CONFERENZA GARR 2025
FRONTIERE DIGITALI

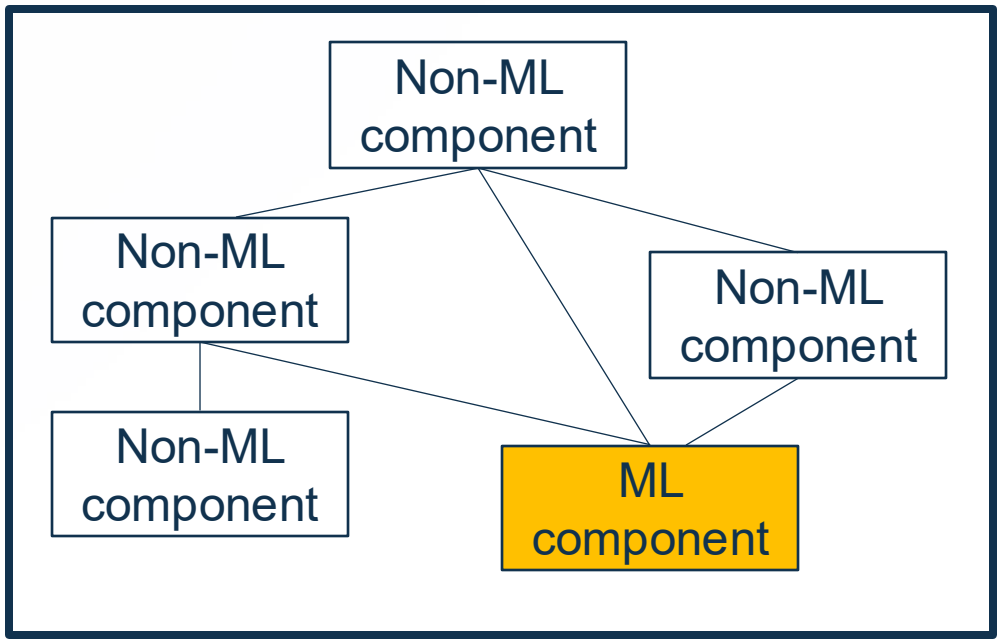
MLOPs as an infrastructure for ML models in production

Prof. Filippo Lanubile

Università di Bari - Dipartimento di Informatica

Machine Learning (ML) models as components of ML-enabled systems

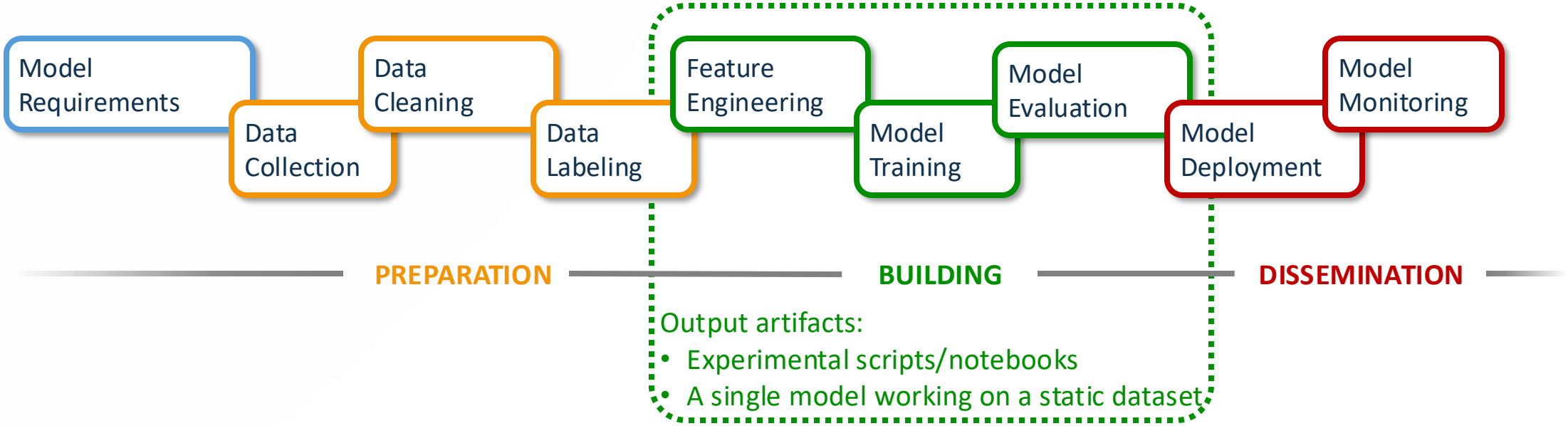
Software systems that uses ML to provide value for users



Environment

- Users
- Sensors
- Actuators

Data scientists tend to focus on ML model building



This is not enough!

The Big Challenge to have an impact:

know how to take an idea and a model developed by data scientists and deploy it as part of a scalable and maintainable ML-enabled system

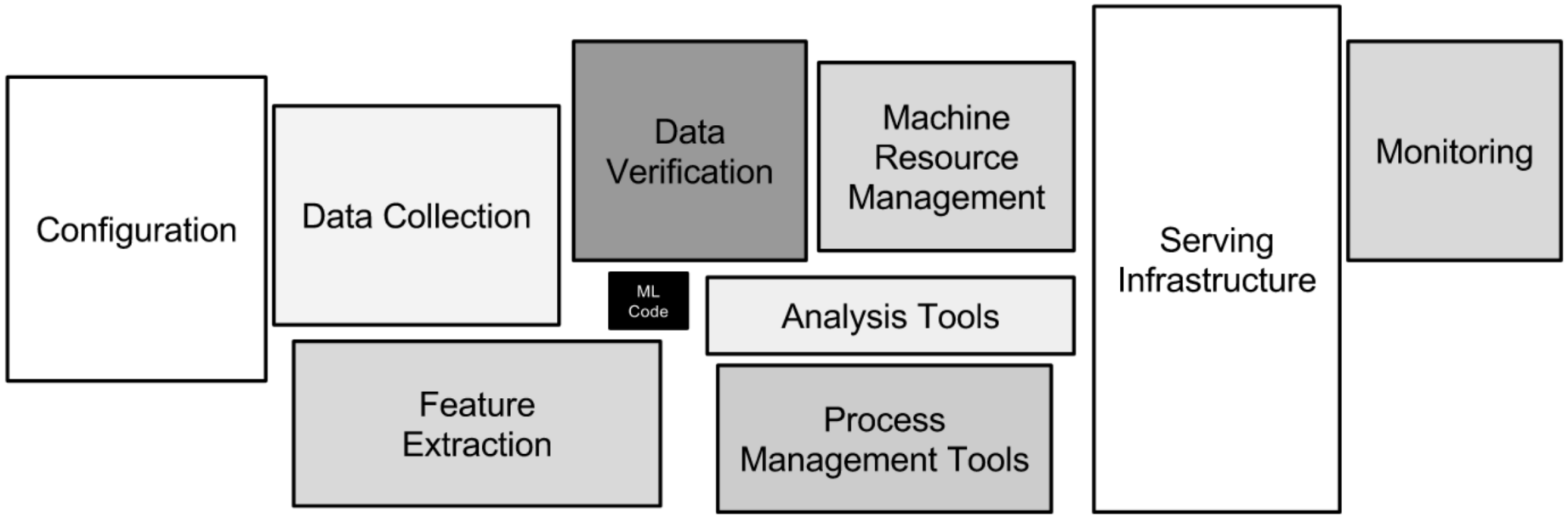
Decomposing the big challenge:
Reproducible and auditable training process

