Operationalization of Machine Learning models in (audit) innovation projects

Aleksei Maliutin MSc
is a consultant at KPMG Netherlands.

Jing Li
is a manager at KPMG Netherlands.

Aram Falticeanu RA MSc
is a senior manager at KPMG Netherlands.

Machine Learning (ML) is a powerful technique that has enormous potential in several domains, including audit. But it is difficult to bring ML into the production environment and iteratively improve the ML-powered products. In this article, we describe the background and the current state of ML, the difficulties of bringing ML-powered (audit) innovation projects into production, and the importance of Machine Learning Model Operationalization Management (MLOps) methodology. In addition, we discuss, as an effective use case, our own MLOps journey within our audit innovation department (Digital Assurance and innovation, Daní).

INTRODUCTION

Machine Learning (ML) has been getting more and more attention in modern economy. From the pure academic field of study in the past, ML has become the foundation of many billion-dollar companies such as Netflix (recommendation system), Uber (matching problem) and Prisma (computer vision), proving that Machine Learning has potential not only in academia but in a production environment as well. In the right hands, Machine Learning can help to solve many practical problems.

The rapid development in the field of Machine Learning has also provided new opportunities for innovation in the field of accounting and auditing. Auditors have access to vast amounts of data. They can use these to gather evidence to support their opinion more effectively. Machine Learning techniques are very powerful tools which the auditor can apply to reach his audit objectives ([Hoog19]). Digitalizing audits and audit innovation is an ongoing process. According to the survey *Deep Shift: Technology Tipping Points and Societal Impact*, which was conducted by the World Economic Forum in 2015, 75% of the 816 executives who participated believe that by 2025, 30% of corporate audit work will be completed by AI/ML ([Oppe21]). Machine Learning has a huge potential to utilize artificial intelligence to learn and provide insights based on auditing data. It is showing its potential in Ratio analysis, Regression analysis, financial statement analysis ([Boer19]). It is still nontrivial to bring Machine Learning-enabled audit innovation solutions into the production environment.

Most organizations greatly underestimate the complexity of productionalizing Machine Learning projects

We, Daní, are committed to the research and development of innovative products for auditing. We embrace innovation and cutting-edge technology, fitting them into the auditing domain and embedding data-driven technology broadly throughout the audit process. In the last few years, we delivered a variety of ML-powered auditing solutions to our clients. In the process of product development and implementation, we have encountered many troubles and realized that continuous iterative product development is very difficult for these ML solutions and products.

In this article, we will first introduce the complexity and difficulties to develop and implement Machine Learning products, the concept of *Machine Learning Model Operationalization Management* (MLOps) and, subsequently, use a business case to describe the evolution process of MLOps in our department.

KEY CHALLENGES OF MACHINE LEARNING IN PRODUCTION

Opportunities always come with challenges. Though with the vast array of open-sourced ML frameworks and learning materials, more and more professionals can experiment with ML and build ML models. These provide huge opportunities, although there are challenges to operationalizing ML for business purposes. Nevertheless 86% of businesses have increased their budgets for Machine Learning projects in 2021, according to the enterprise trends in Machine Learning report from Algorithmia ([Oppe21]), and "87% of organizations still struggle with long model deployment timelines and, at 64% of organizations, it takes a month or longer to deploy even a single model" ([Oppe21]).

From the business point of view, any project can only maximize its value in a production environment, and the Machine Learning project is not an exception.

Most organizations greatly underestimate the complexity of productionalizing Machine Learning projects ([Scul15]), or they treat Machine Learning projects in the same way as software development projects. Moving from experiments in academia, which is known as a "sandbox" environment, into the real world – production scale – is nontrivial.

Moreover, a ML project in academia and a ML project in production tend to have different requirements, see Table 1.

As a result, having the ML model source code is not enough to process millions of incoming requests or to answer to stakeholders' or clients' questions about model performance over time.

Table 1. Requirements for ML in academia and ML in production (adopted from [Huye22]).

	Research	Production
Objectives	Model performance	Different stakeholders have different objectives
Computational priority	Fast training, high throughput	Fast inference, low latency
Data	Static	Constantly shifting
Fairness	Good to have	Important
Interpretability	Good to have	Important

A Machine Learning project is also different from a traditional software project. The most important difference is that in Machine Learning project data is the core of the project, and the code is designed to service the data instead of application behavior. Machine Learning development is more iterative and explorative. Training a model is only a small part of the project, see Figure 1. For instance, according to Sculley et al. ([Scul15]): "only a tiny fraction of the code in many ML systems is actually devoted to learning or prediction". However, a lot of other components are still required to make that prediction available for the user or to deploy the ML model into a production system that generates business value. Building and initially deploying models is just the beginning of the project. Models in production must constantly be monitored, retrained, and redeployed in response to changing data signals in order to ensure maximum performance. All of this means that ML requires its own unique lifecycle ([Data22]).

