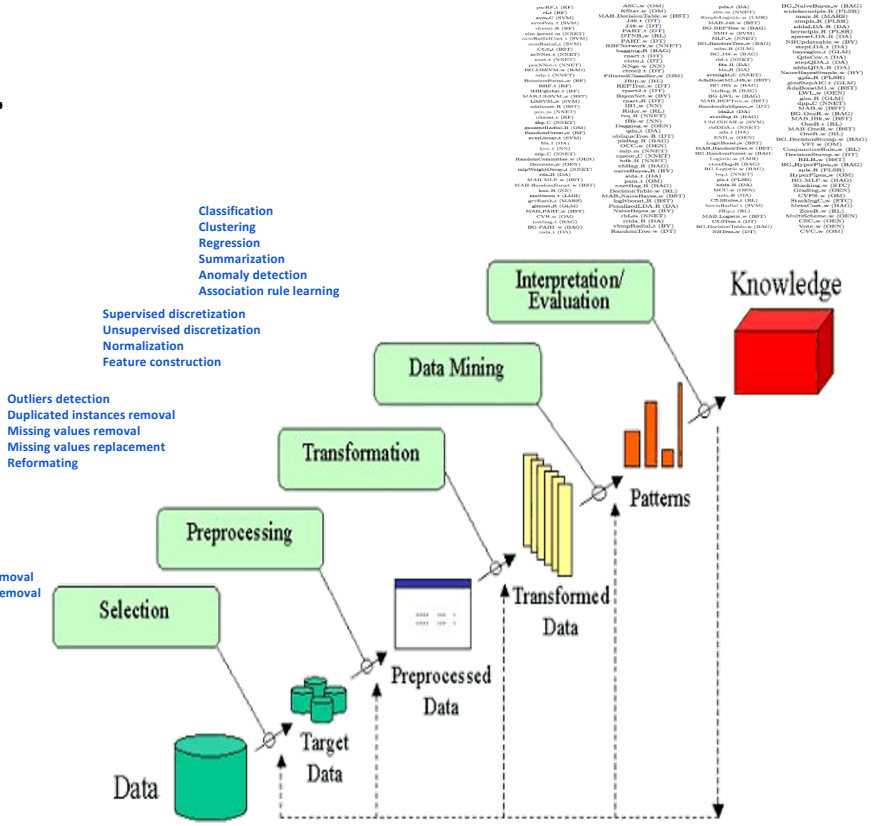
Feedback on the development of a software system for automatic online selection of ML pipelines

Mireille Blay-Fornarino & Frédéric Precioso
Côte d'Azur University
I3S,
France

Towards automatic online selection of ML pipelines

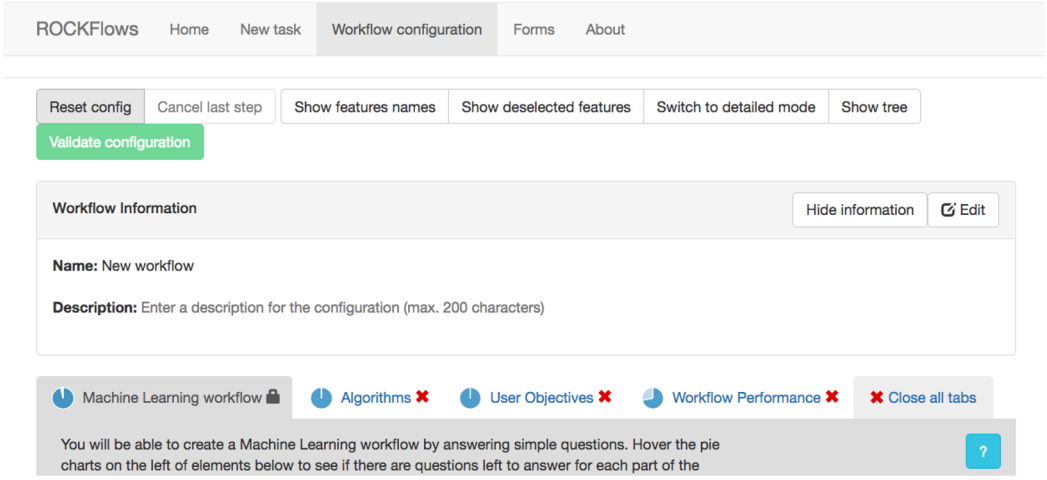
2015: Defining your ML pipeline requires the involvement of highly skilled data scientists*.

