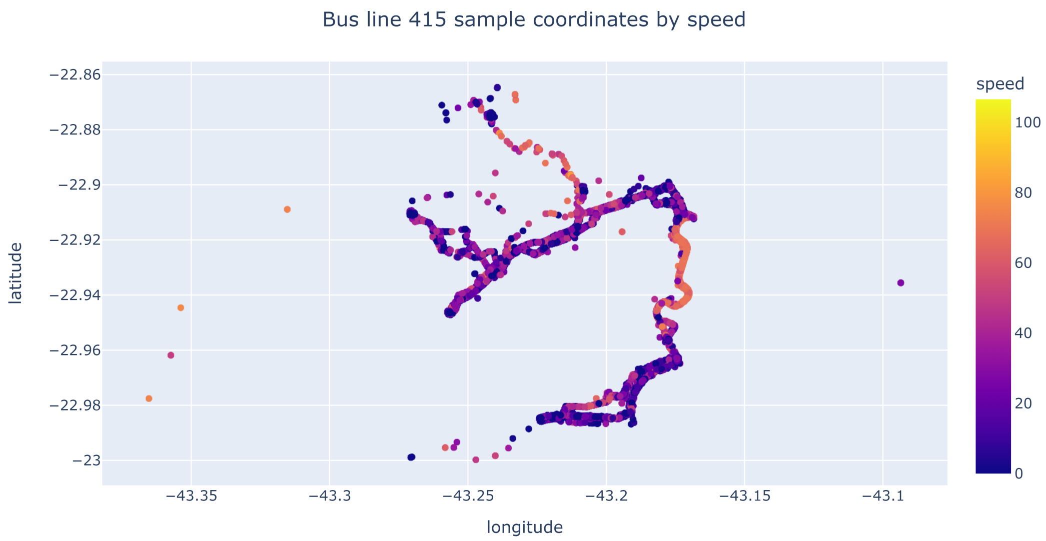

Figure 3.5: Bus line 415 sample points by bus speed.

The bus id distribution in the wandering regions gave no insight into the wandering nature, so we ruled out the issues with a specific bus. The hour of the day distribution suggests a pattern of buses wandering to those regions around midnight. For now, we are assuming that those wandering regions are garages.

An interesting fact is the regions with consistent high-speed samples. Comparing the sample positions with the company route definition (Figure 3.2) it is possible to identify the region as Aterro do Flamengo, which is an expressway with no traffic lights.

3.2.1.3 Preprocessing

While the data set samples mostly fit the official bus route definitions, it lacks context information. It's not clear when the bus started or finished a route, the direction it's going, or if it's heading to a place while out of service, such as a garage. Based on knowledge about the domain, it's possible to add more information to the data set to enhance it to make better use of machine learning techniques.

It's known that most bus lines in Rio de Janeiro have a starting and finishing station. Also, as mentioned in Section 3.2.1.2, we are assuming that the bus goes to the garage at some point. Based on Figure 3.2, we assume that the end and the beginning of the shown routes are the initial and the final stations. We also defined the clear out-of-route destinations on our data set as the two garages. From now on, we will be calling these defined places points of interest, as shown in Figure 3.6.

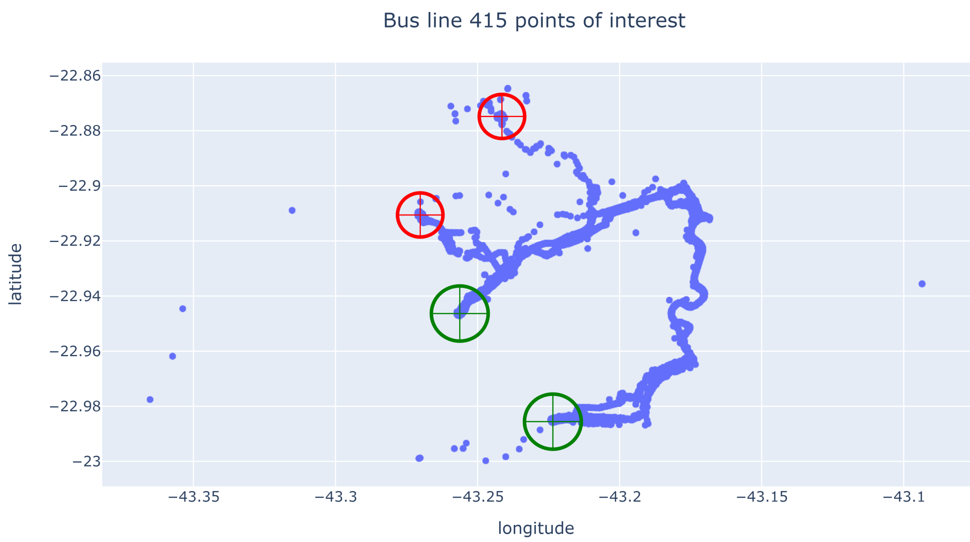

Figure 3.6: Bus line 415 points of interest. Green and red circles represent travel and wanted points, respectively. The radius is the approach region.

The goal of the enhancement is to determine when a trip started, when it finished, and when it wandered off the route. The first step of the algorithm is to sort the data set samples by their time stamp. Then, we mark which samples approached a point of interest. To achieve that, each point of interest has an approach region. When a sequence of samples is inside the approach region, we mark the sample closest to the point of interest with the point of interest name.

With the samples marked, we defined a set of rules to split the samples into sequences that form trips:

1. The samples of the trip must be sequential. There can't be two different trips for the same bus ID in the same timeframe.
2. A trip starts in a sample marked as a final or starting station.
3. A trip ends in the next sample marked as a final or starting station.
4. A trip that starts and ends on a sample marked with the same point of interest is flagged as a circuit.
5. A trip that starts on the final station and ends on the initial station is flagged as reversed.
6. A trip that doesn't pass through all stations besides the initial and final is flagged as cut across.
7. A trip that has a sample marked as a wandering point is flagged as wandering.
8. A trip that has only 1 sample marked as the initial or final station is flagged as incomplete.
9. The duration of a trip is the difference between its first and final sample timestamps.

With the enhanced data set, we are able to analyze the information from a new perspective. Our first step was to filter trips that weren't our focus. We can remove the associated samples from the data set by filtering them.

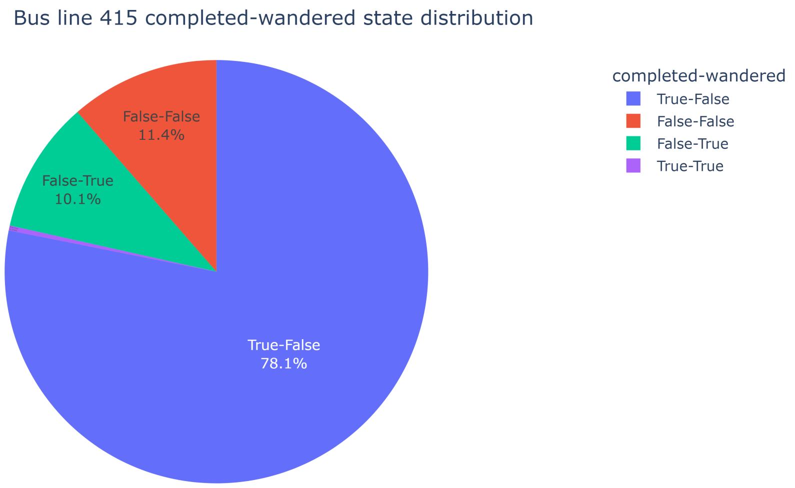

Figure 3.7: Proportion of trips for each completed-wandered state tuples.

We want to remove samples associated with incomplete and wandering trips. Before removing them, we analyzed how many samples we would lose. Figure 3.7 shows that complete and non-wandering trips compose 80 percent of the data set. We considered this number high enough to provide enough data for ML use. All of the coming analysis in this section uses this subset of the data.

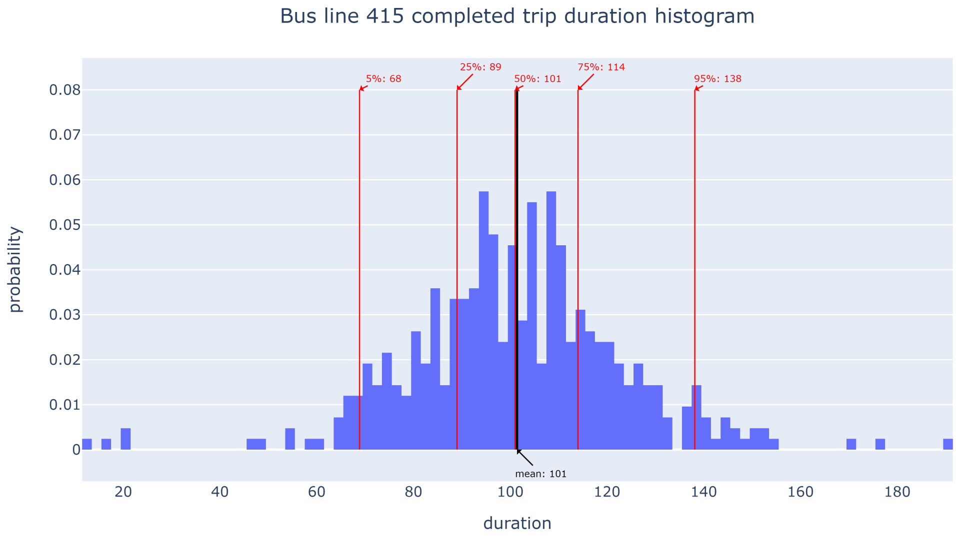

Figure 3.8: Histogram of completed trips duration. The vertical red line marks the 5, 25, 50, 75, and 95 percentiles. The vertical black line marks the mean.

Figure 3.8 shows the histogram we used to check for trips with duration outliers. From the histogram, we can observe that more than 90% of the completed and non-wandering trips fall into a reasonable duration. Trips with a duration below 20 minutes are unlikely to be valid as a normal bus trip. However, due to the lack of information about these trips, and as they represent a very small part of the data set, we decided to keep these potential outliers.

Finally, in Figure 3.9, we have plotted the samples colored by their trip direction. The plot made clear that, depending on the direction of the trip, the bus route is consistently different. This aligns with the fact that some roads are one-way only. Together with the previous analysis of the enhanced data set, we have considered the enhanced data set coherent enough for use in ML.