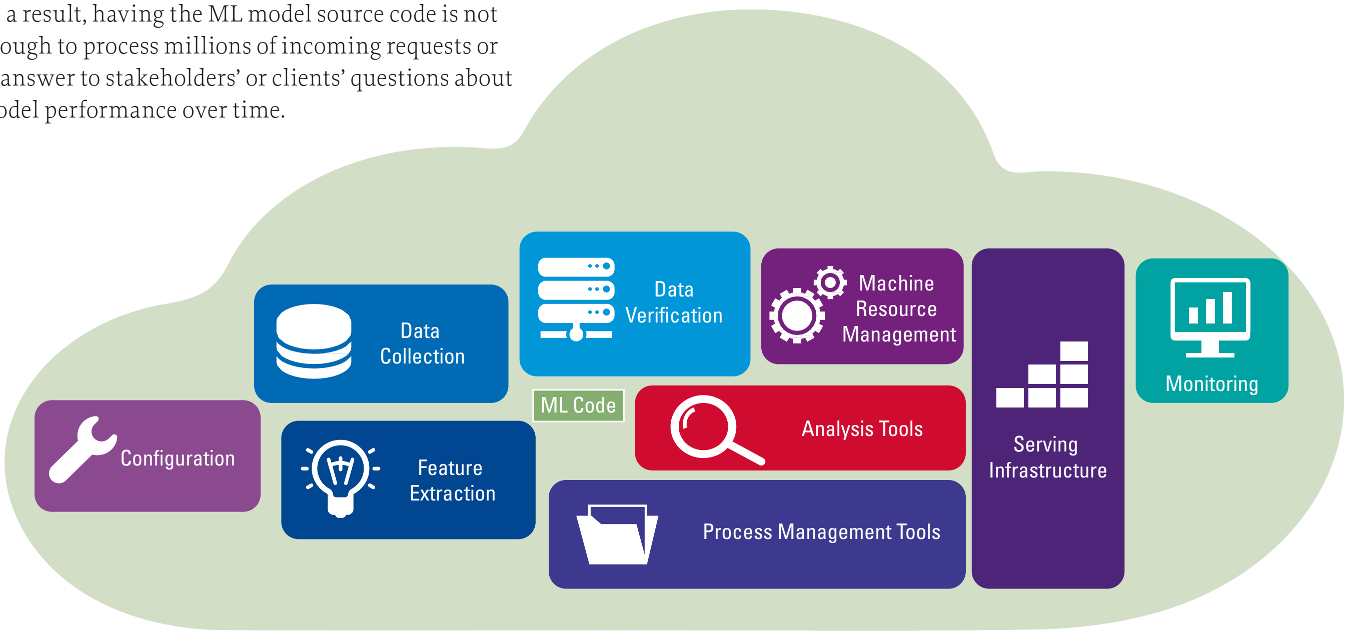


Figure 1. The ML code itself – e.g. model definition and training scripts, is a tiny component of the overall production ML system (adopted from [Scul15]).

INSIGHTS INTO MACHINE LEARNING LIFECYCLE

A Machine Learning project lifecycle has an iterative nature. The four major steps of the lifecycle include *Scoping, Data, Modeling, and Deployment* ([Deep21]). Figure 2 shows these steps and the common underlying activities. In the project scoping phase, we need to define objectives of the project and business fit, identify an opportunity to tangibly improve operations, increase customer satisfaction, or otherwise create value. After the business understanding, we enter the stage of data collection and collation, at which we gather the data, label it and transform it into the correct format as needed. With these data in hand, we can start the modeling phase, define the input and target output of the model, select and train the model, and perform an error analysis to validate the performance, and, optionally, explain the results. In the final phase, we deploy the model into the production environment, i.e. making it available to serve users' requests, and, subsequently, maintain and monitor the system in order to continue to leverage and improve the model through time.

WHAT IS MLOPS?

Within the Machine Learning projects, after a clear business understanding, we first need to process data, use the data to train and develop the Machine Learning model, and then set up the solution in the production environment and monitor the performance. Machine Learning Operations (MLOps) is a methodology designed to help with a successful ML delivery in production. It serves Machine Learning and data science assets as first-class citizens ([Cox20]). The main MLOps components include data management, training of a model, versioning, monitoring, experience tracking etc. ([Vise22]). MLOps methodology covers the full ML project lifecycle end to end in the iterative approach, providing the connection between all the components required for success in production ML.

The complete MLOps iterative process includes three broad phases, usually named "Designing the ML-powered application", "ML Experimentation and Development", and "ML Operations", which are usually required for success in production ML ([Vise22]) based on the definition from ML-ops.org. According to [Vise22], MLOps can be seen as an "Iterative-Incremental process" and usually contains three high-level stages, see Figure 3: ML-featured application "Design, Model Development and Operations".

