MLOps ARISA - Lesson 1 - Motivating the Principles of MLOps

Introduction - What is MLOps?

According to the Continuous Delivery Foundation, "MLOps could be narrowly defined as “the ability to apply DevOps principles to Machine Learning applications“ however [...], this narrow definition misses the true value of MLOps to the customer. Instead, we define MLOps as “the extension of the DevOps methodology to include Machine Learning and Data Science assets as first class citizens within the DevOps ecology“." [1]

In order words, the field of MLOps seeks to apply DevOps principles and methods, which are quite a bit more mature, to the fields of Data Science (DS) and Machine Learning (ML), which are comparatively in their infancy.

This definition, however, might be difficult to understand at a glance, without already being familiar with the details and goals of both DevOps and ML.

Rather than trying to explain, from a purely theoretical perspective, what MLOps is supposed to be, in this course we will be starting from what is the typical endpoint (no pun intended) of a data science project, namely a proof-of-concept (PoC) trained model. These typically come in the form of one or more jupyter notebooks, with a mix of data preprocessing and training code, as well as an inference step where predictions are made on unseen data, usually to evaluate model performance.

To use a more everyday analogy, one which founder Dr John Elder of the industry leading DS and ML Consultancy firm Elder Research[1], is fond of, DS and ML is akin to building a car engine, where MLOps then is constructing the rest of the car.

Depending on the maturity of the organization in question, data engineers might be brought in on occasion, to build a proverbial driveway (data infrastructure) and so on.

As mentioned, since MLOps is still a relatively young field, there are no strictly agreed upon standards and principles that everyone follows when starting a DS/ML project.

There are, however, several attempts at codifying general MLOps principles or roadmaps, early attempts including

the 2010 Google paper[2] [2], which discusses Google's experiences with building an end-to-end Machine Learning platform. While the paper does not even mention the word

[1] See supplementary material 1
[2] See supplementary material 2

MLOps, it still contains valuable insight on the topic regardless, and "The Rules Of Machine Learning" codified in this and earlier papers are essential principles to follow, when building machine learning systems.

Other attempts at codifying MLOps include the Microsoft Azure MLOps Maturity Model [3], as well as the quite more verbose MLOps Roadmap by the Continuous Delivery Foundation [1].

In this course we will be using the steps outline in the MLOps maturity assessment[3] by Maria Vechtomova of Marvelous MLOps [4], (a mainly Databricks focused blog, but containing a goldmine of information about general MLOps principles and practices).

The MLOps maturity assessment outlines the steps to take, and solutions to put in place to be able to consider oneself or one's organization as "MLOps Mature". Those steps are:

1. Documentation

   Business goals/problem that the ML project is addressing, who is involved, as well as general model documentation about choices made in processing data, creating features and training model.

2. Traceability & Reproducibility

   Solutions should be put in place to make sure that each version of a model can be reproduced in terms of parameters, specific dataset, code needed to run training and predictions. In general, it should be possible to restore the ML system to any previous point in time in terms of model and data version and reproduce those same predictions.

3. Code quality

   Whenever code is committed to a version control system, and before it is put into production, it is checked for quality, i.e. whether it conforms to established code quality guidelines and principles. Part of this step is also having a peer review process in place to make sure more people than the original author looks over the code and approve any changes.

4. Monitoring & Support

   Requests to a deployed model, as well as responses should be logged, tracking inputs and outputs, measuring data drift and shifts in model performance. General infrastructure logs to measure and diagnose health of system.

[3] See supplementary material 3

5. Data transformation pipelines & Feature store

   Features are either precomputed and available in the form of a feature store (mostly in big organizations), or at the very least, feature calculation code is stored as code and used in reproducible and automated data pipelines to provided needed data at train and prediction time. Features are easily able to be added to existing pipelines.

6. Model explainability

   Model outputs are accompanied by explanations, allowing regular non-technical users to understand how the model generated an output from a specific set of inputs.

7. A/B testing & Feedback loop

   A/B testing is put in place to validate that any new version of a model is performing better than the previous one, including reproducible responses for each model version and a way for users to provide feedback in case a model is not operating in compliance with business requirements.

As per the rules of machine learning, this journey towards adopting MLOps does not have to be a linear one.

We do not have to have exhaustive documentation of every nook and cranny of our model development and training before we can start adding traceability and reproducibility to our system. As stated in the rules, it is okay to start simple, get a minimally useful infrastructure up and running, and then iterate upon that basis. In the field of ML/DS, and software engineering in general, we are not building skyscrapers where we need to finish the floor below, before moving up. In project management terms, MLOps seeks to use Agile principles rather than waterfall.