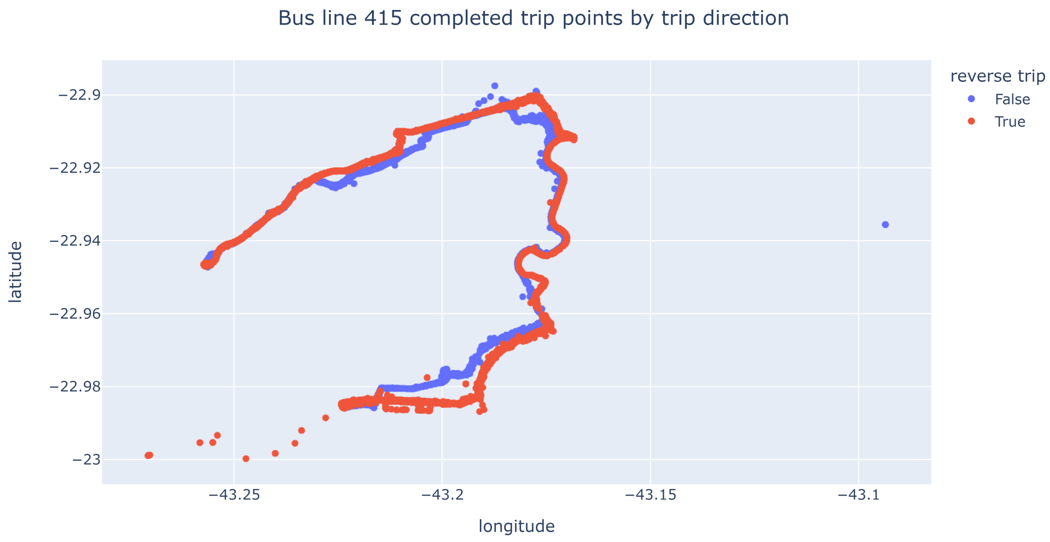

Figure 3.9: Bus line 415 travel points by trip direction based on the enhanced data set. Contains only complete and non-wandering trips.

3.2.2 Model Training and Evaluation

After the enhanced data set was generated, we decided that we had a data set structured enough for a prediction. Our goal is to predict the bus trip duration based on the trip departure time of the day and trip direction.

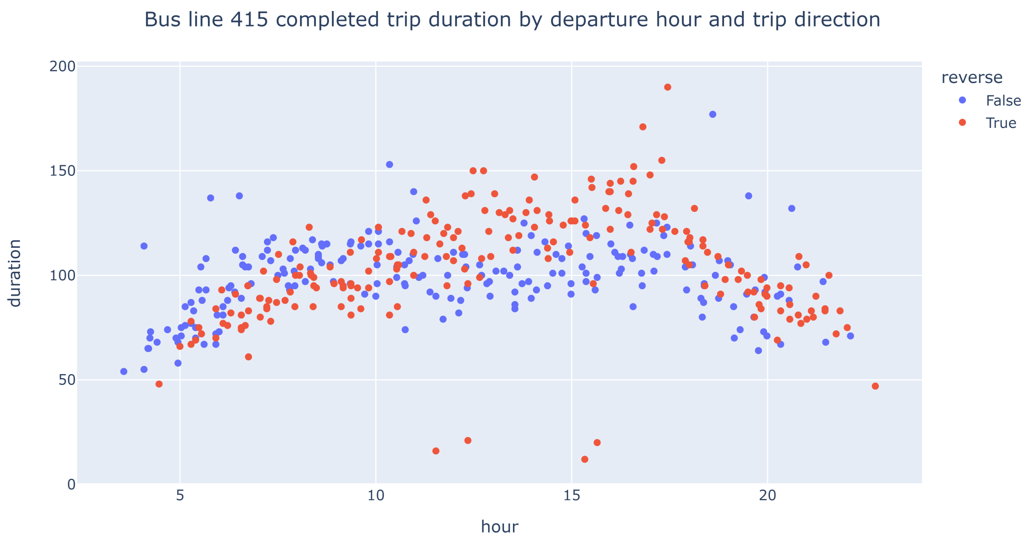

Figure 3.10: Trip duration by departure hour and direction.

Based on the plot shown in Figure 3.10, it is possible to observe that the duration of the trip, based on its hour of departure, forms a pattern. Also, while this pattern displays a visible difference based on the direction of the trip, it is not enough to start our prediction experiment based on this separation, so we used only non-reverse trips for our prediction analysis.

Moreover, in Figure 3.10, we can observe that the pattern could be approachable by a second-grade polynomial function. While this pattern form could be approached by linear regression using a polynomial fit, we decided to use KNN Regression for this experiment due to the scatter observed on the plot.

We used the scikit learn (sklearn) 1.3 (PEDREGOSA et al., 2011). Data was split between training and test in 70% and 30% respectively, executed using the *train_test_split* function provided by the sklearn.

Regarding hyperparameters, to decide how many neighbors to use, we tested all neighbor's quantities between 1 and 30 and measured the root-mean-square deviant (rmse) for all training and test samples. As shown in Table 3.1, around a number of neighbors of 10 the rmse converges between the cross-validation using the training set (train rmse) and the testing set (test rmse), so to avoid overfitting (not generalizable results), we fixed a $n = 10$ for this model. For the distance metric, we used the standard of scikit learn, which is Minkowski. We did use the standard in Figure 3.11 we show the differences between the predicted and observed trip duration.

The prediction model could be further enhanced by testing other algorithms, such as the already mentioned linear regression. The data set model adopted could also be further enhanced by cross-referencing the trip timestamps with the weather forecast databases to account for festivals, soccer matches, and many other events that happen in Rio de Janeiro. However, the main focus of this practical project was to build a proof of concept machine learning model to allow reflection on the impact of MLOps, and for that, we have considered this experience sufficient.

neighbors	train rmse	test rmse
1	0.731544	14.671875
2	6.711409	13.359375
3	7.791946	12.567708
4	8.322148	13.355469
5	8.779866	12.962500
6	8.928412	13.078125
7	9.090125	13.087054
8	8.921141	12.894531
9	8.994780	12.725694
10	8.891275	12.564063
11	9.030506	12.498580
12	9.104027	12.476562
13	9.224058	12.431490
14	9.244008	12.347098
15	9.254586	12.563542
16	9.267198	12.401367
17	9.377813	12.426471
18	9.463833	12.440972
19	9.489933	12.398849
20	9.556040	12.432812
21	9.769255	12.531994
22	9.776388	12.530540
23	9.717245	12.533967
24	9.680928	12.658854
25	9.622282	12.591250
26	9.578988	12.552885
27	9.641561	12.559606
28	9.680968	12.520089
29	9.751215	12.573815
30	9.791499	12.505208

Table 3.1: Knn Root-mean-square deviation of train and test data sets for the number of neighbors ranging from 1 to 30.

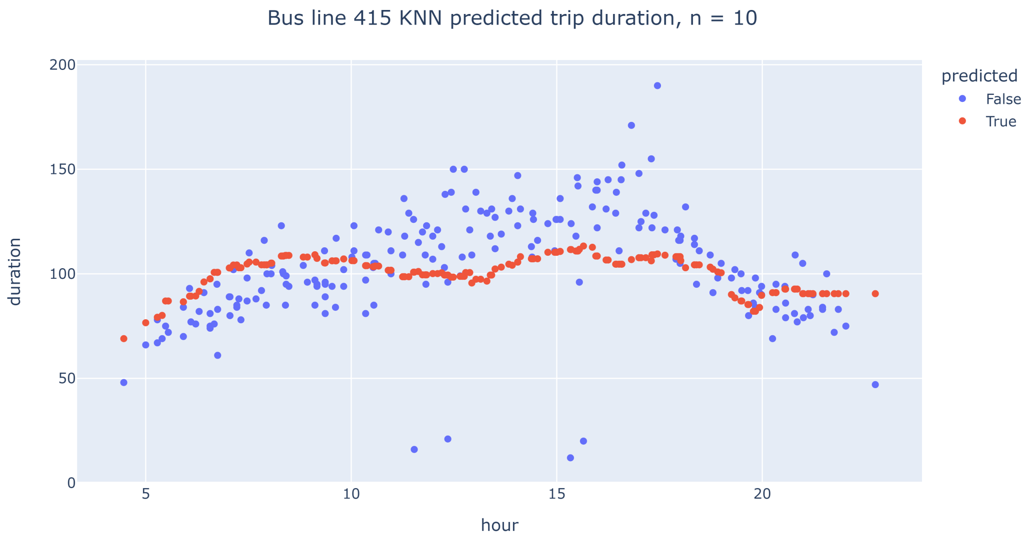

Figure 3.11: Comparison between predicted and observed trip duration for non-reversed trips based on its departure hour.

3.3 Reflections on MLOps within the Practical Project

During the practical experiment developed, we didn't feel the need to use all the MLOps principles discussed in this dissertation. Our understanding is that due to the complexity of the project and its scope (not including dissemination activities such as model deployment and monitoring), not all of the MLOps principles proved necessary. During the creation of the application, the project ran locally on personal machines.

However, we used the versioning principle to be able to deal with the data as it was developed. This principle was used in such a way that, as we studied and improved the data sets (described in Section 3.2.1.3), we versioned the data as we enhanced the data sets. Hence, the study was carried out continuously using data that had already been improved.

On the other hand, as it was an initial project involving offline supervised learning at first, we didn't feel the need to use the MLOps principles we were expecting at first. Despite this, we considered the option of implementing these principles as a way of preparing the project for dissemination. For instance, the implementation of an automated pipeline and the configuration of monitoring on top of the project, in addition to being a good practice, could have brought us some benefits, based on the study of Section 2.4, as follows:

– Improve project monitoring as the project would be configured from the beginning, making it easier to create alarms and monitoring from the initial version.

- Ease the deployment of new versions through an automated pipeline, ensuring that new versions can go into production more quickly and automated, with a limited need for manual interventions.
- Facilitate automating versioning for code, models, and datasets, reducing the risk of inconsistencies.
- Enable using data streaming for incremental learning if the data is collected dynamically and integrated into the automated machine learning pipeline.

Although considering the previously presented benefits, during our development, the academic and offline setting of the practical project was unsuitable for applying the other MLOps principles. However, we believe that if this project makes it into the dissemination phase and gets deployed into production, the need to use these principles would most likely increase.

4
Focus Group: Developer Perception of the Benefits of MLOps for Machine Learning Applications

4.1
Introduction

This chapter describes the methodology used to conduct the two focus group discussions on the benefits and limitations of using the MLOps principles with six experienced machine learning developers and its results. First, Section 4.2 focuses on the methodology, presenting the context and characterization of the participants and how the focus group was designed and conducted. Thereafter, in Section 4.3, we present the detailed focus group results on the benefits and limitations of MLOps principles. Finally, in Section 4.4, we summarize and discuss our main findings.

4.2
Methodology

To assess the MLOps principles from practitioners' point of view, we designed a focus group to promote in-depth expert discussions about the benefits and limitations of applying these principles within ML projects. Focus group is a qualitative research method based on gathering data through the conduction of group sessions, enabling extracting experiences from the participants (KONTIO; BRAGGE; LEHTOLA, 2008). A focus group session is planned for addressing in-depth discussions about a particular topic during a controlled time slot. Additionally, focus group studies have been conducted in software engineering to reveal arguments and feedback from developers (e.g. (MARTAKIS; DANEVA, 2013), (KIM; NOTKIN, 2009), (ALMEIDA et al., 2023)). Thus, we decided to use a focus group as a suitable option for understanding the developer's perception of the benefits of MLOps for supervised online Machine Learning applications.