1.4 million notebooks from GitHub: attempted to execute all 753,405 Python notebooks with unambiguous execution order

RQ7. *How reproducible are notebooks?*
Answer: We were able to successfully run 24.11% of the unambiguous execution order Python notebooks. This number is close to the results of a previous reproducibility study [32] about general computer systems research (24.9%). However, the rate is way smaller (4.03%) when we count only notebooks that produce the same results. The most common causes of failures were related to missing dependencies, the presence of hidden states and out-of-order executions, and data accessibility.

J. F. Pimentel, L. Murta, V. Braganholo and J. Freire. A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks. MSR 2019

Decomposing the big challenge:
ML code is just a small fraction of what you need



The required surrounding infrastructure is vast and complex

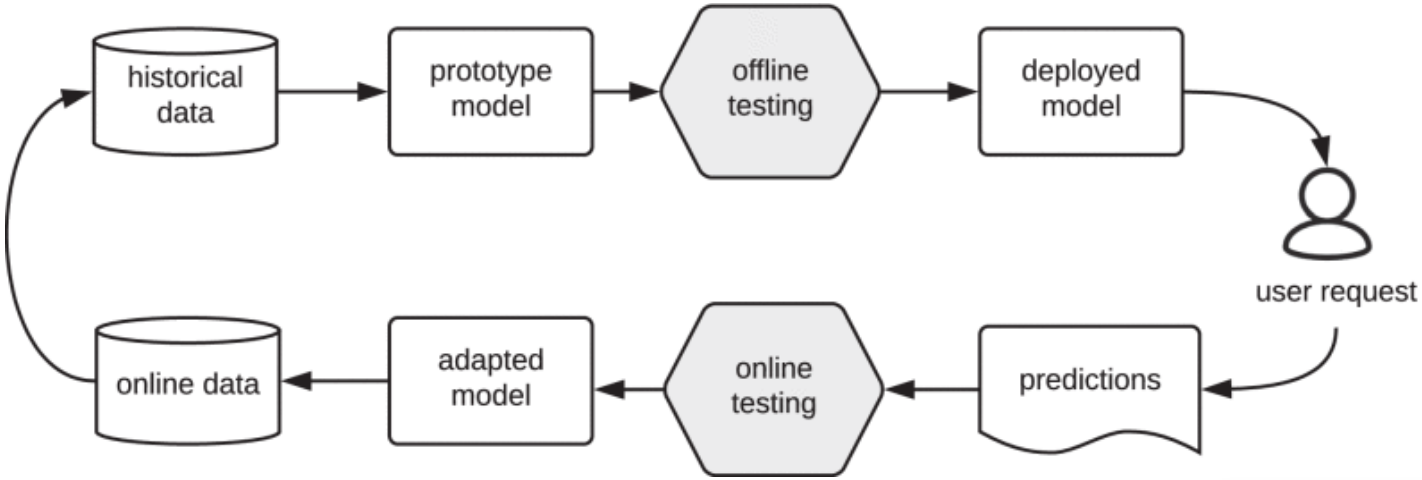
D. Sculley et al., "Hidden technical debt in machine learning systems" NIPS'15: Proc. of the 28th Int. Conf. on Neural Information Processing Systems - 2015

Decomposing the big challenge:
Lack of specification

We cannot test a ML model for correctness, because we do not have a specification of what it means to be correct

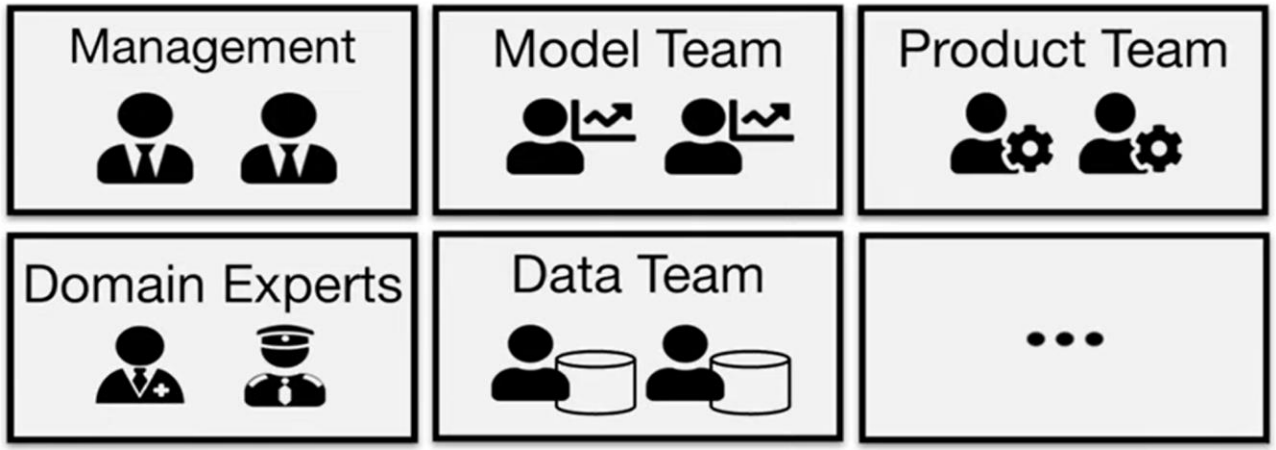
- **We cannot avoid some wrong predictions**

We can evaluate whether a ML model works *well enough* on some test data or in the context of a concrete system



From J. M. Zhang, M. Harman, L. Ma and Y. Liu, "Machine Learning Testing: Survey, Landscapes and Horizons," IEEE TSE, Jan. 2022

Decomposing the big challenge:
Interdisciplinary teams



- Culture clashes: conflicts between data scientists and software engineers
- Lack of ML literacy leads to unrealistic requirements
- Product requirements are often not translated into clear model requirements

Grady Booch
@Grady_Booch

"Machine learning engineering is where we were in Software Engineering 20 years ago. A lot of things still need to be invented. We need to figure out what testing means, what CD (continuous delivery) means, we need to develop tools and environments..."

