



HAL
open science

Applying DevOps to Machine Learning: ROCKFlows, a Story from the Trenches

Mireille Blay-Fornarino, Günther Jungbluth, Sébastien Mosser

► **To cite this version:**

Mireille Blay-Fornarino, Günther Jungbluth, Sébastien Mosser. Applying DevOps to Machine Learning: ROCKFlows, a Story from the Trenches. 2018. hal-01792775

HAL Id: hal-01792775

<https://hal.science/hal-01792775>

Submitted on 15 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Applying DevOps to Machine Learning

ROCKFlows, a Story from the Trenches

Mireille Blay-Fornarino¹, Günther Jungbluth¹, and Sébastien Mosser¹

Université Côte d'Azur, CNRS, I3S, France
{blay,jungblunth,mosser}@i3s.unice.fr

The *Machine Learning* (ML) community is currently blooming with hundreds of new algorithms to implement tasks such as data classification for example [1]. To support data scientists and engineers who have to chose among all these algorithms, we are defining the ROCKFLOWS platform [2, 3] to automatically create a *software product line* of workflows integrating such algorithms. In a nutshell, the idea of ROCKFLOWS is the following: a software developer defines a new algorithm, and uploads it to the platform. The platform confronts it with reference datasets used as benchmarks, and the newly defined algorithm is classified among criteria such as data type, accuracy or execution time. It eventually enriches an algorithm portfolio used by data scientists to configure an ML workflow (*e.g.*, preprocessing, algorithm, postprocessing) that fit their needs *w.r.t* their business objective.

The ML community: Who are the Devs? Who are the Ops? In this context, one might wonder how ROCKFLOWS fits the DevOps paradigm. According to Wikipedia, DevOps “*strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevOps aims at shorter development cycles, increased deployment frequency, more dependable releases, in close alignment with business objectives*” [4]. On the one hand, we call a *developer* the engineer who is in charge of implementing a learning algorithm. It is her responsibility to properly implement and test such an algorithm, at the unit and integration levels (*e.g.*, integration with standard preprocessors to prepare the dataset to classify). The deployment can be done through the integration of a standard library such as Weka [5], or providing a container hosting the algorithm and its dependencies as a self-contained tool. On the other hand, we call an *operational* the engineer who reuses an existing algorithm to solve a particular problem that requires learning capabilities. It is her responsibility to deploy the algorithm in her very context and maintain it *w.r.t* the operational infrastructure she uses.

Bridging the gap between Dev and Ops in the ML community. Based on these definitions, we defend ROCKFLOWS as a DevOps platform for ML algorithms. It provides an experimentation pipeline to support the addition of a new algorithm into a portfolio. From the *dev* point of view, ROCKFLOWS aims to reduce development time and support frequent release of ML algorithms. Using the platform, each new algorithm is verified and automatically estimated *w.r.t* reference benchmarks without any intervention from the developers. The

platform also automatically conducts reproducible experiments to gather metrics associated to the benchmarks, which is essential for workflow comparisons. From the *ops* point of view, ROCKFLOWS supports the definition of dependable releases by helping data scientists to select the right workflow based on the previously described metrics expressed according to business objectives, *e.g.*, “a fast classifier with medium accuracy compatible with CSV data”. It also accelerates the deployment time by automatically generating a turn-key workflow ready to analyze datasets.

Technical implementation. In order to compare different workflows we need to execute these workflows in the same execution context and gather metrics that can be compared. The platform uses *Docker* to create and manage automated, controlled, execution contexts. Every job description inherits from a kernel image that includes tools to gather metrics. By automating metric surveys, equivalence, independence and completeness of the evaluation, processes are ensured. The platform contains an idempotent job scheduler that orchestrates the execution of the jobs of workflows comparison. This property reduces the overall complexity. When a new data set, transformation, algorithm is added to the experiment platform we automatically compute the jobs to be executed and add them to the job scheduler. It accelerates reliable cross-testing of workflows and data-sets. For portfolio construction, we rely on standard feature model merging.

Perspectives. A portfolio of ML Workflows relies not only on automated experiments and deployments but also on evolving processes to compare, evaluate and address problems. So *ops* and *dev* need to constantly communicate to improve the quality of the release process and support changes and enrichments. We foresee the possibility of using specific languages to capture the relations among algorithms, preconditions, experimental reports, visualize the results of experiments.

References

1. Delgado, M.F., Cernadas, E., Barro, S., Amorim, D.G.: Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research* **15**(1) (2014) 3133–3181
2. Camillieri, C., Parisi, L., Blay-Fornarino, M., Precioso, F., Riveill, M., Vaz, J.C.: Towards a software product line for machine learning workflows: Focus on supporting evolution. In Mayerhofer, T., Pierantonio, A., Schätz, B., Tamzalit, D., eds.: *Proceedings of the 10th Workshop on Models and Evolution co-located with ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS 2016)*, Saint-Malo, France, October 2, 2016. Volume 1706 of *CEUR Workshop Proceedings.*, CEUR-WS.org (2016) 65–70
3. I3S: The ROCKFlows platform. <http://rockflows.i3s.unice.fr> (2017)
4. Wikipedia: DevOps Entry. <https://en.wikipedia.org/wiki/DevOps> (2017)
5. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* **11**(1) (November 2009) 10–18