We conducted two focus group sessions with six expert machine learning developers (three in each session) who have experience creating large-scale machine learning applications and have had contact with both MLOps and non-MLOps-based machine learning application development.

ID	Graduation Level	Years of Experience with Machine Learning	Classification of knowledge in ML	Classification of knowledge in MLOps	Job Title	Industry	Company Size
P1	Bachelor degree	3	High	Medium	ML Engineer	Retail and e-commerce	50,000+
P2	Masters Degree	4	High	Medium	Data Scientist	Oil & gas	40,000+
P3	Masters degree	5	High	Medium	Data Scientist	Oil & gas	40,000+
P4	Masters degree	3	High	Medium	ML Engineer	Oil & gas	40,000+
P5	Masters degree	4	High	Medium	ML Engineer	Oil & gas	40,000+
P6	Bachelor degree	5	Very High	High	Data Science Specialist	Finance	3,000+

Table 4.1: Focus group 1 (P1, P2, and P3) and Focus group 2 (P4, P5, and P6) participants information.

4.2.1 Context and Participant Characterization

We selected participants from three different organizations and industries to gain insights from various perspectives. We conducted two focus group sessions with six experts (P1 - P6), whose details are depicted in Table 4.1. We have collected the table data from an online Participant Characterization Form, which we sent to participants minutes before the start of the focus group session. The characterization questionnaire asked participants for their maximum academic degree, number of years working in the field of machine learning, and a rating of their knowledge of machine learning and MLOps, which could be evaluated in the following possibilities: very low, low, medium, high, very high.

It is possible to observe that all the participants have a high level of knowledge and at least 3 years of experience in developing machine-learning applications. Despite their expertise and having worked on MLOps-based projects, participants did not consider themselves highly knowledgeable on MLOps. This might be because many companies seem to be still maturing their MLOps approaches, with data scientists managing ML workflows manually to a great extent (KREUZBERGER; KüHL; HIRSCHL, 2023).

4.2.2 Focus Group Planning and Design

We carefully designed and performed our focus group by following the guidelines proposed by (KONTIO; BRAGGE; LEHTOLA, 2008). The goal of our focus group can be described following the Goal-Question-Metric goal definition template, as follows: *Analyze* the MLOps principles *with the purpose of* characterizing *with respect to* the benefits and limitations of the MLOps principles *from the point of view of* ML experts *in the context of* supervised online machine learning applications.

Figure 4.1 depicts the steps adopted throughout the focus group. We organized these steps in three major phases: **(1)** Preparation for the focus group session; **(2)** Conducting the focus group sessions; and **(3)** Analyzing the data and reporting the results. We describe each phase and step hereafter.

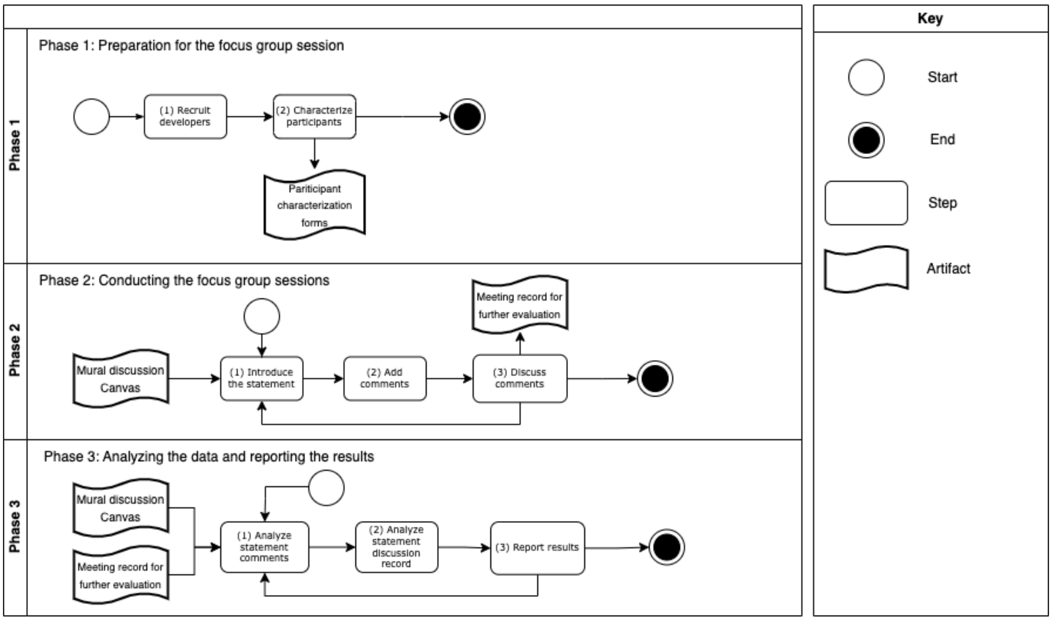

Figure 4.1: Focus Group Overview.

Phase 1: Preparation for the focus group session - This phase consists of the collection of preliminary resources for supporting the execution of the focus group session. For this purpose, we follow two steps.

Step 1 - Recruit developers consisted of recruiting developers with experience in machine learning and MLOps to engage in discussions. We contacted developers from three different organizations and industries to participate in our study. We obtained the acceptance of six experts via a Consent Form in which we explained our research goals and that the information provided by each participant will be treated confidentially and used for study purposes only.

Step 2 - Characterize participants aimed to collect basic information to characterize the participants via the Participant Characterization Form. Our major goal was profiling each developer so we could better interpret our study results. We collected data on the education level, years of experience with machine learning development, and a rating of their knowledge of machine learning and MLOps development (Table 4.1).

Phase 2: Conducting the focus group sessions - This phase consists of collecting data regarding the developer's perception of the benefits and limitations of using the MLOps principles for supervised online machine learning applications. To discuss the benefits and limitations of MLOps, we derived statements regarding commonly used ML-based software delivery metrics (deployment frequency, lead-time for change, and mean time to restore) (VISENGERIYEVA et al., 2023), and asked participants to discuss the effects of using MLOps on these metrics. As these metrics mainly concern automation (including deployment, reproducibility, and testing aspects), we added two additional statements to allow discussions regarding the monitoring and versioning principles. Finally, we added a generic statement on using MLOps principles to gather any additional insights that the experts would like to provide.

We used an online environment to promote discussions on the benefits and limitations of MLOps principles for machine learning applications. Figure 4.2 illustrates the virtual template that we designed using the MIRO online tool. In practice, by using this tool, we were able to build an interactive mural to facilitate the conduction of the focus group sessions.

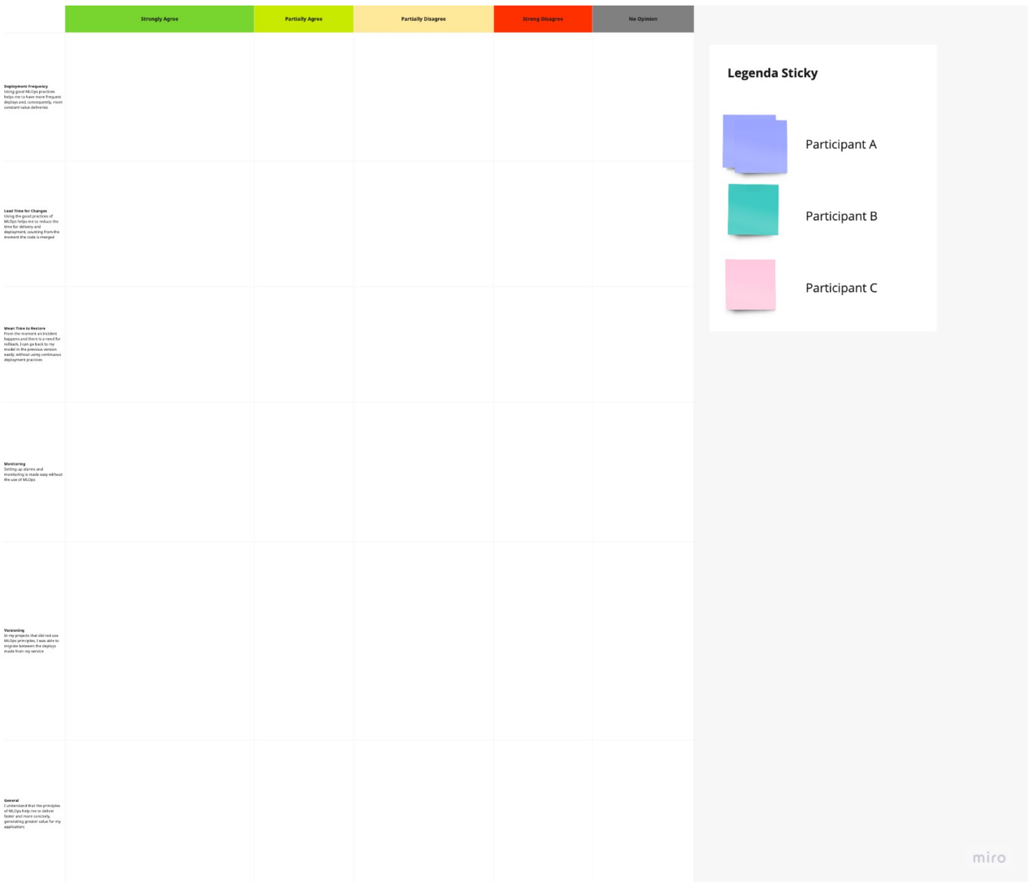

Figure 4.2: Focus Group Template.

Our template is divided into 5 columns that seek to understand whether, for each line containing a statement, the participants, through virtual post-its: strongly agree, partially agree, partially disagree, strongly disagree, or have no opinion over it. Each statement is discussed in isolation, and we defined the dynamics of the focus group session in three steps as follows:

Step 1 - Introduce the statement aimed at introducing each statement. For this purpose, the moderator of the session read this information out loud. Next, in **Step 2 - Add comments**, we asked each participant to add one or more *post-it* for each comment it has over the statement, as in part. Finally, in **Step 3 - Discuss comments**, we asked the participants to explain their comments (and why they agreed or disagreed with the statement) and discussed them within the group. Each comment should be documented as a comment on the post-it as shown in Figure 4.3. The comment was just a brief summary of the reason they selected a column, and we constantly asked participants to share knowledge and experiences surrounding the statement to enrich the discussions and understanding. Additionally, whenever the moderator felt that a comment was poorly written, he asked the participants to provide further considerations.

We emphasize that the focus group session was conducted online via a Zoom Meeting. Additionally, we kept video and audio records of the session to support the data analysis. Both sessions were conducted in August 2023.

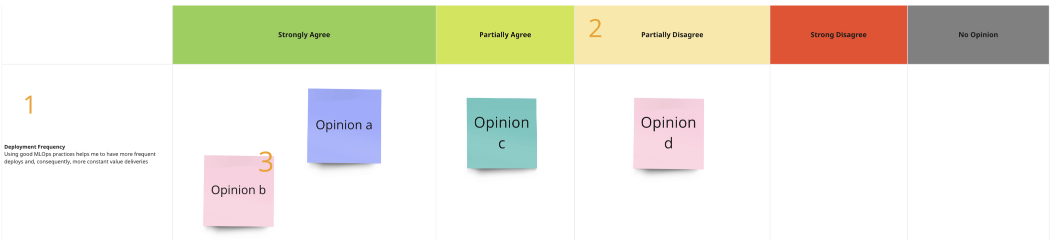

Figure 4.3: Focus Group Template Steps.