Traduci il Tweet

ML best practices in PyTorch dev conf 2018
In the Machine Learning (ML) field tools and techniques for best practices are just starting to be developed.
dvc.org

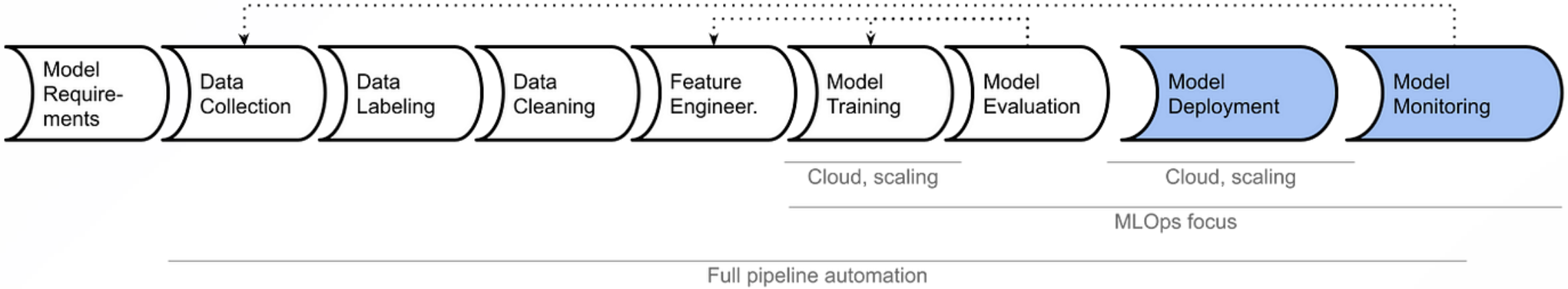
8:49 PM · 24 ago 2021 · Twitter Web App

https://twitter.com/Grady_Booch/status/1430240815058620416?s=20

MLOps comes to help

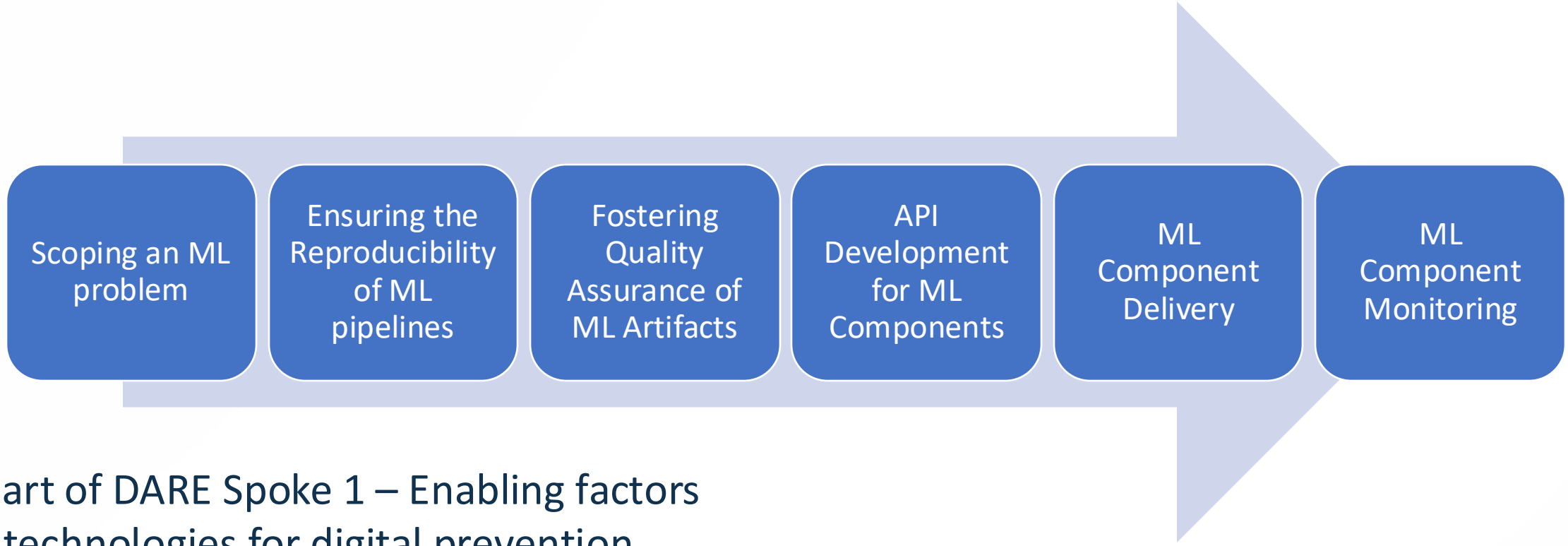
A set of practices and tools to facilitate the creation and evolution of ML-enabled systems

- rooted in software engineering and inspired by DevOps
- **emphasis on the automation of the ML pipeline**



From Christian Kästner. *Machine Learning in Production: From Models to Products*. 2022

MLOps-based Solution Framework to drive the transition from model prototypes to production-ready ML components



As part of DARE Spoke 1 – Enabling factors and technologies for digital prevention

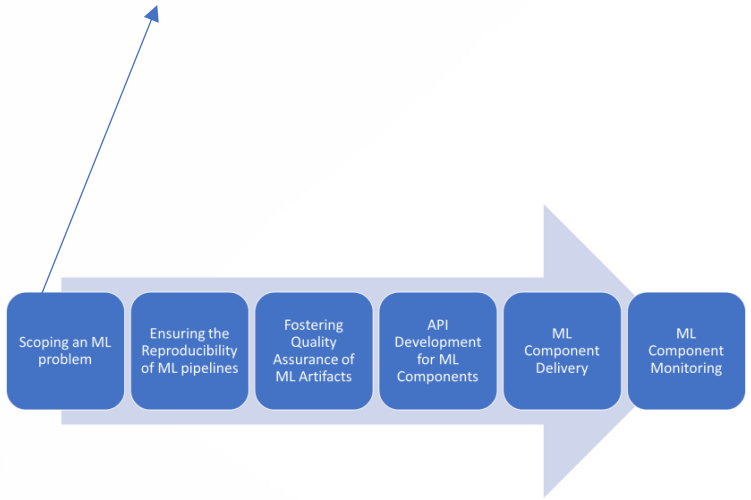
Ministero dell'Università e della Ricerca | Italiadomani | PNC Piano nazionale per gli investimenti complementari al PNRR | DARE DIGITAL LIFELONG PREVENTION

