ML-Ops Intensive

Part 2: Machine Learning Operations and Software Engineering for Machine Learning

Alexander Myltsev[1], Vasily Safronov[1], Andrey Ustyuzhanin[1]

[1] HSE University

April 9, 2021
Skolkovo Institute of Science and Technology

LAMBDA • HSE

Quick self intro

- Alexander Myltsev
  - Graduated from MIPT
  - Backend developer, 12 years (Java/Scala, C#/F#, C++)
  - Researcher at Laboratory of Methods for Big Data Analysis (LAMBDA), HSE University
  - Develop ML models and ship them to end user
- Vasily Safronov – researcher LAMBDA, HSE University
- Andrey Ustyuzhanin – head of LAMBDA, HSE University

Outline

- Motivation for MLOps
- Jupyter-style
- Pipelines and DAGs
- Experiment monitoring

So, what's the big deal?

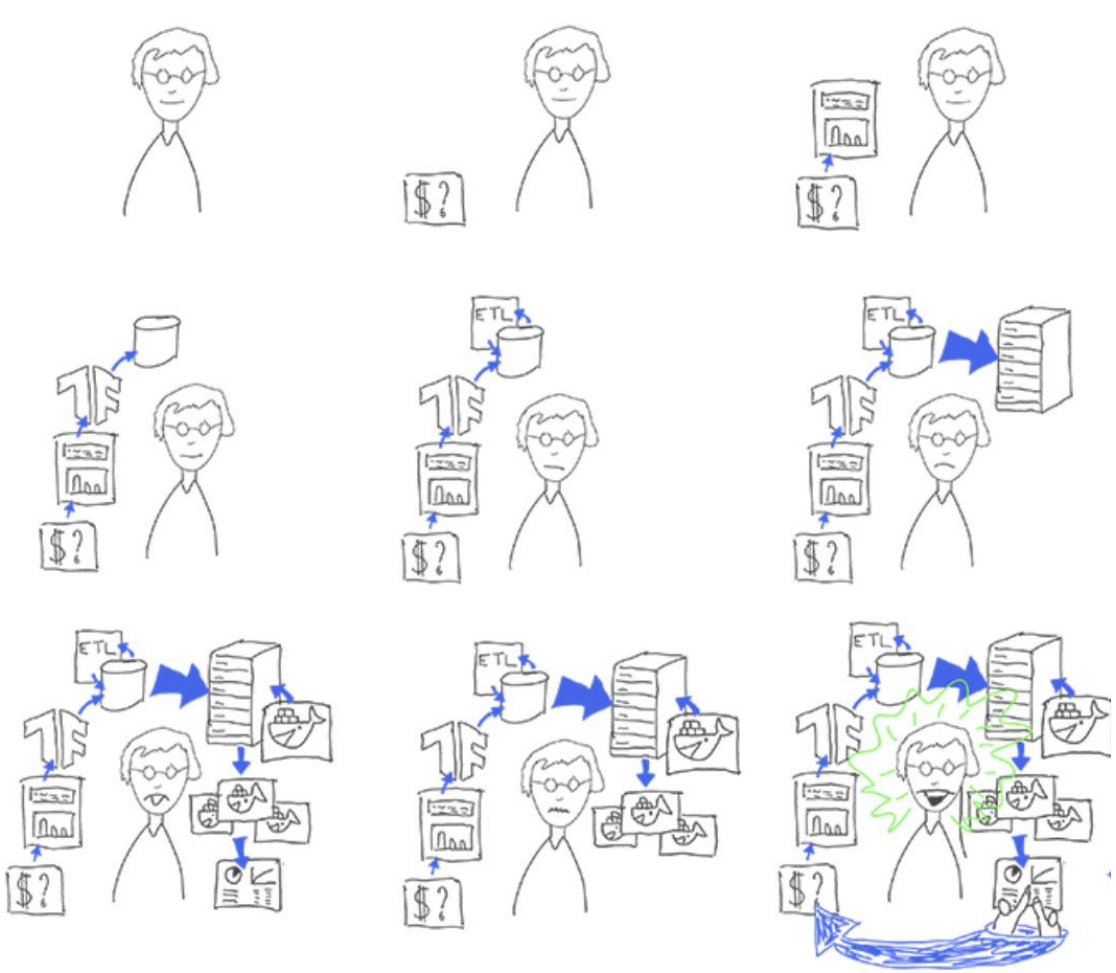
I'm a **data scientist** and not a **software engineer**.

Why do I need that?

https://www.youtube.com/watch?v=-oMZAS9qfrE

So, what's the big deal?