Phase 3: Analyzing the data and reporting the results - in this phase, we analyzed the comments for each statement, also referring to the transcribed audio discussion records to better understand what developers meant with each note. The audio records of the Zoom recording were transcribed using a popular speech-to-text transcription model developed at PUC-Rio's Informatics Department (GROSMAN, 2021). We report the results for each statement in the following section.

4.3 Results

We asked the participants to choose and justify whether they agreed or disagreed with the statements reported under the context of the benefits and limitations of using the principles of MLOps for supervised online machine learning applications. We have collected these comments through post-it notes added by the participants in the session's virtual mural (as illustrated in Figure 4.3).

In order to analyze these comments, we first watched the video and automatically transcribed its audio into plain text. Thereafter, we analyzed all post-it comments written by the participants and associated transcription quotes. In the following subsections, we summarize the comments that emerged for each statement.

4.3.1 Deployment frequency: Using MLOps principles helps me to have more frequent deploys and, consequently, more constant value deliveries

This statement focuses on the deployment frequency metric that is strongly related to the automation principle for MLOps (Section 2.4.1). We expected the participants to discuss how MLOps could help the deployment process and generate more valuable and constant deliveries. Four reported to strongly agree, one participant reported to strongly agree and partially agree, and one participant reported to partially agree. We detail each comment as follows, grouped by strongly agree and partially agree ones.

Strongly agree: One of the main topics discussed was that the use of these principles does tend to be favorable for deliveries with more value, but this is also much related to the real needs of the product.

"If you have a "far away" deadline, with broken and well-defined deliveries, i.e. a broad roadmap, the impact of MLOps ends up being huge. If the first version of the product delivers what you need in practice, the first delivery is already what you need, and MLOps ends up having a low impact." - P6

Despite being in different groups, participants P6 and P5 (Focus Group 2), and P3 and P1 (Focus Group 1), stressed the same point. For both groups, the positive impact of the deployment frequency principles is closely related to the product and application context. As P6 stated, if a product doesn't have a high deployment frequency, it is not worth the effort to prepare a well-structured pipeline following the MLOps principles, which require development and preparation time, instead of using that effort on other fronts.

"Creating a well-defined structure can be very complicated. Creating a deployment pipeline can be very good, but it depends on how it's structured. Having an already in-progress pipeline first can help. If a pipeline is robust with good practices, the value ends up being very high, but getting there is complicated." - P3

In P3's opinion, which was agreed with by participants P1 and P2, the maturity of the deployment pipeline process needs to be at a high-automated level for there to be a significant impact. I.e., they believe that the principle can have an effect on the number of deploys and thus generate more value with constant deliveries; however, this is only true with a roadmap containing automation elements (Section 2.4.1) such as pipelines for continuous integration, continuous delivery, model continuous delivery, and monitoring.

Figure 4.4: P3-P1-P2 Strongly Agree post-its to the sentence "Deployment frequency: Using good MLOps practices helps me to have more frequent deploys and, consequently, more constant value deliveries."

Figure 4.5: P6-P5 Strongly Agree post-its to the sentence "Deployment frequency: Using good MLOps practices helps me to have more frequent deploys and, consequently, more constant value deliveries."

Partially agree: P4 and P5 both mentioned that a manual or poorly structured deployment process tends to generate stress and possibly human error and that, in their experience, they have seen that in these scenarios the development team tends to be afraid to carry out deploys and, consequently, tends to reduce the number of deploys made in order to avoid going through the "traumatic" process. P4 and P5 agreed that the ease of deployment creates more opportunities for development to focus on higher-quality deliveries and other points for improvement.

Figure 4.6: P4-P5 Partially Agree post-its to the sentence "Deployment frequency: Using good MLOps practices helps me to have more frequent deploys and, consequently, more constant value deliveries."

Overall, with respect to deployment frequency, we conclude that participants mainly agree that MLOps principles help to generate more frequent deploys and more value deliveries. On the other hand, they also mentioned that if a product doesn't require a high deployment frequency, it might not be worth the effort to prepare a well-structured pipeline following the MLOps principles, which require development and preparation time, instead of using that effort on other fronts.

4.3.2 Lead time for changes: Using the good practices of MLOps helps me to reduce the time for delivery and deployment, counting from the moment the code is merged

In this statement, we approached another important metric related to the automation principle (Section 2.4.1). The statement intended to understand the experts' experiences on how the automation principle can affect the lead time for changes, i.e. do the MLOps principles help the developers to have code merged faster into a production environment? Four participants reported strongly agree, one participant reported both strongly agree and partially agree, and one reported partially agree. We separate each comment as follows, grouped by strongly agree and partially agree ones.

Strongly Agree: Participant P3 assesses this as one of the main impacts of using MLOPs: the more mature your pipeline, the greater the reduction in lead time. The cost of a manual deployment, of misconfiguring non-automated resources, is high. In addition, reducing the need for manual processes also reduces maintenance time due to problems generated during manual deployments. P4 agreed that the reduction in manual processes tends to reduce the time for changes was the main factor, given that we are analyzing from the moment the commit is merged.

P1 agrees and raises the counterpoint seen in the previous statement that this process of pipeline maturity is time-consuming and, consequently, requires effort and time. So it's important to always weigh up when it's time to automate the process so that it's really worth being in automated mode and no longer manual.

Again, both groups address the same points as P3 (Focus Group 1) raised the same discussion as P6 and P4 (Focus Group 2) about reducing errors generated by manual processes that, consequently, require human intervention.

"Good practices such as versioning and automation help corrections as a matter of urgency. When you incorporate good practices, you reduce manual steps and of course, human errors, standardizing changes and speeding up delivery." - P6

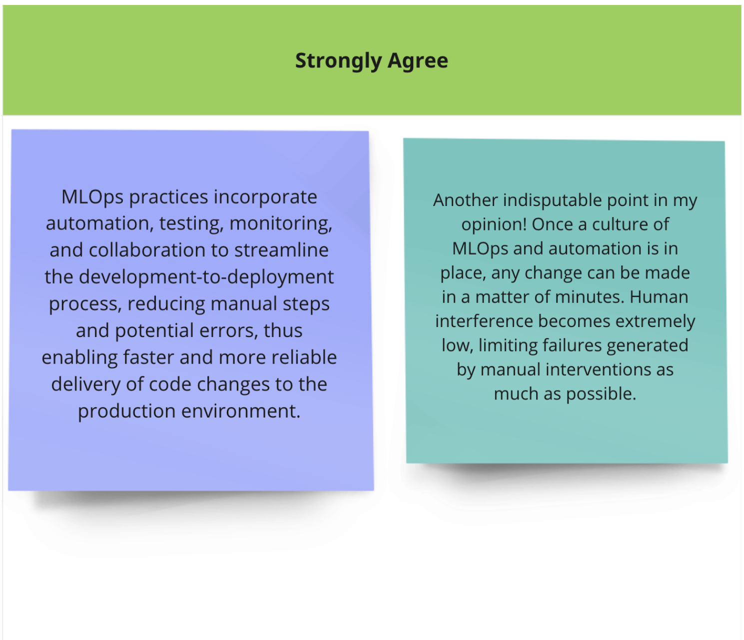

Figure 4.7: P3-P1 Strongly Agree post-its to the sentence "Lead time for changes: Using the good practices of MLOps helps me to reduce the time for delivery and deployment, counting from the moment the code is merged."

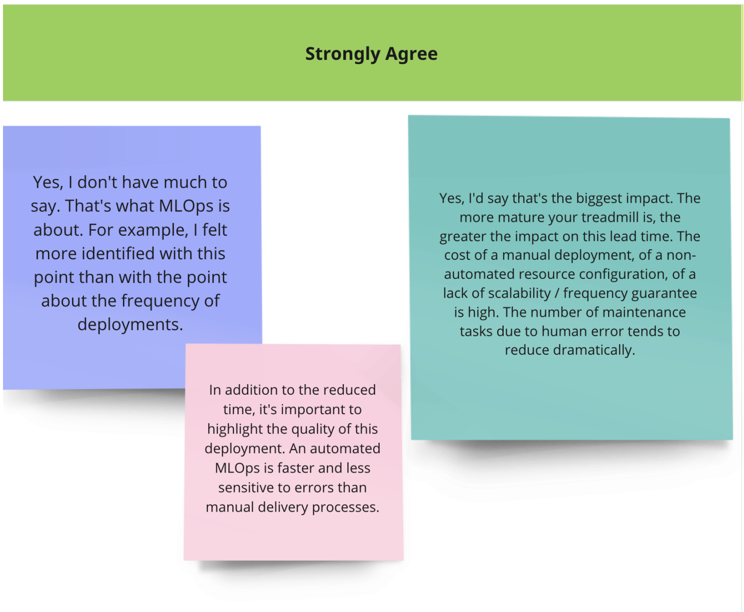

Figure 4.8: P5-P4-P6 Strongly Agree post-its to the sentence "Lead time for changes: Using the good practices of MLOps helps me to reduce the time for delivery and deployment, counting from the moment the code is merged."

Partially Agree: P2 believes that this sentence is "partially correct." In their opinion, sometimes delivering a product with value in a more agile way, with less automation and without following MLOps principles, especially in the so-called initial versions (or v0, i.e. version zero), can be more favorable to the company business. P2 states that it's not necessarily every new product will reach the level of needing to have a well-optimized pipeline and thus a low lead time to change. P1 enforces the argument, again emphasizing how important it is to understand the need for a well-automated deployment structure consistent with your project's real needs, such as its scalability.

Figure 4.9: P2-P1 Partially Agree post-its to the sentence "Lead time for changes: Using the good practices of MLOps helps me to reduce the time for delivery and deployment, counting from the moment the code is merged."

Hence, for lead time to production, it is possible to observe an overall agreement on the reduction of the time for delivery and deployment. Indeed, MLOps principles such as versioning and automation are perceived as helping corrections as a matter of urgency. When you incorporate MLOps principles, you reduce manual steps and human errors, standardizing changes and speeding up delivery.

4.3.3 Mean Time to Restore: From the moment an incident happens and there is a need for rollback, I can go back to my model in the previous version easily, without using continuous deployment practices

This statement was set to understand how the MLOps principles might influence moments of urgency, such as a bug in a production environment. It addresses the mean time to restore metric, i.e. the time it takes an application to recover, usually through a rollback, to a functional state from a non-functional state. This process, which commonly involves a new deployment, tends to be done at moments of tension and stress, as a problem in production can cause damage to the application model, ruining data, as well as possible financial damage to the company, depending on what the application is used to. By bringing up this discussion, our goal is to understand whether, in the experience of our participants, this process done manually and without reproducibility principles and continuous deployment practices is seen as viable.