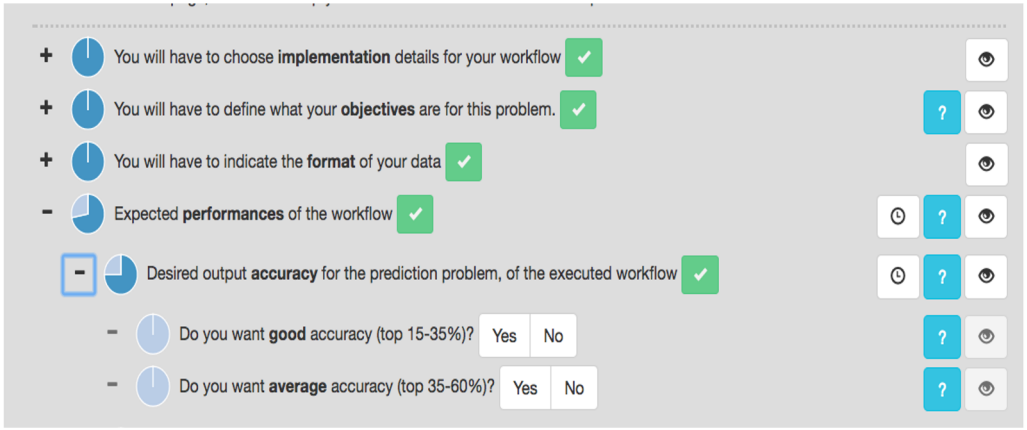
Working assumption: ML pipelines are developed from a common set of core assets in a prescribed manner.



*Postulate: Deep learning or your favorite ML tool can't solve every problem.

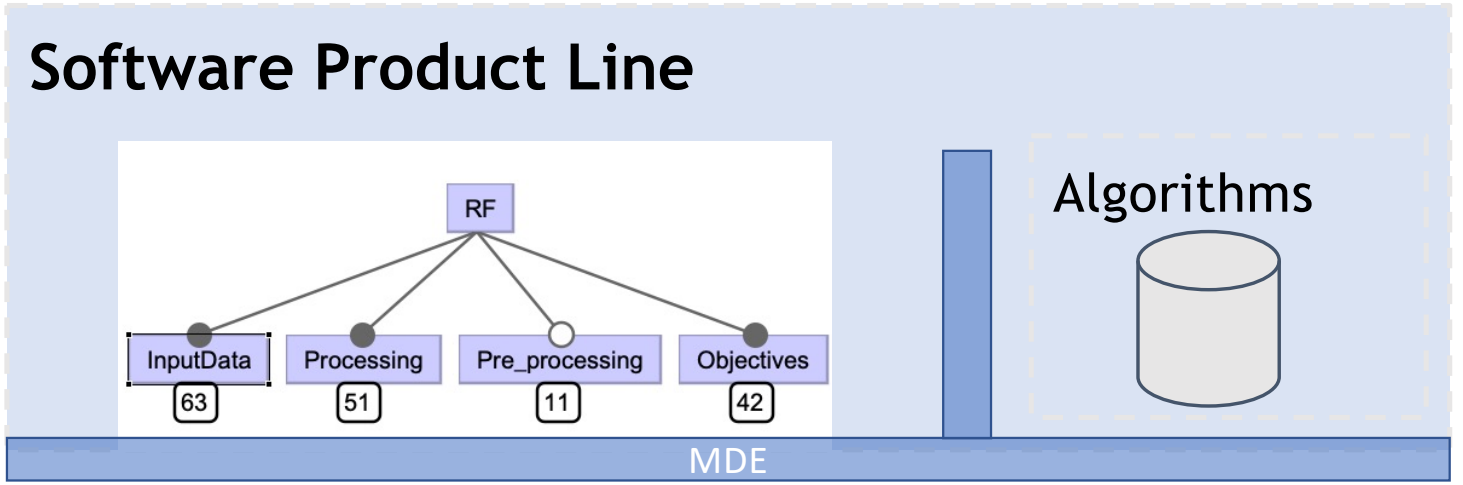
4 years to build ROCKFLOWS, a SPL to produce suitable ML Pipelines