Having laid out the steps on the road to MLOps maturity it becomes clear that there is a lot to cover, and we will probably not reach all the way to step 7 by the end.

This course should be considered more of an MLOps 101, i.e. a beginner's crash-course on the topic, since it would be unrealistic to cover every part of MLOps, given the time allotted. As MLOps exists at the intersection between DevOps, Data Engineering (DE), Machine Learning Engineering (MLE) and Data Science (DS), certain aspects must be given priority over others. For this course, the focus is on model governance, i.e. how models are created, trained, put into production and monitored thereafter, all in a systematic and reproducible fashion. Less focus will be put on the topics of data pipelines and feature stores, which are better left to a dedicated data engineering course. It should also be noted that the course will mainly be concerned with open-

source tools, as the goal is to convey principles rather than how to use certain paid for platforms and programs (for which ample training resources are already available online). In other words, we aim to be tool-agnostic, which is an essential policy to have these days, given the quickly moving state of AI in general where a dozen new tools appear every week.

Motivating MLOps

To motivate the principles of MLOps, or rather to answer the question "Why bother with all of this extra work, just to put a model into production?", we are moving to the example jupyter notebooks, 01 and 02, to give a demonstration of how a typical endpoint for a data science project looks like, and why steps 1 to 7 (or at least to 6) are needed for the models to actually provide maximum value for a client or organization.

Supplementary Materials

Following is a list of optional supplementary readings to provide background and additional context for the lessons.

1. Data Science Connect Keynote, John Elder - The Twin Crises of Science and How to Defeat Them.

https://www.elderresearch.com/resource/videos/the-twin-crises-of-science-and-how-to-defeat-them

(Backup link in case the original disappears:

https://e.pcloud.link/publink/show?code=XZdg3KZ1QlaMytwvrREylDtNgWCLFxt8d8V)

Dr Elder covers the first crisis, "most experimental findings are false", where MLOps attempts to address the second one, "implementation is too rare" (only 20% of models make it into production). *It is critical to point out that the biggest challenge is not always technological or scientific, but getting stakeholder buy-in i.e. when going from notebook to production*.

Dr Elder mentions three key points to successful **real life** implementation:

- Address change management from the very start.
- (Form a close) team with stakeholders; gain trust with transparency; always seek their thriving.
- Deliver a tool that fits seamless into their environment (i.e. fix a pain point, do not create a new one).

An as a final point: *Success is impossible without implementation!*

Note: Elder Research has a YouTube channel and mailing list, both of which are highly recommended: https://www.elderresearch.com/insights.

2. Towards ML Engineering: A Brief History Of TensorFlow Extended (TFX).

https://arxiv.org/ftp/arxiv/papers/2010/2010.02013.pdf

An early paper by Google (2010), containing one of the early descriptions of an end to end machine learning platform, TensorFlow Extended, developed by Google for their in-house data science work, pioneering both the concept of such a platform, but also starting to codify principles essential to MLOps, including their know well known "Rules

Of Machine Learning", see also https://developers.google.com/machine-learning/guides/rules-of-ml for the actual rules.

3. Marvelous MLOps - MLOps Maturity Assessment.

https://marvelousmlops.substack.com/p/mlops-maturity-assessment

A blog post on the Marvelous MLOps Substack aiming to offer actionable steps to bring a data science project from being MLOps immature to 100% MLOps maturity. This is the general framework being used in this course.

4. MLOps Principles.

https://ml-ops.org/content/mlops-principles

A more "user-friendly" description of the general principles of MLOps, including some helpful diagrams, laying out how ML pipelines should be structured (adapted from Google, but made more digestible). This page covers the same principles as the below material, but in a more condensed format.

5. Practitioners guide to MLOps: A framework for continuous delivery and automation of machine learning.

https://services.google.com/fh/files/misc/practitioners_guide_to_mlops_whitepaper.pdf

For the more ambitious student, this whitepaper lays out in great detail, all the general principles and practises of MLOps (according to Google of course), at both an executive level and at the more technical level. After reading this paper, the student will generally be aware of all the important challenges and principles associated with the field of MLOps as described by an institution that is leading in the field.

References

[1] CDFoundation, "CD Foundation MLOps Roadmap 2024," [Online]. Available: https://github.com/cdfoundation/sig-mlops/blob/main/roadmap/2024/MLOpsRoadmap2024.md.

[2] Google, "Towards ML Engineering: A Brief History Of TensorFlow Extended (TFX)," [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2010/2010.02013.pdf.

[3] Microsoft, "Microsoft Azure MLOps Maturity Model," [Online]. Available: https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/mlops-maturity-model.

[4] Marvelous MLOps, "Marvelous MLOps Substack," [Online]. Available: https://marvelousmlops.substack.com.