In this statement, three participants reported partially disagreeing, and three reported strongly disagreeing. Following, we detail each comment grouped by partially disagree and strongly disagree.

Partially disagree: Comments from participants P1, P5, and P6 were on the subject of the maturity of two components: the deployment pipeline and the application as a whole. For P5 and P6, the complexity of their dataset, model, and code influences their ability to do a rollback if necessary. P5 shared that, in a recent experience, it was possible to maintain a description and store each model and dataset created by the development team manually, but as the application gained scope and naturally grew, it became impossible to maintain this manually.

Participant P6 adds that this non-automated versioning, i.e. without following the MLOps principles, can be done, but it is common for one of the three scopes that need to be versioned (data, code, and model) not to be stored in a structured way, making the rollback process difficult. In other words, it is important to have a mature pipeline containing continuous delivery so that it is possible to perform a rollback, but it would not be impossible, in the early stages of a machine learning application, to do this versioning and storage manually, and performing a manual rollback.

"At one of the projects I worked on, I used a platform that helps automate my machine learning application. Even though we paid for this platform to help with the process, it still required a specific configuration from the internal development team to structure the versioning. When we needed to perform a rollback using what was built through the platform, we discovered that there was an error in the way we structured the versioning, and even with a platform helping with the continuous deployment process, we were unable to execute the rollback easily." - P1

Through this practical example from participant P1, in their opinion, as the application grows, versioning needs to be well structured so that rollbacks can be carried out quickly and easily.

Figure 4.10: P1 Partially Disagree post-its to the sentence "Mean Time to Restore: From the moment an incident happens and there is a need for rollback, I can go back to my model in the previous version easily, without using continuous deployment practices."

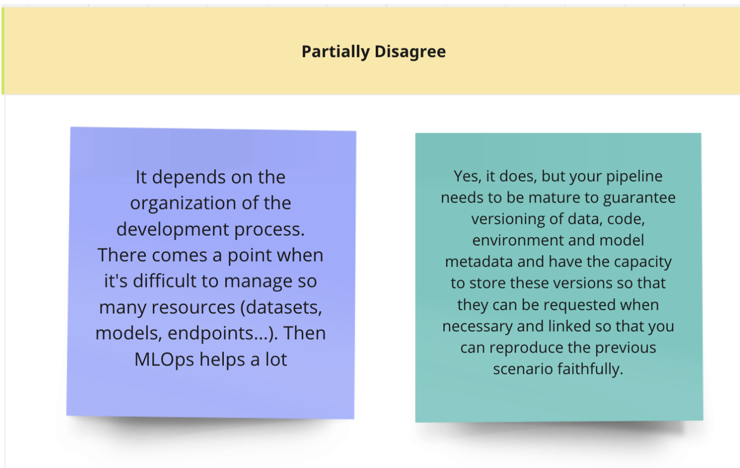

Figure 4.11: P5-P6 Partially Disagree post-its to the sentence "Mean Time to Restore: From the moment an incident happens and there is a need for rollback, I can go back to my model in the previous version easily, without using continuous deployment practices."

Strongly disagree: In P2 and P3's experience, structuring data correctly is an essential part of a supervised online machine learning application. For P2, returning to previous versions without versioning can be impossible. They state that there's no going back if you don't have traceability of the changed data. P3 points out that an application without the principle of versioning as part of the good practices of continuous deployment tends not to be "reliable."

"Without these [continuous deployment practices], it seems that the application will never be reliable. Without good practices and continuous deployment, we lose the reduction of several steps, such as testing, which avoids the causes of incidents and new incidents when restoring an already ongoing incident, which can happen without good practices. We don't have security." - P3

Participant P4 raised a different point of view: the human side of having a problem in production. According to the participant, there is an impact on people under pressure during a crisis, such as a bug in production, which increases the likelihood of human error. Therefore, an automated continuous deployment process is one of the main benefits of following the MLOps principles. In P4's opinion, crisis recovery and decision-making tend to be "easier" when there is an automated process that doesn't need to be taken into account during the crisis.

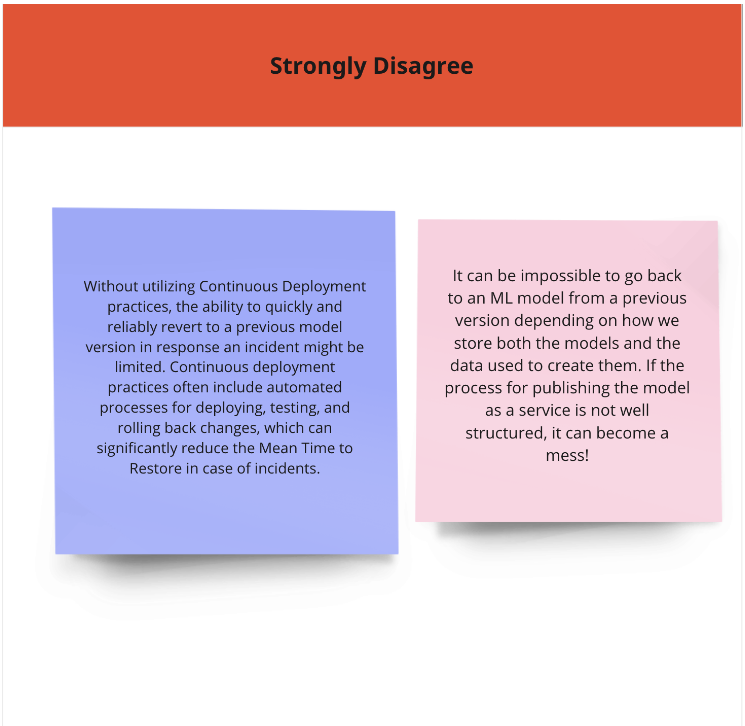

Figure 4.12: P3-P2 Strongly Disagree post-its to the sentence "Mean Time to Restore: From the moment an incident happens and there is a need for rollback, I can go back to my model in the previous version easily, without using continuous deployment practices."

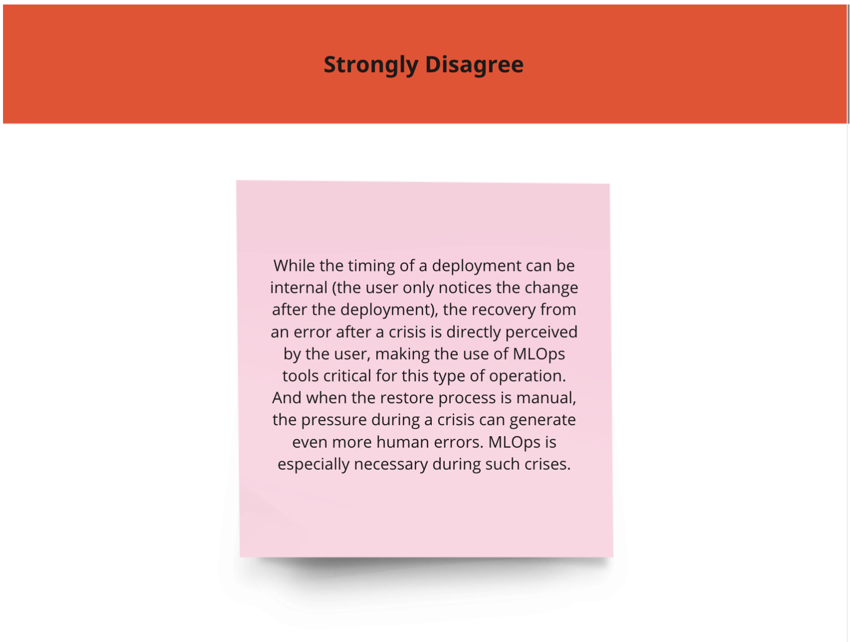

Figure 4.13: P4 Strongly Disagree post-its to the sentence "Mean Time to Restore: From the moment an incident happens and there is a need for rollback, I can go back to my model in the previous version easily, without using continuous deployment practices."

Therefore, with respect to mean time to restore, it is understandable that participants mainly agree that maintaining code, datasets, and model versioned and stored manually in the early stages of a machine learning application will make the processes feasible. However, by providing a structured pipeline following the automation and versioning principles, a machine learning application tends to make the reproducibility and rollback process easier and more successful, avoiding error-prone manual steps.

4.3.4 Monitoring: Setting up alarms and monitoring is made easy without the use of MLOps

This statement's goal is to evaluate the use of MLOps to ensure monitoring by reflecting on whether, in the experience of the experts, it was possible to achieve monitoring and alarmist with quality in their projects without relying on the MLOps principles.

For this statement, one participant reported partially agreeing, two partially disagreeing, one strongly disagreeing, and two reported not having a formed opinion on the topic. The opinions on this statement were very diverse, as well as being the topic with the highest number of participants reporting that they had no opinion on the subject. Following are the opinions provided by the participants.

Partially agree: P6 was the only participant who reported partially agreeing that even without the use of MLOPs, it is possible to create alarms and monitoring in a smooth manner. In the participant's opinion, it is possible, but with a greater amount of effort. P6 pointed out that in their experience in different projects and companies, the monitoring part took a back seat during the prioritization of the project tasks.

In addition, P6 commented that the implementation of monitoring and alarmists arises from necessity when the first "pain" of not having monitoring occurs. In these cases, the negative experience generated by the lack of monitoring creates an urgency to develop it and, consequently, in the need to have something implemented quickly, the practices presented in the MLOps principle were left behind. We then asked: "Even without following the principles, was what was generated considered sufficient?" - the answer was positive, which we believe justifies their partially favorable opinion of the statement.

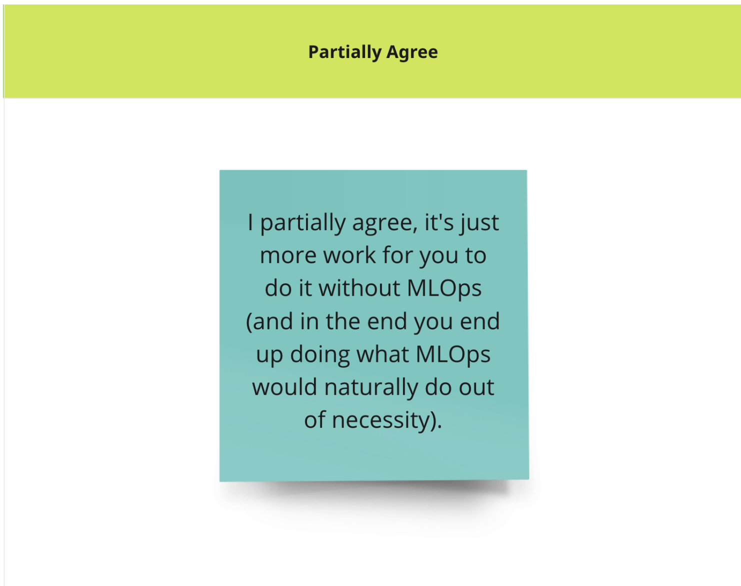

Figure 4.14: P6 Partially Agree post-its to the sentence "Monitoring: Setting up alarms and monitoring is made easy without the use of MLOps."