To get a non-biased evaluation of the existing organization's MLOps environment, Microsoft ([Micr22]) and Google ([Goog22]) define the maturity models, which help clarify the MLOps principles and practices in accordance with the explicit requirements and characteristics per each level (see Table 2). The maturity model can be seen as a metric of maturity of one's Machine Learning production environment as well as the processes associated with it ([Micr22]); it also reflects the velocity of training new models given new data or training new models given new implementations ([Goog22]). For example, the MLOps level 0 process ([Goog22]) contains a list of disconnected manual steps, which leads to difficulties with releases and quick model degradation in the real world. In contrast, MLOps level 2 (level 4 according to Microsoft classification) is the most advanced process, which includes continuous pipeline integration, automated model training and testing, and automatic model releases.

Figure 2. Machine Learning project life cycle (adopted from [Deep21]).

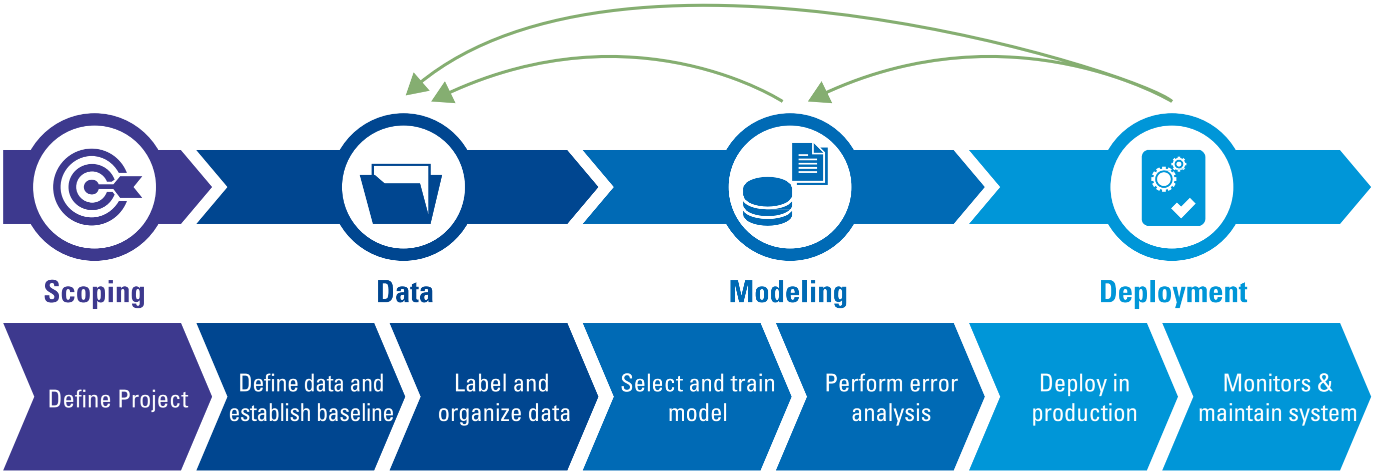


Figure 3. An overview of MLOps Iterative-Incremental process (adopted from [Vise22]).