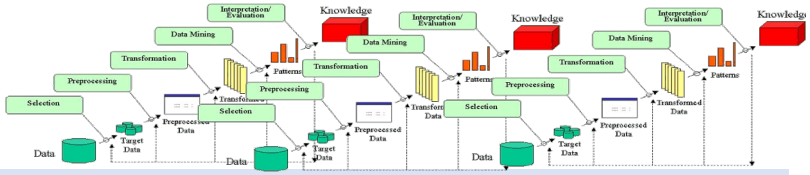




ROCKFlows' simplified architecture

Here is my problem





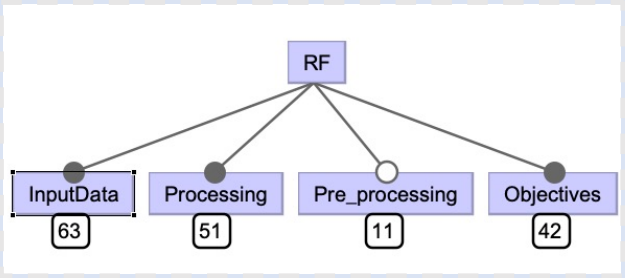
The line scope is "classification », the assets are the algorithms, the products are pipelines.

ROCKFlows: MDE to handle some of the ML variability evolution

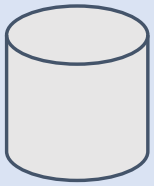
Here is my problem

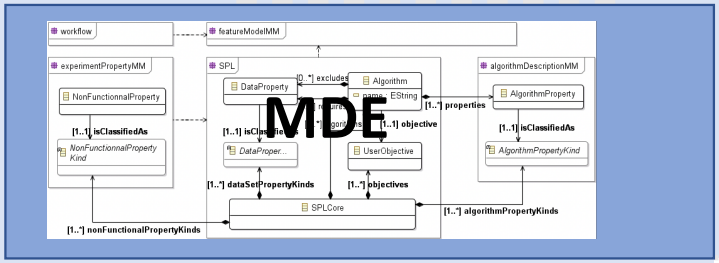
Graphical User Interface

Software Product Line



Algorithms

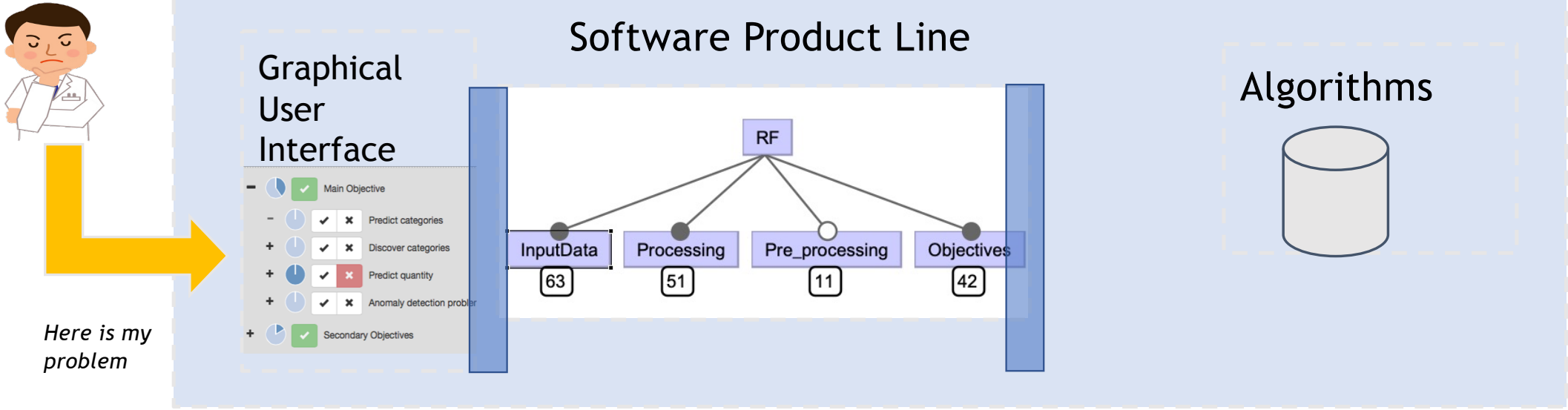




Model-Driven Engineering combined with separation of concerns has been used to manage some of the evolving requirements [3], for example, the integration of new algorithms [1] or new criteria [2].

