Data Sciences with and for Data Scientists

Yann Brault, Mireille Blay Fornarino, Florent Jaillet, Yassine El Amraoui

From your analysis of a problem to a notebook and back again

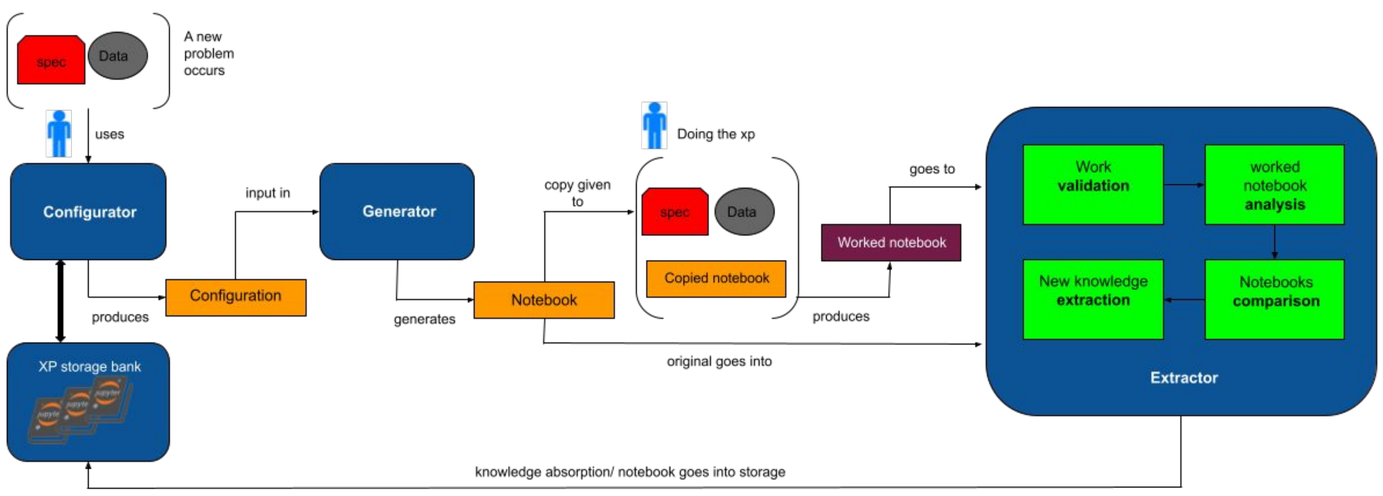
What is it ?

- Generates Notebooks with pre-existing code artifacts according to your problem and solution specifications
- Extracts knowledge from past XPS inside a supervised and monitored environment
- Reduces solution space by questioning the Data scientist

Why are we doing it ?

- To help the Data scientist to specify her problem before leaving the machine to search for a solution
- To avoid Cargo Cult: The « best » workflow will not be the same for each problem [1]
- To reduce the resources needed for learning today: Less meta-learning, more reuse-learning

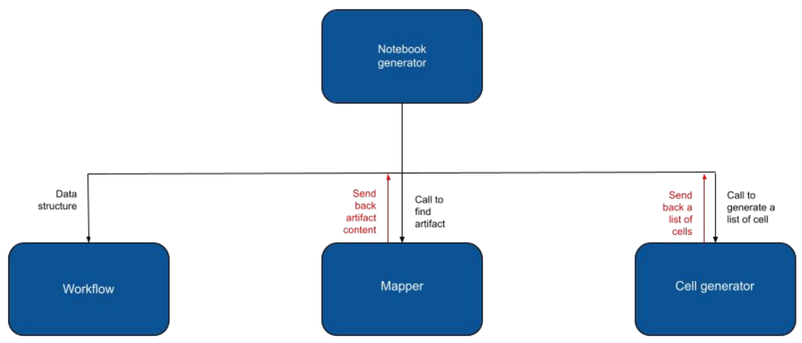
How does it work ?



About the notebook generator:

Challenges:

- Respect data scientists' practices but improve them
- Suggest additional steps (visualization, data grouping,…)
- Make adaptations easier
- Split the work into several workflows (from data analysis to tests)



Realization:

- The input contains a list of artifacts
- Artifacts can be data treatments, algorithms, metrics, visualization
- Artifacts are found by a mapper
- Artifacts are selected by the data scientist during the configuration process

Perspectives

Towards notebook-related knowledge extraction:

- Solution validation
  - Check reproducibility [2]
  - Check problem-solution consistency
- Solution analysis
  - Check the notebook's content
  - Check specific structural patterns
  - Check specific syntactic and semantic pattern
- Notebooks comparison
  - Match cells between original and worked notebooks
  - Compare the content on matched cells
  - Analyze content of new cells
- Solution's knowledge extraction
  - Check the consistency of potential new version of an artifact
  - Extract new content to update artifacts
  - Store notebook and configuration

Towards supervised and monitored environment:

- Architecture requirements analysis:
  - On premise
  - ML student-friendly environment
  - Multi-users
  - Resources Monitoring
  - Research-friendly tech stack
  - Scalability
  - DevOps

References:
- [1] : Wolpert, David (1996), The Lack of A Priori Distinctions between Learning Algorithms, Neural Computation, pp. 1341 – 1390 – No free lunch Theorem
- [2] : Casseau, C.et al.(2021) 'Immediate Feedback forStudents to Solve Notebook ReproducibilityProblems in the Classroom',Proceedings of IEEE Symposiumon Visual Languages andHuman-Centric Computing, VL/HCC. IEEE Computer Society,2010-Octob. doi:10.1109/VL/HCC51201.2021.9576363

Laboratoire I3S – UMR 7271 – CNRS
CS 40121 – 06903 Sophia Antipolis Cedex – France
Contact: yann.brault@etu.unice.fr, blay@i3s.unice.fr

** This work has been supported by the French government, through the UCA DS4H Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-17-EURE-0004."*