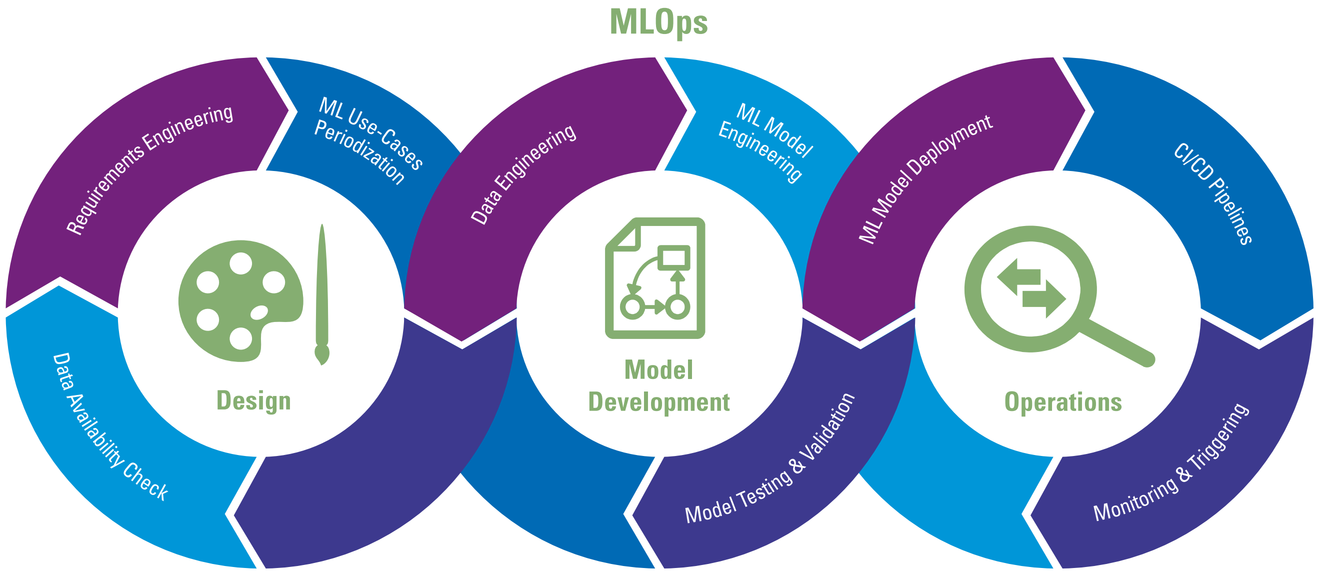


Table 2. MLOps maturity level overview (based on [Micr22] and [Goog22]).

	level		level	
Microsoft	0	No MLOps	0	**Google**
	1	DevOps, no MLOps		
	2	Automated training	1	
	3	Automated model deployment		
	4	Full MLOps Automated Operations	2	

HOW CAN MLOPS HELP WITH INNOVATION?

Machine Learning can make a significant difference in several domains because it makes it possible to learn from data and initiate for example next steps in an automated way. In the remaining part of this article, we focus on audit as a use case to make this more tangible. However, keep in mind that this could also apply to your industry.

Machine Learning has the potential to make audit procedures more efficient and effective but due to the uniqueness of the audit business itself, new Machine Learning auditing procedures require rigorous validation, explanation, and regulation. In other words, running a Machine Learning project or service in the audit domain has its own specific requirements.

Although one should consider MLOps principles equally important in general use cases, we can highlight a few crucial for the audit domain. First of all, corporate and government regulations can require a complete provenance of the model deployed in production. For example, it includes but is not limited to information, such as how the model was built, what data was used, and what parameters and configurations were used, which falls under the functionality of the *Versioning* concept of MLOps. Model *Monitoring* is another important topic to pay attention to. For all models deployed in production, i.e. accessible for use, one must be sure that they perform as expected. Moreover, all deviations from the original expectation should be quickly detected. Unfortunately, “the one constant in the world is change” ([Deep21]). Therefore, one also needs to be aware of model degradation and be able to react quickly to deliver the original model prediction quality over a wide time frame. Therefore, by real-time monitoring of incoming requests and model inferences, we can assure that required adjustments are applied in time.

Within the audit domain we have strict regulation requirements for Machine Learning-powered auditing services. We should detect and monitor the risk of ML/AI based on its *Quality*, *Integrity*, *Explainability*, *and Agility*. We can combine the four steps of the lifecycle, *Scoping*, *Data*, *Modeling*, *and Deployment*, with the requirements of *Quality*, *Integrity*, *Explainability*, *and Agility*. For example, we need to answer the following simplified key questions in the different phases.

Scoping phase:

1. What does the model do?
2. Is the model compliant with the compliance requirements in the specific (audit) domain?

Modeling phase:

3. Does the model do what it needs to do?

Deployment phase:

4. Do the models keep doing what they need to do?

As MLOps covers the full ML project lifecycle end to end and helps achieve project goals with iterative approach. For all these questions, we can get the answers with the help of MLOps in a systematic and sustainable way. The MLOps can therefore help with the efficient delivery of the AI/ML solutions in the audit innovation domain, and perform procedures under control, increase transparency in all ML-based projects and gain the trust of our clients.

> MLOps can help with the efficient delivery of the AI/ ML solutions in the audit innovation domain

WHAT IS THE MLOPS JOURNEY WITHIN DANÍ?

Academic discussions are great, but putting theoretical knowledge into practice in the business field seems even more important, doesn't it? We work at the Digital Assurance and innovation department (Daní) at KPMG NL, which is the audit innovation department. We are developing ML/AI powered data-driven solutions and tools to innovate and digitalize auditors' daily work. This is beneficial to our professionals and end-clients as it makes the audit more effective and efficient. In addition, we unlock new insights that enrich the audit service. In this section, we share the story of integrating MLOps concepts into our way of working. As we have mentioned, modern Machine Learning open-source frameworks can help us build models with a few lines of code. Three years ago, we started the Machine Learning experiments with the local Jupyter Notebook training model on our laptop, using Excel sheets to keep and track the results of our experiments, and deploying a standalone web service for each model to be used in production. With this workflow, we met some problems that were hard to tackle during the project development. First of all, data processing and model training had been limited by the power of our laptops, resulting in the fact that iterative process of model improvement could take days and even months. Secondly, it was hard to keep in sync experiment results between team members and traced back the saved model. Finally, making the saved model available for the end user had included a lot of non-ML codes such as a web application code or cloud configuration, and it was hard to maintain. An attentive reader can conclude this was a zero-maturity level for MLOps.