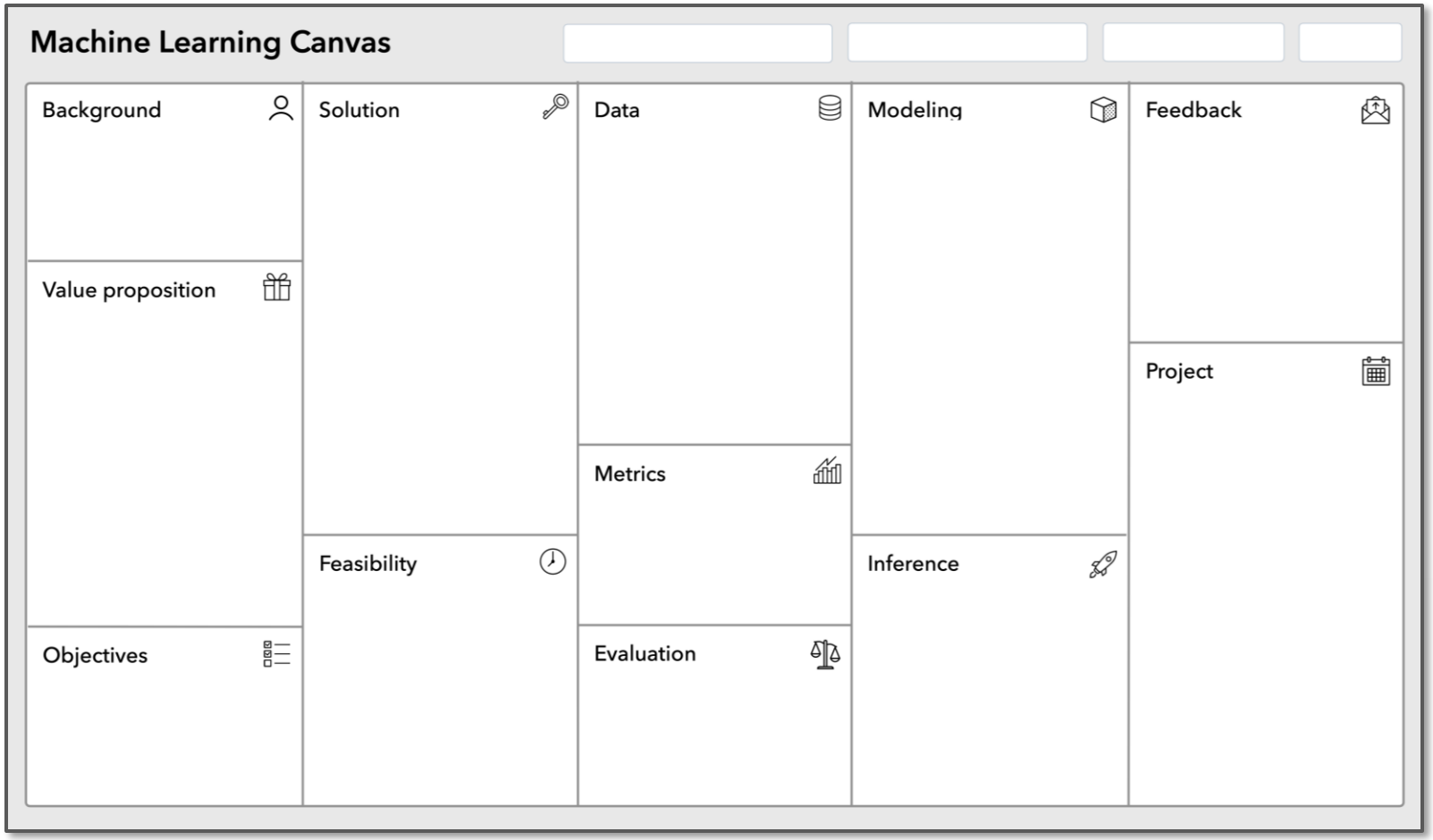
Scoping an ML problem

Scoping an ML problem

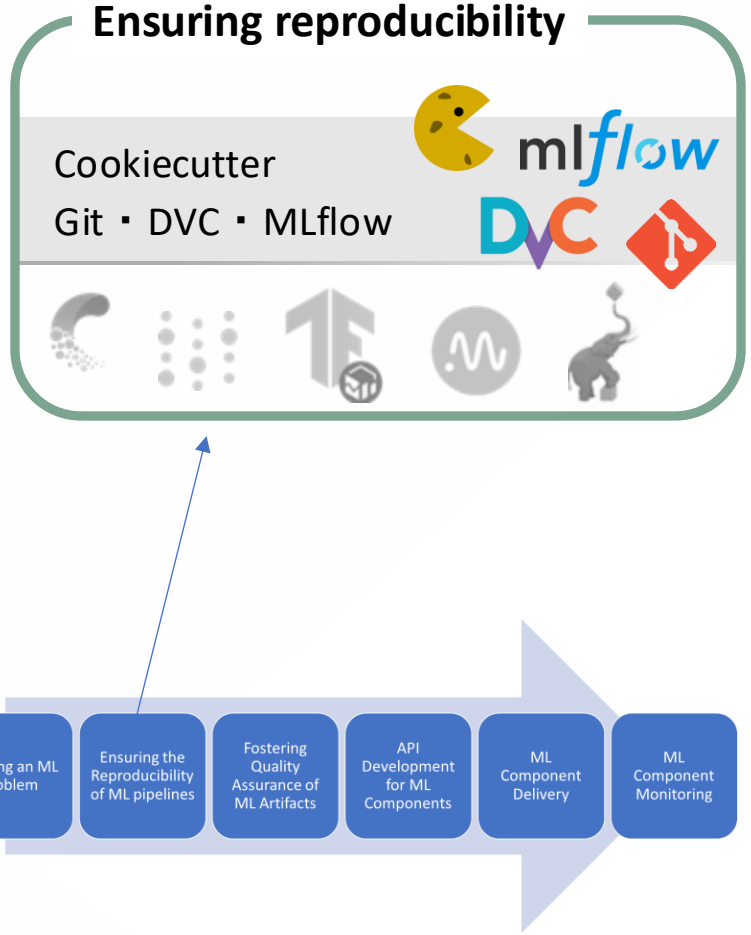
Checklists for requirements elicitation

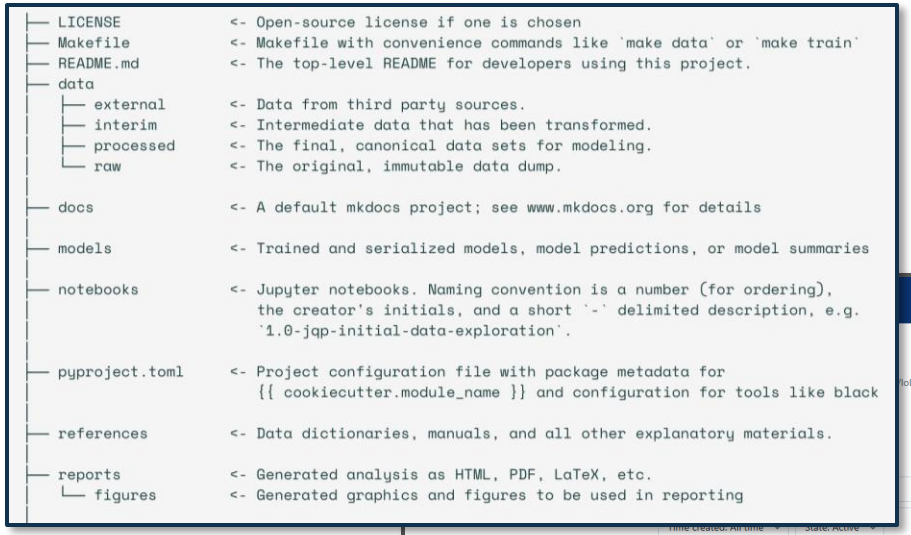
ML Canvas