[1] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?," *J. Mach. Learn. Res.*, vol. 15, 2014,
[2] B. Bilalli, A. Abelló, and T. Aluja-Banet, "On the predictive power of meta-features in OpenML," Int. J. Appl. Math. Comput. Sci., vol. 27, no. 4, 2017.
[3] C. Camillieri, et al. "Towards a Software Product Line for Machine Learning Workflows: Focus on supporting evolution," in *Models and Evolution 2016*, vol. 1706, pp. 65–70,

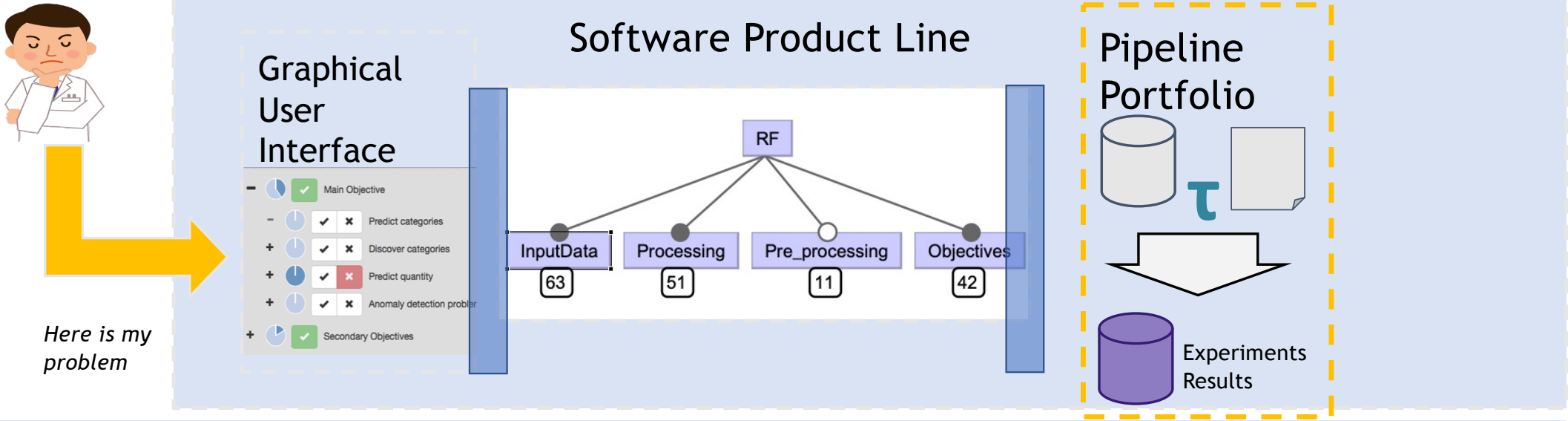
« what is the best pipeline for my problem » has no easy answer



No-Free-Lunch Theorem[1] states that the best algorithm will not be the same for each dataset.

[1] Wolpert, David (1996), The Lack of A Priori Distinctions between Learning Algorithms, Neural Computation, pp. 1341-1390.

ROCKFlows: Meta-learning to compare pipelines



To predict the accuracy / total time/ memory usage that a pipeline will have on users datasets, without doing the evaluation phase, we need meta-learning and a porfolio [1] of pipelines.

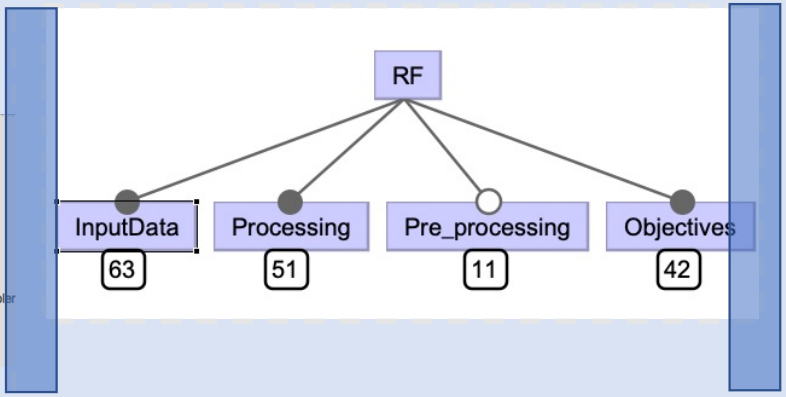
[1] Lars Kotthoff, Algorithm Selection for Combinatorial Search Problems: A Survey, AI 2014

ROCKFlows: A platform for experimentation to tame evolution

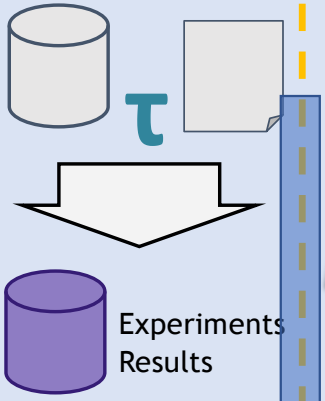
Here is my problem

Graphical User Interface

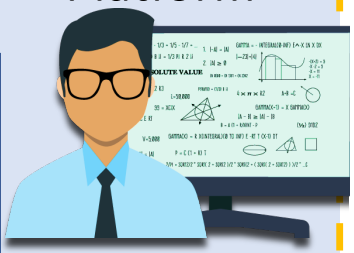
Software Product Line



Pipeline Portfolio



Experiment Platform



To compare experiments, we need to control the execution environment [1] [2].