> Jupyter Notebook: a popular local development and debugging tool for Machine Learning experience.
> TFX: a TFX pipeline is a sequence of components that implement an ML pipeline that is specifically designed for scalable, high-performance Machine Learning tasks.
> Kubernetes: an open-source container orchestration system for automating software deployment, scaling, and management.

To improve our workflow, decrease time to market and avoid a lot of extra code we had for each ML-powered solution, our team decided to make a move towards MLOps methodology. Our MLOps journey started at the beginning of 2021. At that time, we started to work on the "sesame" solution – the NLP (Natural Language Processing) tool to process financial reports and highlight the sentiments (negative, positive, or neutral) of such reports. We were focusing on the *Automation* and *Repro-*

ducibility principles of MLOps with the development of the project. It was our first project where we delivered an end-to-end Machine Learning pipeline, starting from data ingestion, processing incoming data, training the Machine Learning model and evaluating it afterward based on our evaluation requirements, and delivering the trained model to the specified location. Our technical stack included: TensorFlow Extended (TFX) – "an end-to-end platform for deploying production ML pipelines" ([Tens22]), and Kubeflow – "the Machine Learning Toolkit for Kubernetes" ([Kube22]). With TFX we were building our ML pipeline, while Kubeflow was used as an orchestration tool, helping to use cloud recourses efficiently.

The sesame project was challenging for us, since the TFX framework was still under active development (the public available release is in alpha stage), many underlaying components could have significant changes in the future and documentation was quite limited. Another challenge for us was that TFX was initially developed within Google and mainly focused on the Google Cloud Platform while we were utilizing Azure Cloud. For example, since the start of our work, the data ingestion process has changed, and simpler out-of-the-box options and formats have been added. The lack of complete documentation led us to multiple trials and attempts to build a robust and clear *Train component* logic. Fortunately, after that non-trivial but exciting work, the TFX pipeline for "sesame" project came to a success: However, we have not covered all MLOps principles in the sesame project. For example, an initial data exploration was done with local Jupyter Notebooks, model deployment contained a few manual steps

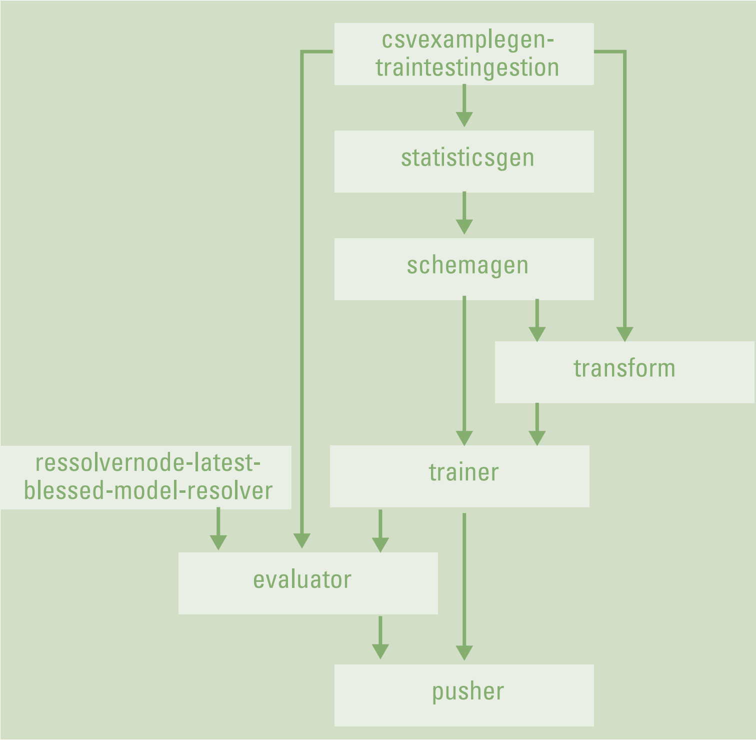
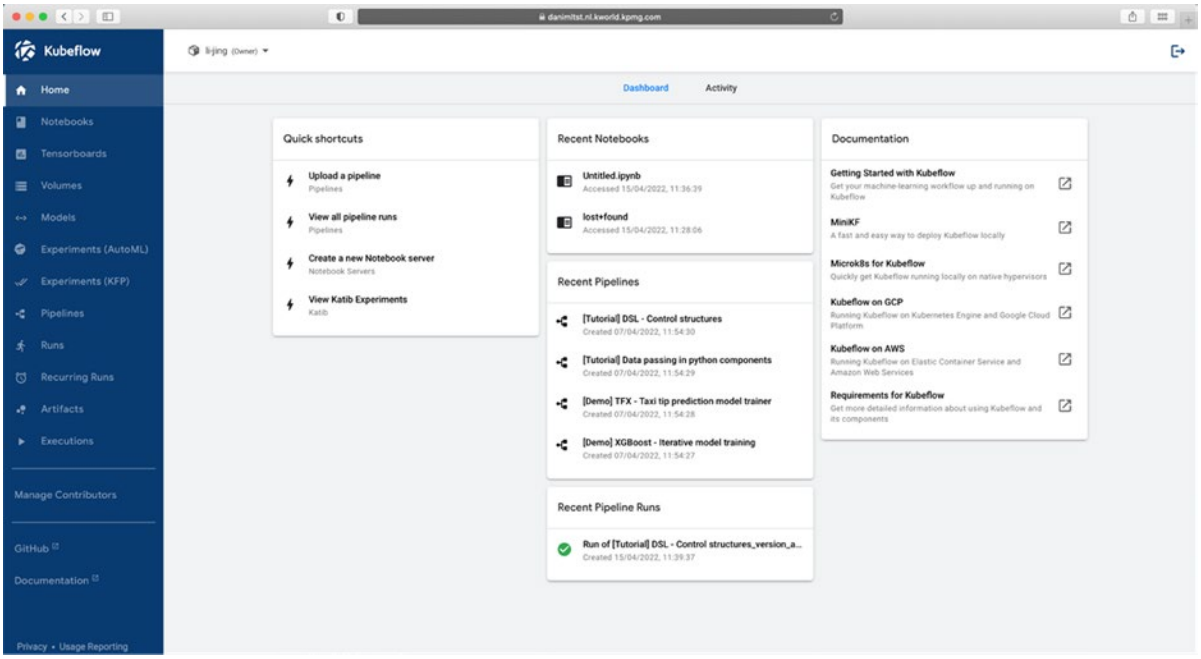