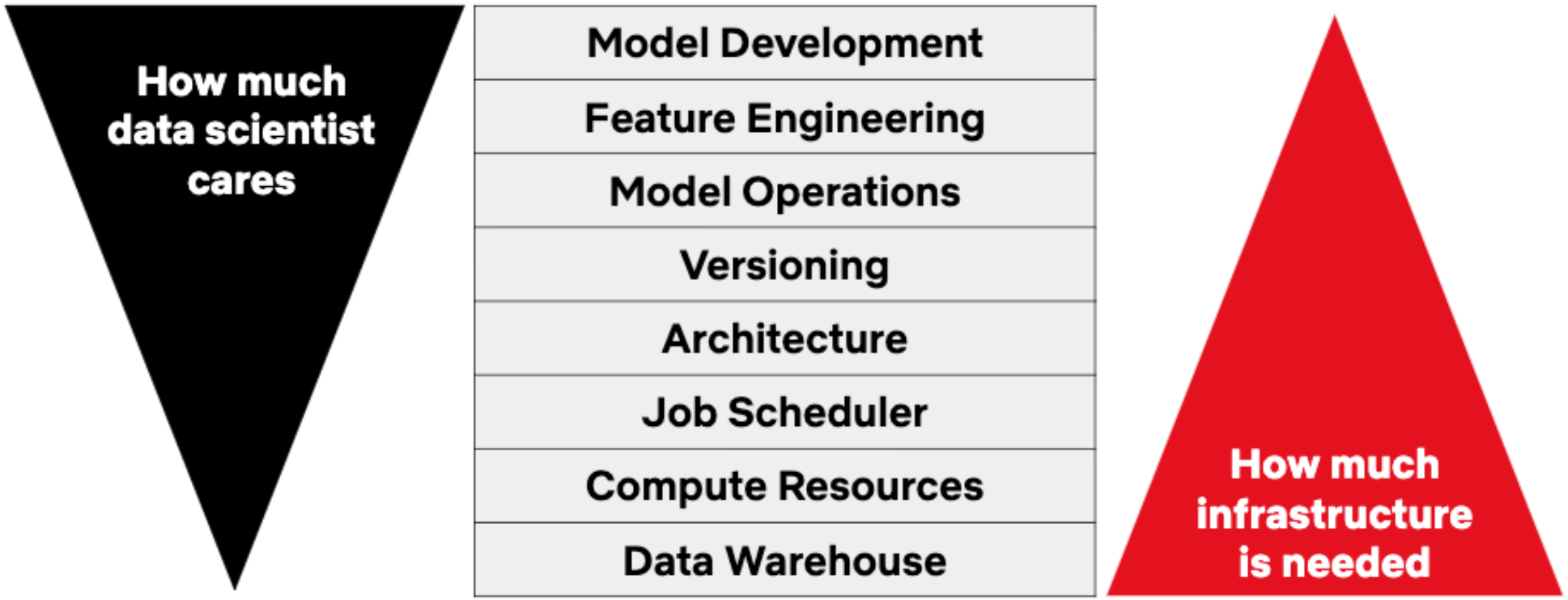
Because **ML** has become **complex**.



https://www.youtube.com/watch?v=-oMZAS9qfrE
https://42.tut.by/385837

So, what's the big deal?