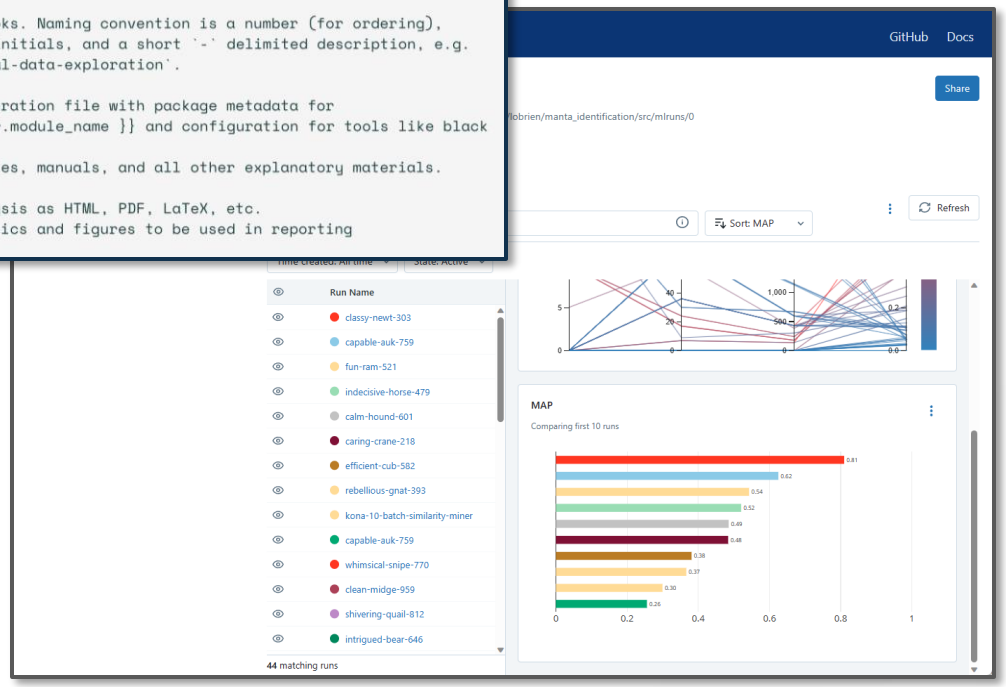




Ensuring the Reproducibility of ML pipelines







Fostering Quality Assurance of ML Artifacts

Fostering QA

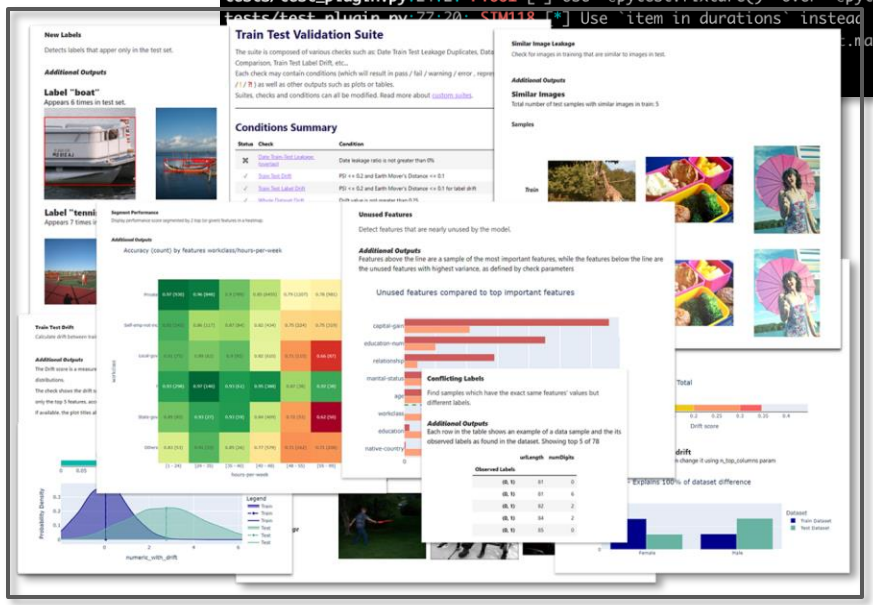
Pytest · Ruff · Pynblint
Code Carbon ·
Deepchecks

my[py]