Figure 4. Sesame end-to-end pipeline, visualization from Kubeflow Dashboard.

Figure 5. Kubeflow Central Dashboard, main page.



Quick shortcuts

Upload a pipeline
Pipelines

View all pipeline runs
Pipelines

Create a new Notebook server
Notebook Servers

View Katib Experiments
Katib

Recent Notebooks

Untitled.ipynb
Accessed 15/04/2022, 11:36:39

lost+found
Accessed 15/04/2022, 11:28:06

Recent Pipelines

[Tutorial] DSL - Control structures
Created 07/04/2022, 11:54:30

[Tutorial] Data passing in python components
Created 07/04/2022, 11:54:29

[Demo] TFX - Taxi tip prediction model trainer
Created 07/04/2022, 11:54:28

[Demo] XGBoost - Iterative model training
Created 07/04/2022, 11:54:27

Recent Pipeline Runs

✓ Run of [Tutorial] DSL - Control structures_version_a...
Created 15/04/2022, 11:39:37

Documentation

Getting Started with Kubeflow
Get your machine-learning workflow up and running on Kubeflow

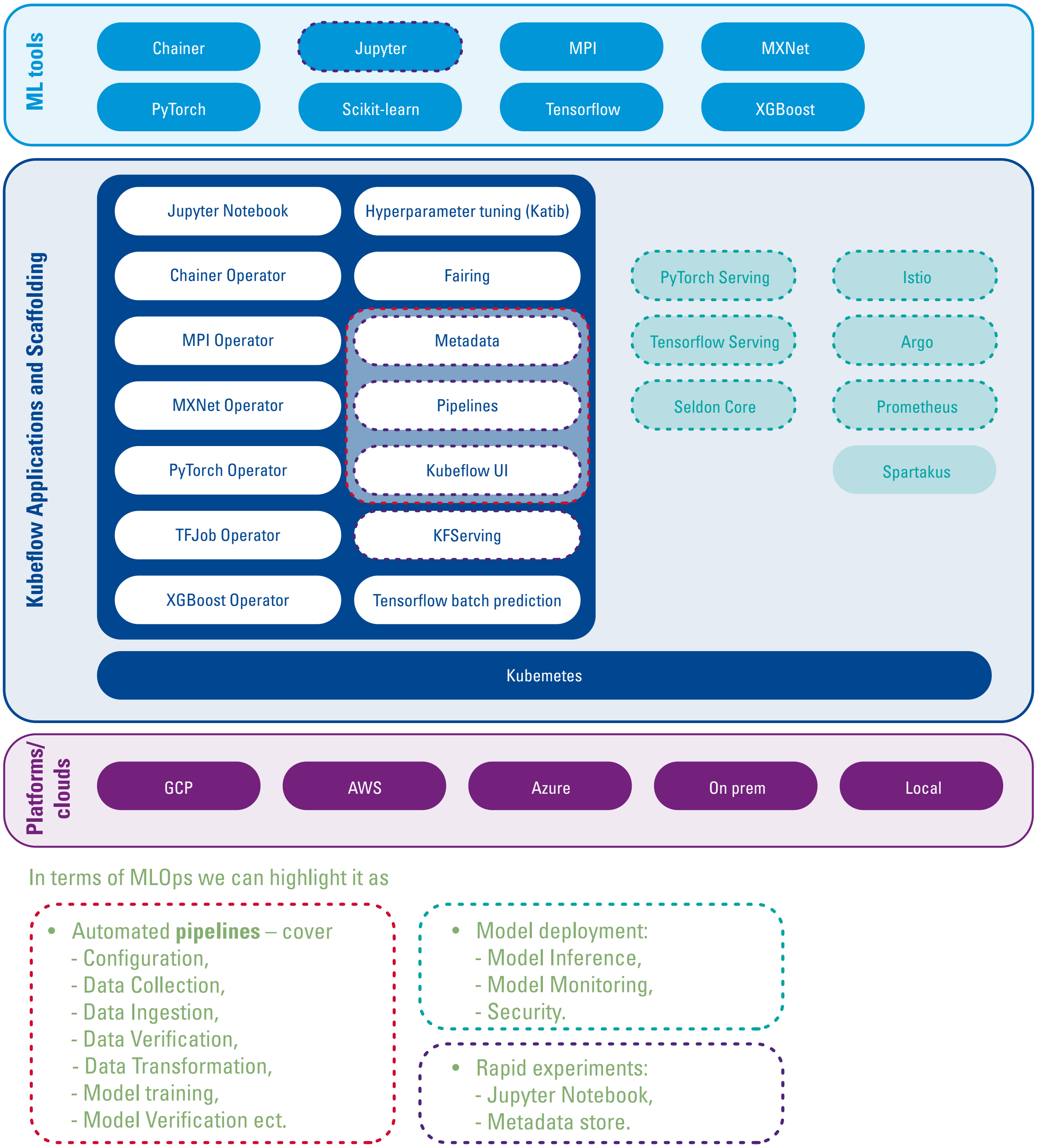
MiniKF
A fast and easy way to deploy Kubeflow locally

Microk8s for Kubeflow
Quickly get Kubeflow running locally on native hypervisors

Kubeflow on GCP
Running Kubeflow on Kubernetes Engine and Google Cloud Platform

Kubeflow on AWS
Running Kubeflow on Elastic Container Service and Amazon Web Services

Requirements for Kubeflow
Get more detailed information about using Kubeflow and its components



In terms of MLOps we can highlight it as

- Automated **pipelines** – cover
  - Configuration,
  - Data Collection,
  - Data Ingestion,
  - Data Verification,
  - Data Transformation,
  - Model training,
  - Model Verification ect.

- Model deployment:
  - Model Inference,
  - Model Monitoring,
  - Security.

- Rapid experiments:
  - Jupyter Notebook,
  - Metadata store.

Figure 6. Main components and their functions in Kubeflow (adopted from [Kube22]).

such as moving model to the cloud storage location and creating model deployment. That is why we have not stopped and still continue on our MLOps integration journey.