[1] B. Benni, M. Blay Fornarino, S. Mosser, F. Precisio, and G. Jungbluth, "When DevOps meets meta-learning: A portfolio to rule them all," in *Proceedings - 2019 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion, MODELS-C 2019*, Sep. 2019, pp. 605–612, doi: 10.1109/MODELS-C.2019.00092.

[2] C. Duffau, C. Camillieri, and M. Blay-Fornarino, "Improving confidence in experimental systems through automated construction of argumentation diagrams," in of the 19th International Conference on Enterprise Information Systems, 2017, vol. 2.

In 2020, ROCKFlows is “operational” … Retrospective and Perspectives

- 9 unit preprocessors
- 68 algorithms
- 100 data sets
- 25,038,720 total combinaisons
- 5,244,948 valid workflows
- 3,526,516 valid workflows for the Iris data set

Lesson 1: In Continuous Development Context matters

The shift to time series analysis in an industrial context has given us a different perspective on the practices and analysis of the problem[2] : generalizability [4] and abstraction debt [1].

In a context where the purpose is research, and the team is small, the multiplicity of technologies makes code evolution very difficult.

Towards context-driven simplification:
Updates can no longer be done continuously.

[1] D. Sculley et al., “Hidden Technical Debt in Machine Learning Systems,” Adv. Neural Inf. Process. Syst., vol. 28, 2015.
[2] V. Basili, L. Briand, D. Bianculli, S. Nejati, F. Pastore, and M. Sabetzadeh, “Software Engineering Research and Industry: A Symbiotic Relationship to Foster Impact,” IEEE Softw., vol. 35, no. 5, pp. 44–49, Sep. 2018, doi: 10.1109/MS.2018.290110216.
[3] https://www.joelonsoftware.com/2002/11/11/the-law-of-leaky-abstractions/
[4] L. Briand, D. Bianculli, S. Nejati, F. Pastore, and M. Sabetzadeh, “The Case for Context-Driven Software Engineering Research: Generalizability Is Overrated,” *IEEE Softw.*, vol. 34, no. 5, pp. 72–75, 2017, doi: 10.1109/MS.2017.3571562.

Lesson 2: Continuous Learning*

Don't leave the problem specification task in the hands of the data scientist[2]

Expecting the machine to learn (sometimes even what we know) is possible.
Doing so without care is destructive and dangerous.

How many resources does it take to learn? When will we have to relearn? At what price? What are the limits of what we have learned[1] ? Even with the resources of google, it is not reasonable not to ask these questions.

A compromise has to be found between the added value of machine-learning, its requirements and the acquisition of new knowledge for Human.
Our community knows it: Meta-learning is not necessarily Machine-learning.

*Continuous ~~integration~~ learning is the practice of automating the integration of data changes from multiple contributors into a single software project.

[1] H. Degroote, B. Bischl, L. Kotthoff, and P. De Causmaecker, "Reinforcement Learning for Automatic Online Algorithm Selection - an Empirical Study," in *ITAT 2016 Proceedings, CEUR Workshop Proceedings Vol. 1649*, 2016, vol. 1649, pp. 93–101, [Online]. Available: http://ceur-ws.org/Vol-1649/93.pdf.

[2] G. A. Lewis, S. Bellomo, and I. Ozkaya, "Characterizing and Detecting Mismatch in Machine-Learning-Enabled Systems," Mar. 2021, Accessed: Jul. 19, 2021. [Online]. Available: https://arxiv.org/abs/2103.14101v1.

Lesson 3: Continuous Feedback:

Integrate Humans in the loop, not only their traces !

We thought of the feedback loop as the evolution of the portfolio in an automatic process; feedback from experiments may or may not reinforce the quality of our predictions; connexion to other systems like OpenML[1].

Our challenge today is to effectively integrate data scientists' input into the feedback loop, not just the results of the experiments.

[1] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "OpenML: Networked Science in Machine Learning," *SIGKDD Explor.*, vol. 15, no. 2, pp. 49–60, 2013, doi: 10.1145/2641190.2641198.

Pending questions

1- Meta-modeling and continuous development: what place for context-driven research?

2- Machine Learning and continuous and sustainable integration: what compromises, what requirements?

3- How to integrate humans into the loop, not just their traces ?