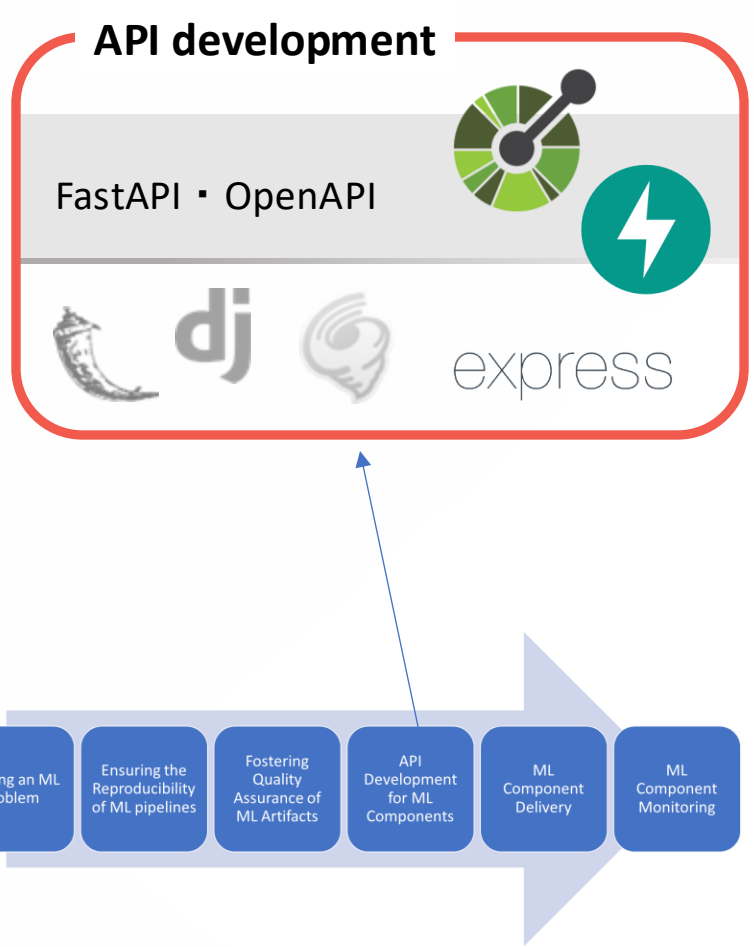
Black

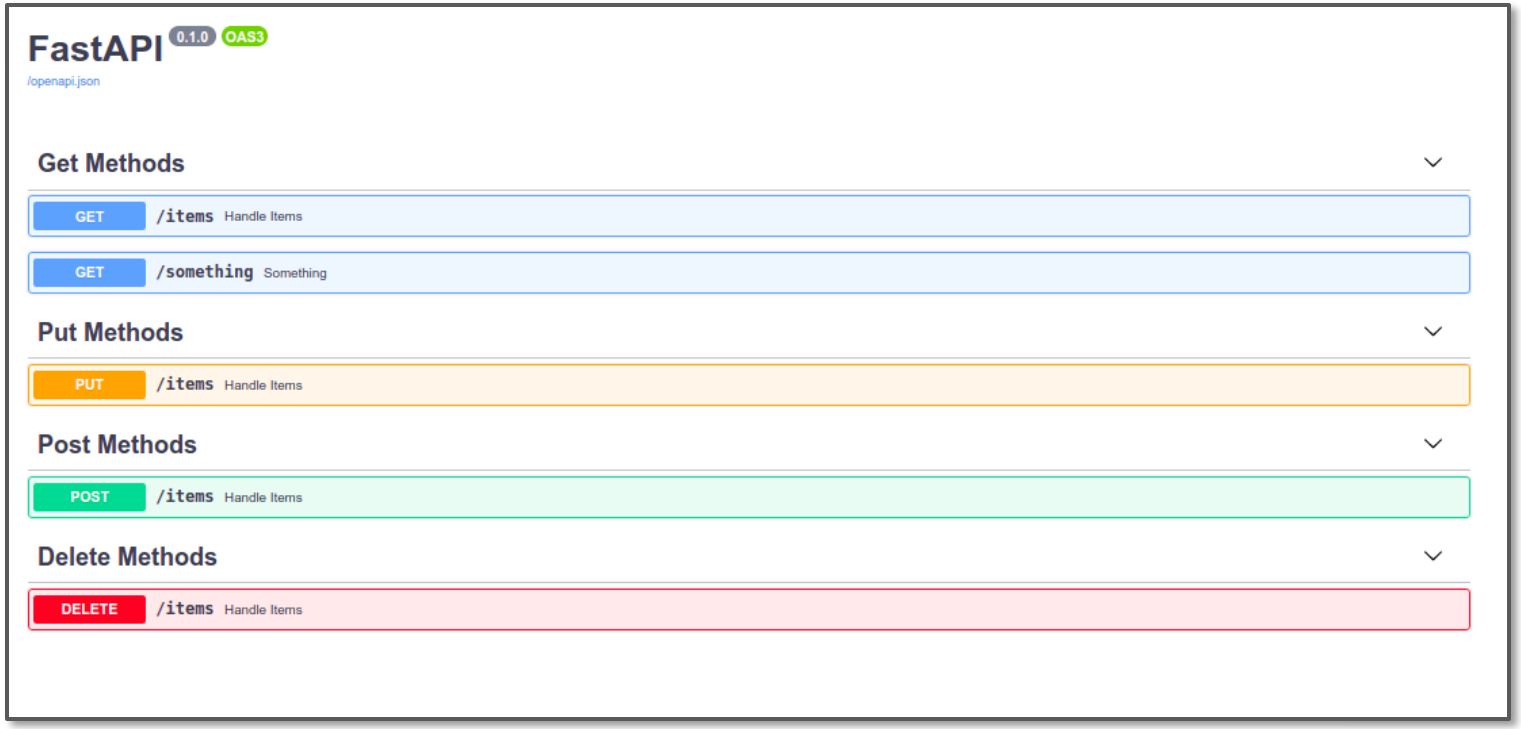
Scoping an ML problem | Ensuring the Reproducibility of ML pipelines | Fostering Quality Assurance of ML Artifacts | API Development for ML Components | ML Component Delivery | ML Component Monitoring

```
(.venv) → pytest-split git:(611def1) ruff check .
src/pytest_split/plugin.py:79:2: PT023 [*] Use `@pytest.mark.tryfirst()` over `@pytest.mark.tryfirst`
src/pytest_split/plugin.py:138:37: C416 [*] Unnecessary `dict` comprehension (rewrite using `dict()`)
src/pytest_split/plugin.py:185:9: RET501 [*] Do not explicitly `return None` in function if it is the only po
src/pytest_split/plugin.py:185:9: PLR1711 [*] Useless `return` statement at end of function
tests/test_algorithms.py:8:1: I001 [*] Import block is un-sorted or un-formatted
tests/test_algorithms.py:20:30: SIM118 [*] Use `x in durations` instead of `x in durations.keys()`
tests/test_algorithms.py:73:9: PT006 [*] Wrong name(s) type in `@pytest.mark.parametrize`, expected `tuple`
tests/test_algorithms.py:97:9: PT006 [*] Wrong name(s) type in `@pytest.mark.parametrize`, expected `tuple`
tests/test_cli.py:1:1: I001 [*] Import block is un-sorted or un-formatted
tests/test_cli.py:12:2: PT001 [*] Use `@pytest.fixture()` over `@pytest.fixture`
tests/test_ipynb.py:1:1: I001 [*] Import block is un-sorted or un-formatted
tests/test_ipynb.py:31:30: SIM118 [*] Use `x in durations` instead of `x in durations.keys()`
tests/test_plugin.py:13:2: PT001 [*] Use `@pytest.fixture()` over `@pytest.fixture`
tests/test_plugin.py:21:5: PT022 [*] No teardown in fixture `example_suite`, use `return` instead of `yield`
tests/test_plugin.py:24:2: PT001 [*] Use `@pytest.fixture()` over `@pytest.fixture`
tests/test_plugin.py:77:20: SIM118 [*] Use `item in durations` instead of `item in durations.keys()`
                                                                    ark.parametrize`, expected `tuple`
```