Within Dani we already noticed a positive difference by using Kubeflow. For example, a few years ago, we spent four days on delivering a single model trained inside Jupyter Notebook into production environment. One day was usually spent on wrapping that model with cloud-related configuration, another day was spent on alignment with the application development team about model communication protocol, and finally the third day was spent on the actual deployment. So, for a single instance of modeling it took four days on average. And in case of tiny changes in the input data, model architecture or specific client requirements, that process had to be repeated. However, following MLOps principles and utilizing the power of Kubeflow can decrease that delivery time from days to a few hours.

Up to now, we have decreased the time for the model's iteration step – to try new model parameters or new input data and deliver an updated model in the production environment – in our ML-powered solutions from days to hours, adding a lot of business value, since the time to market of validated ML-powered solutions is decreased significantly.

In the future, with the help of Kubeflow, within Daní we will be able to set up hundreds of rigorous validation experiments for Machine Learning auditing procedures simultaneously, which can help us speed up the model training time and shorten the time to market. As described in the MLOps maturity models, the higher the maturity level, the more components in the Machine Learning lifecycle are automated, manageable, and traceable.

By increasing the level of automation of our MLOps platform, we can get close to the real-time model training based on newly added data, smooth and robust model deployment afterwards. Finally, production grade monitoring as well as governance of models used by our clients add trust to Daní's ML-powered solutions. In Table 3, we explicitly highlight the important properties of ML projects and added value for the Daní team and our users. These insights can help you to see the benefits as well and determine your own business case in other domains.

> We have decreased the time for the model's iteration step in our ML-powered solutions from days to hours

Table 3. Added value of using MLOps at Daní team.