Partially disagree: Participants P5 and P4 reported partially disagreeing with the statement. In their opinion, the possibility of creating alarms and monitoring, in general, depends little on the implementation process, but they believe that the MLOps principles practiced in areas other than monitoring specifically can facilitate the development of alarms and monitors.

"Even in projects I've worked on that didn't use MLOPs, we already had some alarms that worked well." - P5

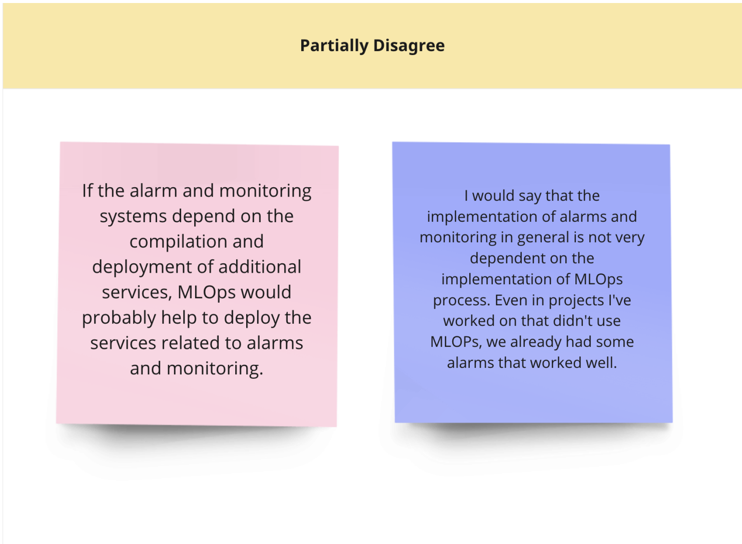

Figure 4.15: P4-P5 Partially Disagree post-its to the sentence "Monitoring: Setting up alarms and monitoring is made easy without the use of MLOps."

Strongly disagree: P3 reported strongly disagreeing that setting up alarms and monitoring is made easy without the use of MLOps. In their opinion, although it is possible to set up monitoring without MLOps, it enhances speed, agility, automation, and integration for more effective and consistent monitoring practices. The participant stated that monitoring following the MLOps principles generates alerts proactively, i.e. following the principles and good practices dictated by MLOps will naturally generate a scenario prone to creating monitoring and alarms more smoothly. The participant states:

"Creating alarms and monitoring without using MLOps is like having a hammer and missing the other tools needed to build a house." - P3

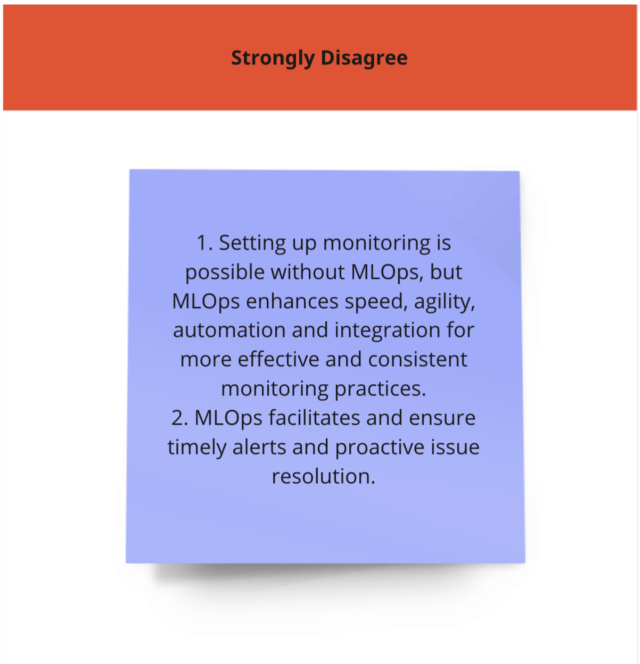

Figure 4.16: P3 Strongly Disagree post-its to the sentence "Monitoring: Setting up alarms and monitoring is made easy without the use of MLOps."

No opinion: Participant P2 stated that they didn't have enough experience on the subject to feel comfortable giving an opinion.

However, participant P1 stated that his decision to place himself without an opinion was due to positive and negative experiences with using and not using the MLOps principles. Like other participants, their belief is that it is possible to create monitoring and alarms without using the principles, but that if the MLOps principles are used, implementation tends to have better quality and is easier. P1 adds that there are trade-offs on both sides depending on terms like the timing of the project, the number of people available, and, consequently, the effort available to implement good practices and MLOps principles.

In their opinion, the area of monitoring and alarming is not considered relevant and talked about in machine learning circles, and although they have seen a greater focus on the subject recently, the topic is generally raised by people who have knowledge of software engineering beyond the application in machine learning projects.

P1 also points out that monitoring doesn't need to be extensive and complex at the outset; identifying an anomaly and sending a warning by email or a communication platform can be considered that an application is being monitored and alarmed, even if it may be inefficient sometimes. P1 points out that the important thing is for this culture of monitoring and alarmism to be discussed "more and more" among machine learning developers so that, through these discussions, the maturity of the machine learning's monitoring culture can be improved.

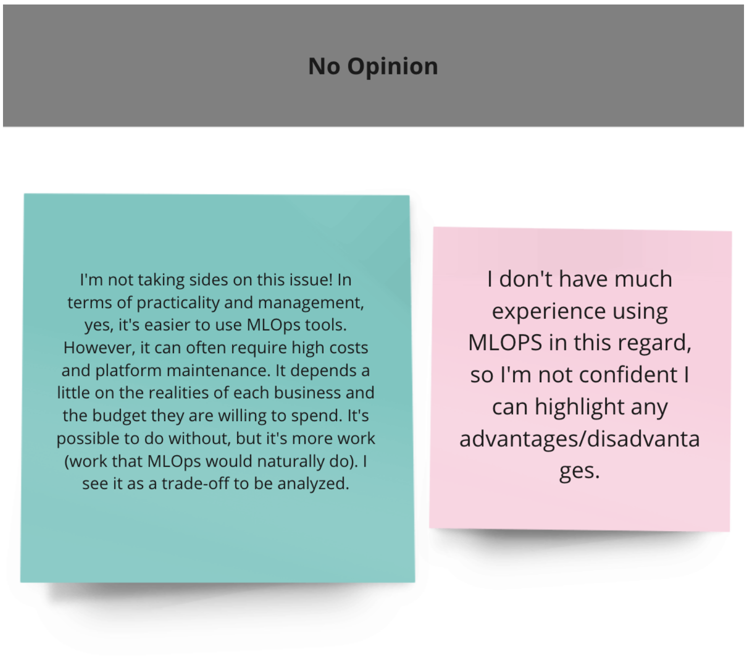

Figure 4.17: P1-P2 No Opinion post-its to the sentence "Monitoring: Setting up alarms and monitoring is made easy without the use of MLOps."

Overall, with respect to monitoring, we can observe that, even though the participants' opinions about the statement were diverse, a common aspect was that monitoring is usually left behind from the application scope in the early stages. We can also conclude that it is possible to build monitoring without following the MLOps principles; however, implementing the MLOps principles will facilitate and come into great hands when developing alarms and monitors.

4.3.5 Versioning: In my projects that did not use MLOps principles, I was able to migrate between the deploys made from my service

Through this statement, our goal is to understand from the experts' opinions whether they consider it feasible to carry out common actions such as rollback and model version exchange (reproducibility) in projects that did not use the MLOps principles in their implementation.

Two participants reported partially disagreeing, and four reported strongly disagreeing with the statement. Following, we present the arguments from the participants over why they partially or strongly disagree with the statement.

Partially disagree: As previously reported, the two participants who reported partially disagreeing with the sentence gave examples of projects they had worked on and were able to maintain without versioning. In all cases, however, it was only possible to maintain the project without versioning principles being implemented in the early stages of the project. P5 stated that not following the MLOps principles implies the need to maintain a very organized manual process so that no time is wasted in times of urgency, such as when performing rollbacks during a bug in the production environment.

Participant P3 stated that they believe it is possible to implement initial forms of versioning without following the MLOps principles, but through good practices, it will be possible to achieve a more systematic and automated approach to version management and migrations.

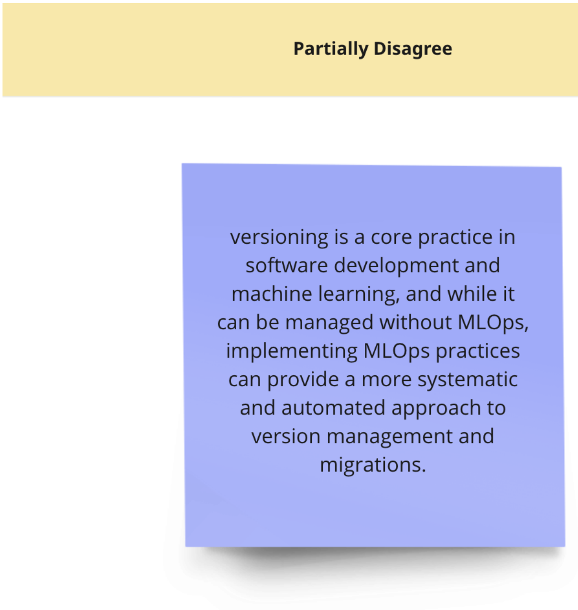

Figure 4.18: P3 Partially Disagree post-its to the sentence "Versioning: In my projects that did not use MLOps principles, I was able to migrate between the deploys made from my service."

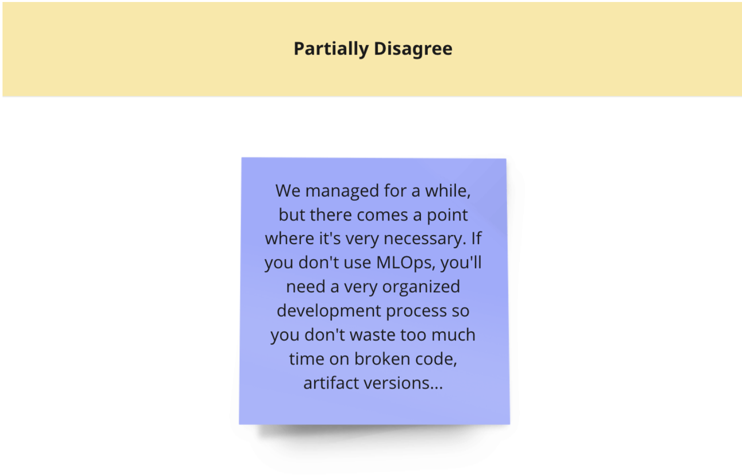

Figure 4.19: P5 Partially Disagree post-its to the sentence "Versioning: In my projects that did not use MLOps principles, I was able to migrate between the deploys made from my service."