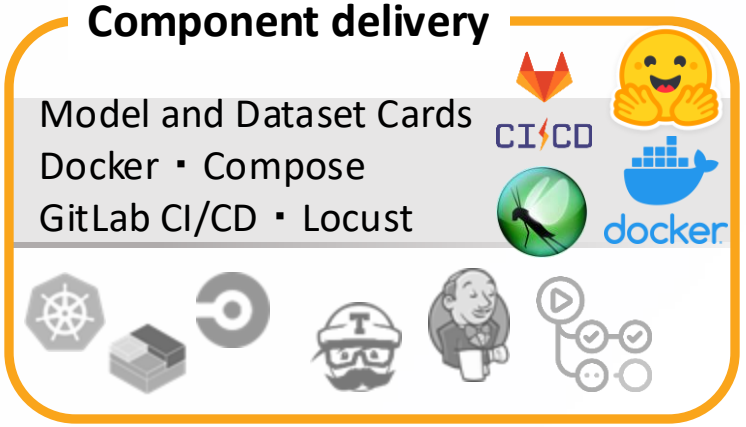


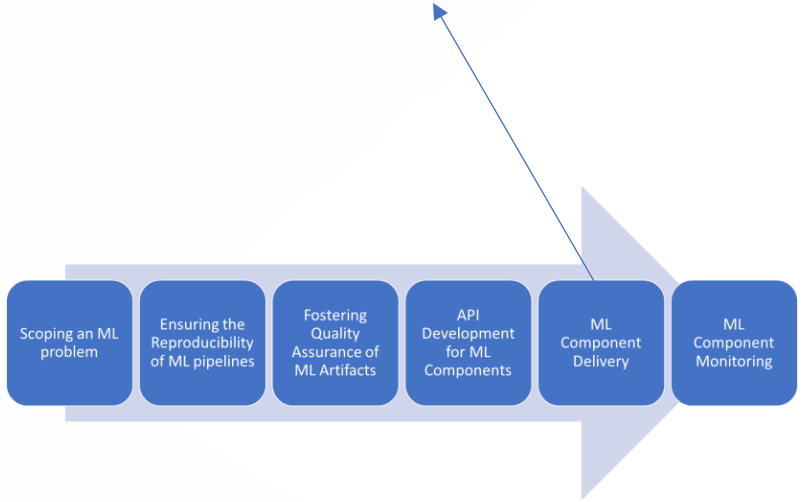
API Development for ML





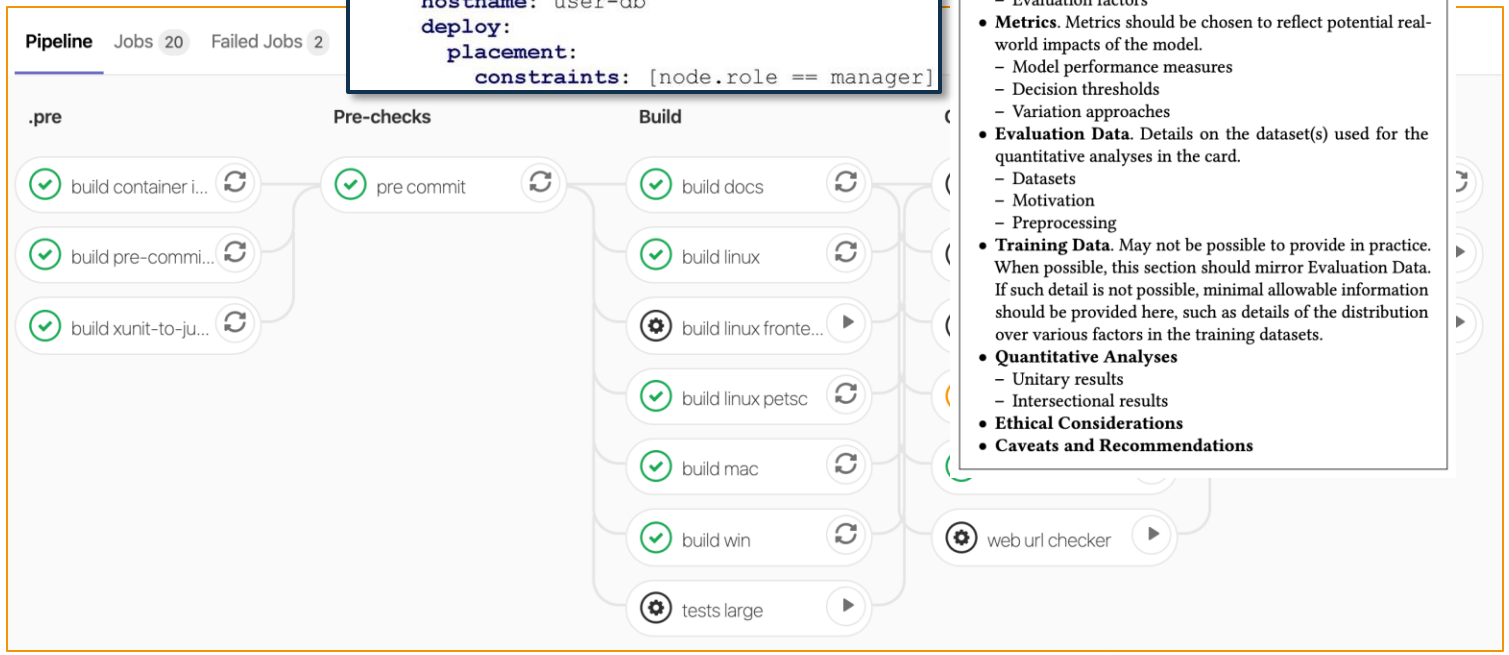
ML Component Delivery






```yaml
services:
  recommendation-engine:
    image: ubuntu
    tty: true
    volumes:
    - DataVolume:/DataVolume
    labels:
      brownout.feature: "optional"
    deploy:
      replicas: 2
      restart_policy:
        condition: none
      placement:
        constraints: [node.role == worker]

  user-db:
    image: weaveworksdemos/user-db
    hostname: user-db
    deploy:
      placement:
        constraints: [node.role == manager]
```