	In the past	Now	In the future	Added value for Daní
Time to production (per model)	~4 days	~4 hours	< 1 hour	Users get quicker access to a more efficient model. As a team, we have a shortened feedback loop from our users .
Experimental setup	Local laptop. No centralized experiment tracking.	Unified cloud-based environment with experiment tracking.		Clearly record all experiments we conduct and see their results.
Resources for training	Only local laptop.	In the cloud.	In the cloud with cost optimization.	Faster training enables trying more model(s)/ parameters, probability to find the optimal ones are higher.
Model governance and lineage	Only parameters used for training.	Parameters, artifact versioning, meta information about data used for training.	Versioning the code, data, and ML model artifacts.	Transparent history of model adds trust to Daní's ML-powered solutions.
Model monitoring	–	System usage metrics, performance metrics.	Detect degradation of the predictive quality of the ML mode.l	We will know when actions are required to keep the high model quality for the end users.

CONCLUSION

The potential of Machine Learning projects is enormous, and these projects are an integral part of audit innovation. However, as a cutting-edge technology, it is still difficult to implement Machine Learning projects in a production environment. Hopefully, with MLOps, we can accelerate the process, shorten the time to market, and realize the vision of efficient ML project management, while ensuring performance, quality and trust.

Acknowledgement

We want to express our appreciation to Marcel Boersma for all the help and support in writing of this article.

References

[Boer19] Boersma, M. (2019). Network theory in audit. *Compact* 2019/4. Retrieved from: https://www.compact.nl/en/articles/network-theory-in-audit/

[Data22] DataRobot (2022). Machine Learning Life Cycle. Retrieved from: https://www.datarobot.com/wiki/machine-learning-life-cycle/#:~:text=The%20machine%20learning%20life%20cycle%20is%20the%20cyclical%20process%20that,to%20derive%20practical%20business%20value.

[Deep21] DeepLearning.AI (2021, October). Steps of an ML project. Retrieved from: https://community.deeplearning.ai/uploads/short-url/ljaUfNUPCIt81AGwsQKiL6GWCcZ.pdf

[Cox20] Cox, T., Neale, M., Marck, K., Hellström, L., & Rimoto, A. (2020, September 25). MLOps Roadmap 2020. *Github.* Retrieved from: https://github.com/cdfoundation/sig-mlops/blob/master/roadmap/2020/MLOpsRoadmap2020.md

[Goog22] Google (2022). MLOps: Continuous delivery and automation pipelines in machine learning. Retrieved from: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

[Hoog19] Hoogduin, L. (2019). Using machine learning in a financial statement audit. *Compact* 2019/4. Retrieved from: https://www.compact.nl/en/articles/using-machine-learning-in-a-financial-statement-audit

[Huye22] Huyen, C. (2022). *Designing Machine Learning Systems.* O'Reilly. Retrieved from: https://www.oreilly.com/library/view/designing-machine-learning/9781098107956/

[Kube22] Kubeflow (2022). Kubeflow: The Machine Learning Toolkit for Kubernetes. Retrieved from: https://www.kubeflow.org/

[Micr22] Microsoft (2022). Machine Learning operations maturity model. Retrieved from: https://docs.microsoft.com/en-us/azure/architecture/example-scenario/mlops/mlops-maturity-model

[Oppe21] Oppenheimer, D. (2021). *2021 enterprise trends in machine learning.* Algorithmia. Retrieved from: https://info.algorithmia.com/hubfs/2020/Reports/2021-Trends-in-ML/Algorithmia_2021_enterprise_ML_trends.pdf?hsLang=en-us

[Scul15] Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., … & Dennison, D. (2015). Hidden technical debt in machine learning systems. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama & R. Garnett (eds.), *Advances in Neural Information Processing Systems 28.*

[Tens22] TensorFlow (2022). TensorFlow Extended (TFX) is an end-to-end platform for deploying production ML pipelines. Retrieved from: https://www.tensorflow.org/tfx

[Vise22] Visengeriyeva, L., Kammer, A., Bär, I., Kniesz, A., & Plöd, M. (2022). MLOps Principles. *Ml-Ops.Org.* Retrieved from: https://ml-ops.org/content/mlops-principles

About the authors

Aleksei Maliutin is a consultant at KPMG, mainly focusing on ML Engineering and Data Science. In 2021, he took one of the leading roles in delivering the Data Mass solution used by Dani, and now he is the technical leader of MLOps infrastructure in Dani team.

Jing Li is a manager at KPMG. In her current role as a tech lead and product and technology management team member within the KPMG Dani department, she focuses on closely working together with auditors to create audit solutions and innovations.

Aram Falticeanu is a senior manager at KPMG. In his current role as a senior manager and management team member within the KPMG Dani department, he focuses on leading the creation of data propositions and the development of solutions to make the audit more effective and efficient.