Strongly disagree: A common point in almost all the comments from participants who strongly disagree is that the versioning principle is essential for achieving quality. P4 commented on their not believing in the possibility of a team developing an application to an acceptable level of quality without automated version control. The participant also pointed out their very negative experiences in projects that didn't prioritize developing a versioning process at first, and the consequences were negative as the project progressed.

For P1 and P2, the trade-offs of not implementing good versioning MLOps practices are "indisputable." P2 states that the effort spent in the short term is rewarded exponentially over a short period of time through the common need to migrate between versioning code, data, and models.

"You gain little by not having and lose a lot by not having." - P1

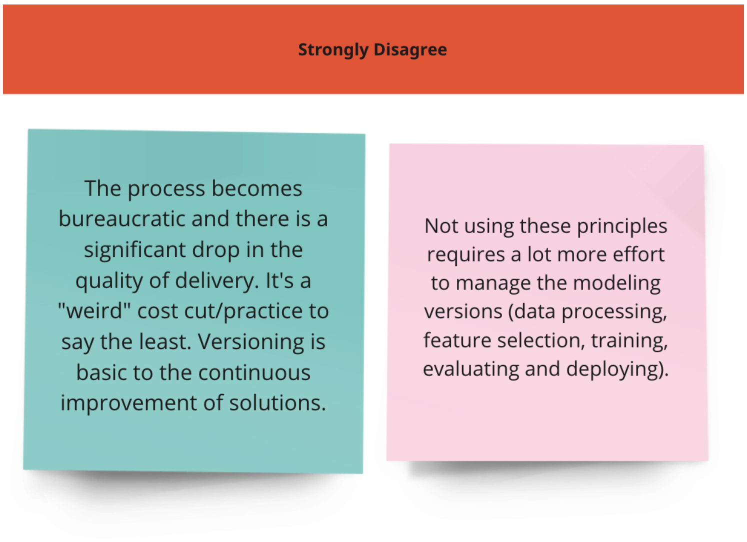

Figure 4.20: P1-P2 Strongly Disagree post-its to the sentence "Versioning: In my projects that did not use MLOps principles, I was able to migrate between the deploys made from my service."

Figure 4.21: P6-P4 Strongly Disagree post-its to the sentence "Versioning: In my projects that did not use MLOps principles, I was able to migrate between the deploys made from my service."

Thereupon, for versioning, it is possible to observe an overall disagreement with the feasibility of running a complex production-ready machine learning application without the MLOps principles. Aside from early-stage machine learning applications, versioning plays an important part in making possible reproducibility by enabling previous datasets and models to be identified and facilitating the application to perform rollback to previously trained models and labeled datasets.

4.3.6 General MLOps principles: I understand that the principles of MLOps help me to deliver faster and more concisely, generating greater value for my application

This statement was added as a generic statement on using MLOps principles to gather any additional insights that the experts would like to provide. It brings the focus to the use of MLOps principles in delivering complex and valuable machine learning applications, suggesting the use of MLOps principles as a tool that helps developers to deliver faster and more concisely.

Hence, the discussion of this last topic was almost a summary of all the points covered in the previous statements. Indeed, we had five participants who reported strongly agreeing and one who reported partially disagreeing. Below, we detail each of these comments grouped by strong agreement and partial disagreement.

Strongly agree: Understandably, this was the statement with the least heterogeneous range of responses between the two groups of participants. However, the comments by which the participants justified their decision were diverse.

For P2, the principles generate faster and more valuable deliveries, especially in the automation part of the life cycle.

Participant P1 states that development should avoid going overboard with the use of tools that don't make sense for the project. The participant comments that the important thing, in their opinion, is that there is clear planning about what needs to be delivered as a product from the start of the project and that this planning should take into account "the good practices implementation", as part of the basic implementation of a machine learning application.

In their experience, the problem usually lies when the project planning isn't done precisely and, moreover, is done by company teams that don't take into account the entire scope of a project with implementation quality.

"You need to know your reality and your problem in order to know how to act on it. MLOps is not a job, it's a culture, and if you want to use MLOps you have to face it as a culture and bring people into your team who follow

and believe in that culture to get there." - P1

Participant P3 adds that creating an environment that uses the MLOps principles is a constant duty of self-awareness, which is independent of the maturity of the company or project in which it is being developed.

"The delivery of valuable deployments is continuous, the machine learning applications can have multiple functionalities and each delivery generates more value. To be fast is to be consistent with the schedule, and the principles of MLOps favor this delivery in the schedule if it is well suited. The biggest problem is how to make this schedule fit, having to align it with the product team. We lack the maturity to be able to bring these principles into the schedule." - P3

Participant P6 said that their experience with "the two sides of the same coin" highlights that the project creation process is closely linked to the need to follow and use these principles. For the participant, the creation process should make the project's intent clear and this should be enough to decide whether or not to implement the MLOps principles. A project with a short scope, with low maintenance needs, is not usually worth the effort of having a well-planned and thought-out structure using the MLOps principles; on the other hand, a large-scale, complex project, mostly online, has the implementation of the MLOps principles as an almost essential basis for the success of the project, with a view to quality of the project in the long term.

Figure 4.22: P1-P3-P2 Strongly Agree post-its to the sentence "General: I understand that the principles of MLOps help me to deliver faster and more concisely, generating greater value for my application."

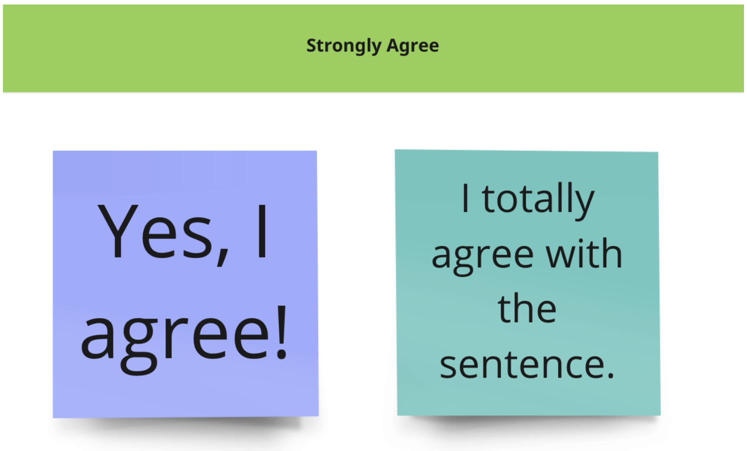

Figure 4.23: P5-P6 Strongly Agree post-its to the sentence "General: I understand that the principles of MLOps help me to deliver faster and more concisely, generating greater value for my application."

Partially disagree: Participant P4 was the only one who declared not to strongly agree with the statement. Despite the surprising opposite of the majority, the explanation of why the participant chose to disagree with the statement was considered fair and reasonable by the other participants.

The participant explains that although they have knowledge about MLOps and their benefits, this knowledge was not obtained in their work environment but through private interest and study. P4 points out that the problem, in their opinion, is not the characteristics of MLOps but rather both the academia and industry environments, which do not have MLOps as part of their culture. They believe that today there is a need to create a culture of knowledge and use of MLOps, especially in academia, and that this maturing will increasingly influence the need to use these principles, i.e. as few companies use these principles today, their benefits are not seen competitively.

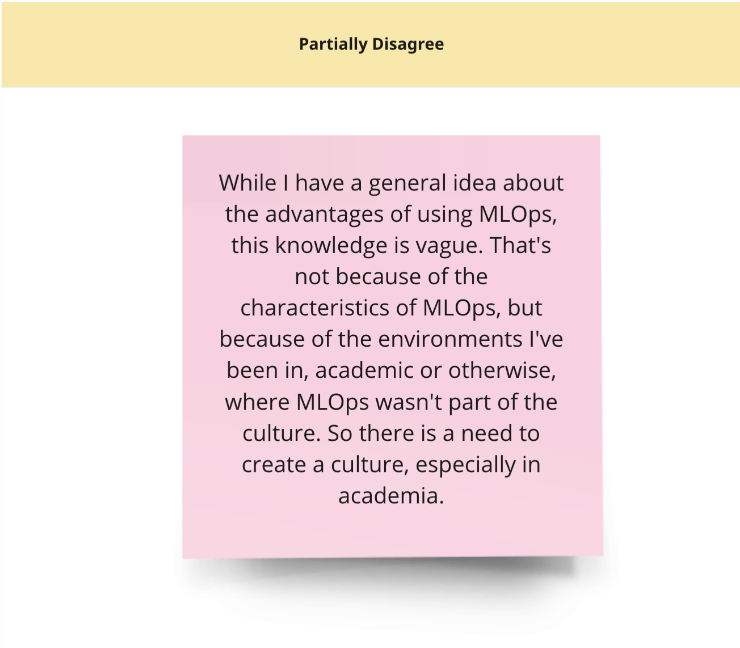

Figure 4.24: P4 Partially Disagree post-its to the sentence "General: I understand that the principles of MLOps help me to deliver faster and more concisely, generating greater value for my application."

While there is mainly an agreement, the partially disagreeing comment is similar to what we observed in the previous statements: the use of MLOps principles is closely related to the need and capacity to implement them. Although the benefits are understandable compared to mature projects, implementing the MLOps principles could imply changing and delaying the planned roadmap, which is usually focused on delivering usable results. The culture of using MLOps as a basic part of developing complex software with quality is not yet so disseminated.

4.4 Discussion

In Section 4.3, we collected from the participants multiple insights that we believe assess the benefits and limitations of using MLOps for supervised online machine learning applications. Indeed, from the enriching discussions, we summarize some of the highlights that were brought up:

- The development team should consider the product scope and roadmap before implementing some MLOps principles. If a product doesn't have a high deployment frequency, it might not be worth the effort to prepare a well-structured pipeline following the MLOps principles, which require development and preparation time.
- MLOps practices can help corrections as a matter of urgency. When the development team incorporates MLOps principles such as automation and versioning, it might reduce manual steps and human errors, standardizing changes and speeding up delivery.
- Performing an application version change or a rollback is possible without implementing MLOps principles such as an automated deployment pipeline and versioning for an early-stage application. Even so, it is common for an early-stage non-automated pipeline to fail or make the rollback process longer.
- Monitoring usually takes a back seat during the prioritization of the project tasks. That might explain the multiple arguments and viewpoints the two groups discussed over the topic. It was stated that it's possible to create alarms and monitoring, even if they are not too elaborate. Even so, having MLOps principles implemented will facilitate and come into great hands when developing alarms and monitors.
- Keeping the implementation at a good quality level without versioning is feasible in the early stages of the implementation. Afterward, it usually becomes complicated to maintain an application that does not have versioning of code, model, and/or data.
- The culture of creating good quality software by following good engineering practices, including the MLOps principles, is still not so well spread over the academia and industry machine learning companies. Creating an environment that uses the MLOps principles is a constant duty of self-awareness and should be aligned with the project requirements. It may be facilitated by the maturity of the company or project in which it is being developed.

5
Conclusion

5.1
Contributions

This dissertation assessed the benefits of MLOps for supervised online machine learning applications. To address our goal, we first developed a practical supervised machine learning project to deepen the understanding of the problem and the MLOps principles usage possibilities. Thereafter, we conducted two focus group sessions among experienced machine learning developers to understand the benefits and limitations of using the MLOps principles.