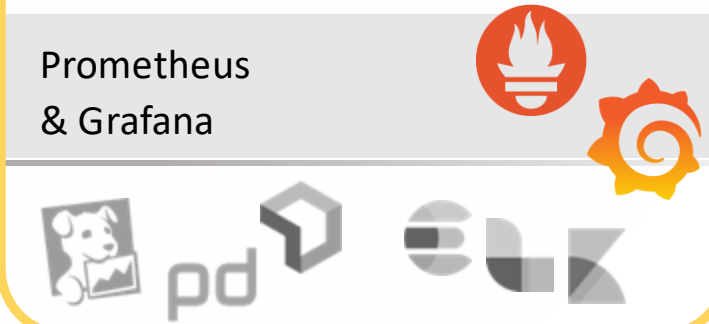
Model Card

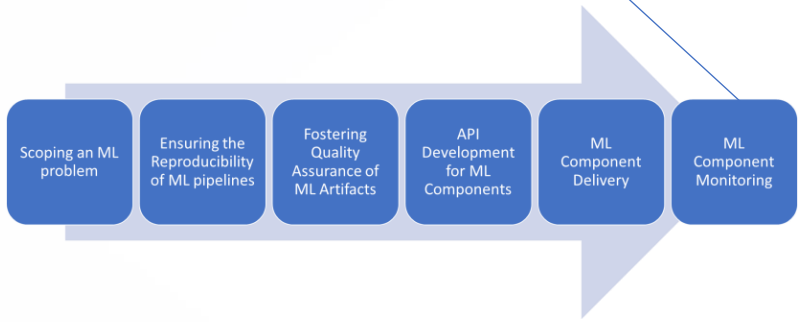
- **Model Details**. Basic information about the model.
  - Person or organization developing model
  - Model date
  - Model version
  - Model type
  - Information about training algorithms, parameters, fairness constraints or other applied approaches, and features
  - Paper or other resource for more information
  - Citation details
  - License
  - Where to send questions or comments about the model
- **Intended Use**. Use cases that were envisioned during development.
  - Primary intended uses
  - Primary intended users
  - Out-of-scope use cases
- **Factors**. Factors could include demographic or phenotypic groups, environmental conditions, technical attributes, or others listed in Section 4.3.
  - Relevant factors
  - Evaluation factors
- **Metrics**. Metrics should be chosen to reflect potential real-world impacts of the model.
  - Model performance measures
  - Decision thresholds
  - Variation approaches
- **Evaluation Data**. Details on the dataset(s) used for the quantitative analyses in the card.
  - Datasets
  - Motivation
  - Preprocessing
- **Training Data**. May not be possible to provide in practice. When possible, this section should mirror Evaluation Data. If such detail is not possible, minimal allowable information should be provided here, such as details of the distribution over various factors in the training datasets.
- **Quantitative Analyses**
  - Unitary results
  - Intersectional results
- **Ethical Considerations**
- **Caveats and Recommendations**

ML Component Monitoring

Monitoring

Prometheus
& Grafana





Pilot Project:
Predicting neurodegenerative diseases and brain aging

Center for Neurodegenerative Diseases and the Aging Brain (CMND)

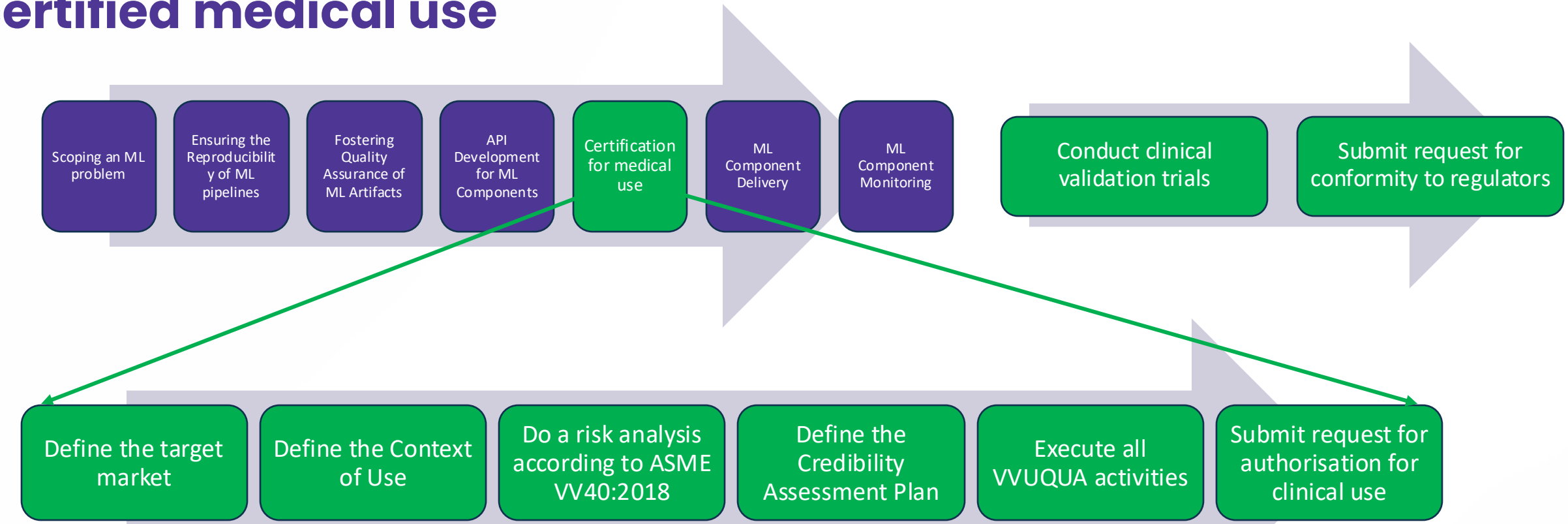
University of Bari Aldo Moro, Tricase (LE)

- directed by Prof. Giancarlo Logroscino MD
- dedicated to the research, diagnosis, and treatment of neurodegenerative diseases (Alzheimer's, Parkinson's, ALS, and other nervous system disorders)
- Advanced diagnostic techniques such as functional MRI (fMRI) and PET scans

UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

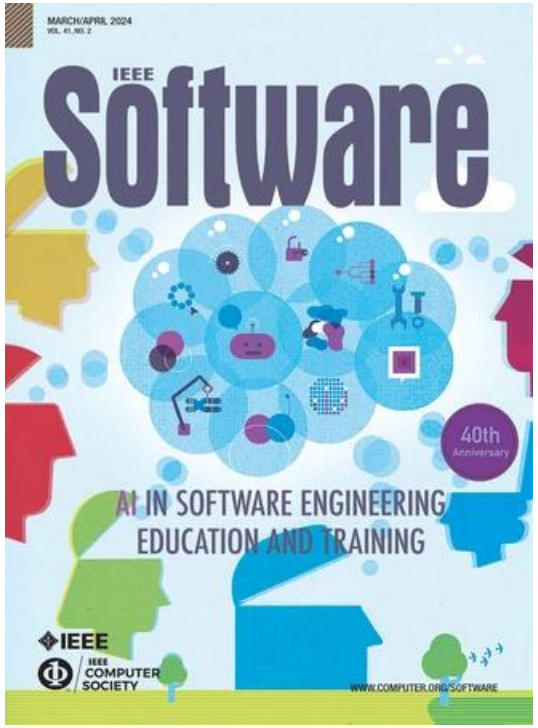
PIA FONDAZIONE DI CULTO E RELIGIONE CARDINALE GIOVANNI PANICO
Azienda Ospedaliera

Evolving the MLOps-based Solution Framework for certified medical use



Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | PNC Piano nazionale per gli investimenti complementari al PNRR Ministero dell'Università e della Ricerca | DARE DIGITAL LIFELONG PREVENTION

Teaching MLOps through project-based courses



IEEE Software

Training Future Machine Learning Engineers: A Project-Based Course on MLOps

Mar.-Apr. 2024, pp. 60-67, vol. 41
DOI Bookmark: 10.1109/MS.2023.3310768

Authors

Filippo Lanubile, Department of Informatics and leads the Collaborative Development Research Group, University of Bari, Bari, Italy
Silverio Martinez-Fernandez, Universitat Politècnica de Catalunya-BarcelonaTech, Barcelona, Spain
Luigi Quaranta, Collaborative Development Research Group, University of Bari, Bari, Italy

Thank You!

This research was co-funded by the Italian Complementary National Plan PNC-I.1 "Research initiatives for innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, "DARE - DigitAl lifelong pRevEntion" initiative, code PNC0000002, CUP: B53C22006420001