The practical project implemented a supervised regression machine learning application using KNN. The application uses information on Rio de Janeiro's public bus line routes and calculates the bus trip duration based on the trip departure time of the day and trip direction. Due to the scope of the first version and given that it was not deployed into production, we didn't feel the need to use the MLOps principles we were expecting at first. Indeed, we identified the need for only one principle, the versioning principle, to align versions of the code and the data. Nevertheless, we present our interpretation of the benefits that could be achieved if other principles were implemented.

As the second part of our goal, to assess the benefits of MLOps principles for supervised online machine learning applications, the conducted focus group revealed that machine learning developers believe that the benefits of using MLOps principles are many but that they do not apply to all the projects. It is important to balance the trade-offs of investing time and effort in implementing the MLOps principles, considering the scope and needs of the project. According to the experts, this investment tends to pay off for larger applications with continuous deployment that require well-prepared automated processes. On the other hand, for initial versions of machine learning applications, the effort taken into implementing the principles might enlarge the scope of the project and increase the time needed to deploy a first version to production.

The main benefits include improving deployment frequency and reducing the lead time for changes. Following MLOps principles also helps avoid error-prone manual steps, enabling to restore the application to its previous state, and having a robust continuous automated deployment pipeline. Additionally, having MLOps principles implemented will facilitate and come into great hands when developing alarms and monitors, providing consistent versioning of code, model, and data.

The culture of using MLOps principles is still not so well spread over academia and machine learning companies. Creating an environment that uses the MLOps principles is a constant duty of self-awareness, which should be independent of the maturity of the company or project in which it is being developed. The operation of large complex real-world applications is still too little focused on the academic space.

5.2 Limitations and Future Work

While we considered the data set proposed for our practical project adequate for the analysis goals, it could be further improved to require less knowledge about the domain. We also encourage considering different algorithms and hyperparameter tuning options as a possible improvement to the project prediction results. Questions such as "Is it possible to determine when the bus wandered using only the sample patterns?" or "Using the speed information of the sample, is it possible to detect when the bus arrived at the final station?" were raised during the execution of the data enhancement. These questions could be explored in future work.

The enhanced data set could be further improved by cross-referencing the trip timestamps with other foreseeable conditions, such as weather conditions, traffic affecting festivals, weekdays, and school calendars. How the expanded set of variables would allow for more precise and complete prediction algorithms is an interesting topic to revisit.

As one of our limitations, moving forward with the project as mentioned above, it will also be possible to analyze how other principles besides versioning would be used and their benefits in practice. As an example, to answer the questions above with data collected in real-time, the application needs to go online, making it possible to test the impact of deployment and automation when deploying the application to a live production environment. We believe that the benefits of MLOps are related to the scope and necessities of the application. Thus, with the development of the mentioned improvements to the project, the MLOps benefits will be most likely reached.

Our focus group dynamic was able to assess the benefits and limitations of MLOps principles for supervised online machine learning applications. However, further and other types of empirical studies should be conducted to reproduce the investigation more deeply assessing the MLOps principles.

Thus, future work in this direction might complement the revealed insights of our study.

6
Bibliography

ALMEIDA, C. et al. Negative effects of gamification in education software: Systematic mapping and practitioner perceptions. **Information and Software Technology**, Elsevier, v. 156, p. 107142, 2023.

ALZUBI, J.; NAYYAR, A.; KUMAR, A. Machine learning from theory to algorithms: an overview. In: IOP PUBLISHING. **Journal of physics: conference series**. [S.l.], 2018. v. 1142, p. 012012.

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

AYKOL, M.; HERRING, P.; ANAPOLSKY, A. Machine learning for continuous innovation in battery technologies. **Nature Reviews Materials**, v. 5, n. 10, p. 725–727, jan. 2020.

BRECK, E. et al. The ml test score: A rubric for ml production readiness and technical debt reduction. In: IEEE. **2017 IEEE International Conference on Big Data (Big Data)**. [S.l.], 2017. p. 1123–1132.

BRECK, E. et al. The ml test score: A rubric for ml production readiness and technical debt reduction. In: IEEE. **2017 IEEE International Conference on Big Data (Big Data)**. [S.l.], 2017. p. 1123–1132.

CALEFATO, F. et al. Assessing the use of automl for data-driven software engineering. **arXiv preprint arXiv:2307.10774**, 2023.

DRIDI, S. Supervised learning-a systematic literature review. OSF Preprints, 2021.

EPPERSON, W. et al. Strategies for reuse and sharing among data scientists in software teams. In: **Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice**. [S.l.: s.n.], 2022. p. 243–252.

ESCOVEDO, T.; KOSHIYAMA, A. **Introdução a Data Science: Algoritmos de Machine Learning e métodos de análise**. [S.l.]: Casa do Código, 2020. 174-177 p.

GOOGLE. **MLOps: Continuous delivery and automation pipelines in machine learning**. 2023. (accessed: 03-30-2023). Disponível em: <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning#top_of_page>.

GREFF, K. et al. The sacred infrastructure for computational research. In: HUFF, K. et al. (Ed.). **Proceedings of the 16th Python in Science Conference**. [S.l.: s.n.], 2017. p. 49 – 56.

GROSMAN, J. **Fine-tuned XLSR-53 large model for speech recognition in English**. 2021. <https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english>.

HERAS, A. D. L.; LUQUE-SENDRA, A.; ZAMORA-POLO, F. Machine learning technologies for sustainability in smart cities in the post-covid era. **Sustainability**, v. 12, n. 22, 2020. ISSN 2071-1050. Disponível em: <https://www.mdpi.com/2071-1050/12/22/9320>.

IBM. **What is supervised learning?** 2023. (accessed: 03-24-2023). Disponível em: <https://www.ibm.com/topics/supervised-learning>.

IBM. **What is the k-nearest neighbors algorithm?** 2023. (accessed: 03-30-2023). Disponível em: <https://www.ibm.com/topics/knn>.

ITERATIVE.AI. **DVC - Data version control**. 2023. (accessed: 09-20-2023). Disponível em: <https://dvc.org/>.

JABBARI, R. et al. What is devops? a systematic mapping study on definitions and practices. In: **Proceedings of the Scientific Workshop Proceedings of XP2016**. New York, NY, USA: Association for Computing Machinery, 2016. (XP '16 Workshops). ISBN 9781450341349. Disponível em: <https://doi.org/10.1145/2962695.2962707>.

KALINOWSKI, M. et al. **Engenharia de Software para Ciência de Dados: Um guia de boas práticas com ênfase na construção de sistemas de Machine Learning em Python**. [S.l.]: Casa do Código, 2023.

KIM, M.; NOTKIN, D. Discovering and representing systematic code changes. In: **Proceedings of the 31st International Conference on Software Engineering**. USA: IEEE Computer Society, 2009. (ICSE '09), p. 309–319. ISBN 9781424434534. Disponível em: <https://doi.org/10.1109/ICSE.2009.5070531>.

KOCIELNIK, R.; AMERSHI, S.; BENNETT, P. N. Will you accept an imperfect ai? exploring designs for adjusting end-user expectations of ai systems. In: **Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 2019. (CHI '19), p. 1–14. ISBN 9781450359702. Disponível em: <https://doi.org/10.1145/3290605.3300641>.

KONTIO, J.; BRAGGE, J.; LEHTOLA, L. The focus group method as an empirical tool in software engineering. In: ______. **Guide to Advanced Empirical Software Engineering**. London: Springer London, 2008. p. 93–116. ISBN 978-1-84800-044-5. Disponível em: <https://doi.org/10.1007/978-1-84800-044-5_4>.

KREUZBERGER, D.; KüHL, N.; HIRSCHL, S. Machine learning operations (mlops): Overview, definition, and architecture. **IEEE Access**, v. 11, p. 31866–31879, 2023.

LWAKATARE, L. E. et al. From a data science driven process to a continuous delivery process for machine learning systems. In: SPRINGER. **Product-Focused Software Process Improvement: 21st International Conference, PROFES**

2020, Turin, Italy, November 25–27, 2020, Proceedings 21. [S.l.], 2020. p. 185–201.

MÄKINEN, S. et al. Who needs mlops: What data scientists seek to accomplish and how can mlops help? In: IEEE. **2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)**. [S.l.], 2021. p. 109–112.

MARTAKIS, A.; DANEVA, M. Handling requirements dependencies in agile projects: A focus group with agile software development practitioners. In: . [S.l.: s.n.], 2013. v. 2013, p. 1–11.

MEULEN, R. van der; MCCALL, T. **Gartner Says Nearly Half of CIOs Are Planning to Deploy Artificial Intelligence.** 2018. <https://www.gartner.com/en/newsroom/press-releases/2018-02-13-gartner-says-nearly-half-of-cios-are-planning-to-deploy-artificial-intelligence>. Accessed: 2023-03-16.

MOOVIT. **Bus line 415 official route**. 2023. (accessed: 06-30-2023). Disponível em: <https://moovitapp.com/index/pt-br/transporte_p%C3%BAblico-line-415-Rio_de_Janeiro-322-1847721-29396639-2>.

NAQA, I. E.; MURPHY, M. J. What is machine learning? In: _____. **Machine Learning in Radiation Oncology: Theory and Applications**. Cham: Springer International Publishing, 2015. p. 3–11. ISBN 978-3-319-18305-3. Disponível em: <https://doi.org/10.1007/978-3-319-18305-3_1>.

Pandey, M.; Rautaray, S. S. **Application of Machine Learning in Industry 4.0. In Machine Learning: Theoretical Foundations and Practical Applications.** [S.l.]: Springer Singapore, 2021. 57-87 p.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

POSOLDOVA, A. Machine learning pipelines: From research to production. **IEEE Potentials**, v. 39, n. 6, p. 38–42, 2020.

PROJECTS, L. L. **MLFlow Project**. 2023. (accessed: 09-20-2023). Disponível em: <https://mlflow.org/>.

RAY, S. A quick review of machine learning algorithms. In: IEEE. **2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)**. [S.l.], 2019. p. 35–39.

SHALEV-SHWARTZ, S. Online learning and online convex optimization. **Foundations and Trends® in Machine Learning**, v. 4, n. 2, p. 107–194, 2012. ISSN 1935-8237. Disponível em: <http://dx.doi.org/10.1561/2200000018>.

SHETTY, S. H. et al. Supervised machine learning: Algorithms and applications. **Fundamentals and Methods of Machine and Deep Learning: Algorithms, Tools and Applications**, Wiley Online Library, p. 1–16, 2022.

STODDEN, V.; BORWEIN, J.; BAILEY, D. H. Setting the default to reproducible. **computational science research. SIAM News**, v. 46, n. 5, p. 4–6, 2013.

VISENGERIYEVA, L. et al. **MLOps Principles**. 2023. (accessed: 03-24-2023). Disponível em: <https://ml-ops.org/content/mlops-principles>.