Infrastructure Stack for Data Science



https://docs.metaflow.org/introduction/what-is-metaflow

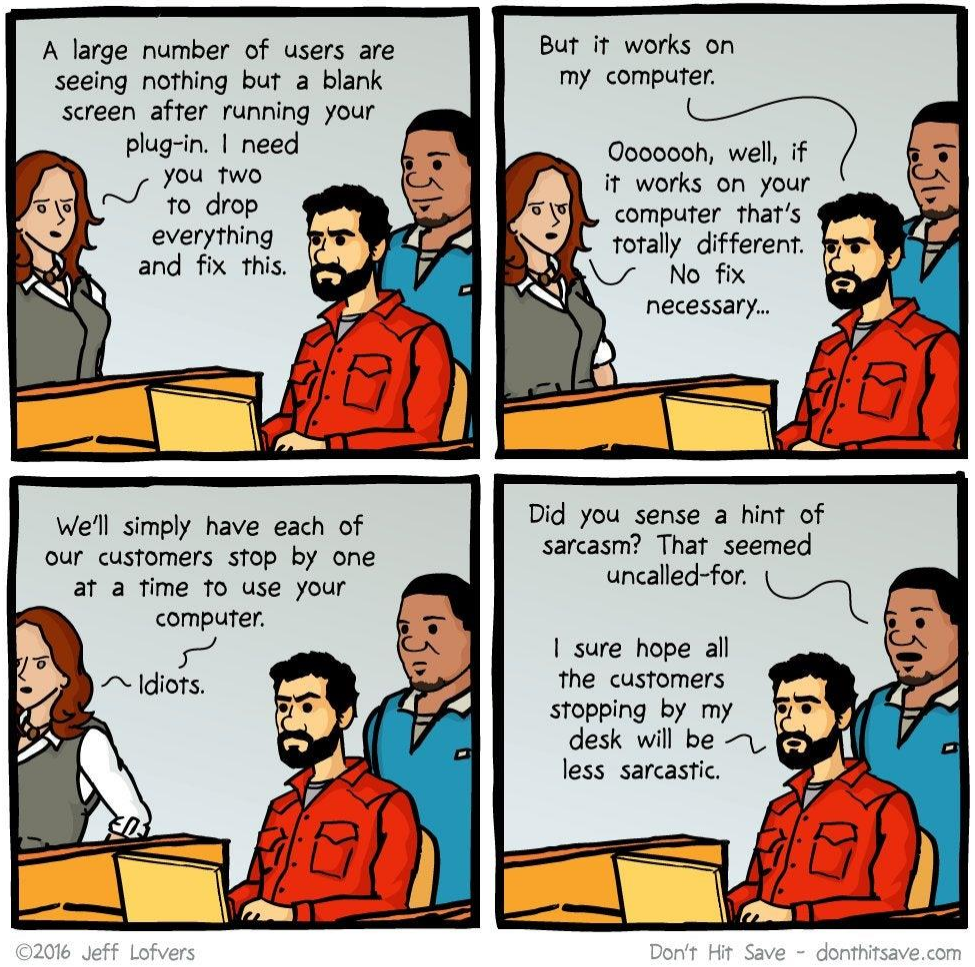
Do ML for the people around you

To share your result you would probably need:

- more **powerful** computer
- **reproduce the environment** on colleague's computer
  - without chatting him for hours
- **version** the code, data and model
- attach to **complicated data sources** and pipelines
- **compare** the model with a one you made 6 months ago

Do ML for the people around you

Reproducibility – **reproduce** the experiment **anytime** & **anywhere**



Jupyter Notebooks

Powerful Jupyter notebook

- How to launch it on **powerful computer** for many days?
- How to **reopen it after several days** and see the results?
- How to edit it with **other people**?
- How to **version** it?
- How to extract **a reusable code** from it?

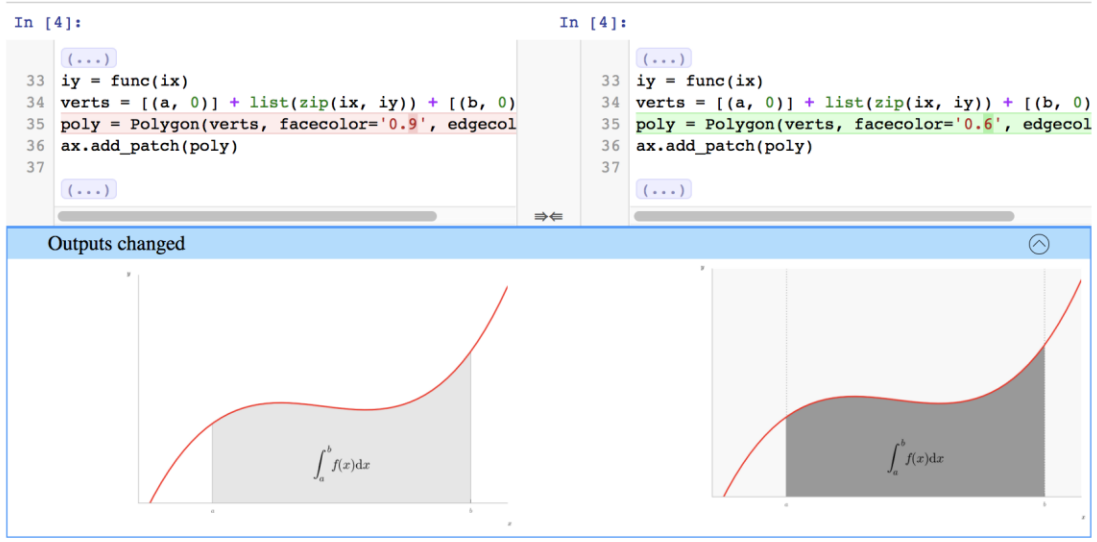
Notebooks in Cloud

- Cluster at your organisation
- Kubeflow Notebook Servers
  - https://www.kubeflow.org/docs/components/notebooks/
- Yandex DataSphere
  - https://cloud.yandex.ru/services/datasphere

Notebooks diffing and merging

- ipynb is a json format
  - https://ipython.org/ipython-doc/3/notebook/nbformat.html
- nbdime – diffing and merging of Jupyter Notebooks
  - https://nbdime.readthedocs.io/

```
{
  "metadata" : {
    "signature": "hex-digest", # used for authenticating unsafe outputs on load
    "kernel_info": {
        # if kernel_info is defined, its name field is required.
        "name" : "the name of the kernel"
    },
    "language_info": {
        # if language_info is defined, its name field is required.
        "name" : "the programming language of the kernel",
        "version": "the version of the language",
        "codemirror_mode": "The name of the codemirror mode to use [optional]"
    }
  },
  "nbformat": 4,
  "nbformat_minor": 0,
  "cells" : [
      # list of cell dictionaries, see below
  ],
}
```



Notebooks to Code

Ready to launch the notebook

- How to convert notebook to code?
  - E.g. cluster supports only task scheduler
- nbconvert lets convert notebooks to other formats
  - https://nbconvert.readthedocs.io/
- papermill lets parameterize and execute notebooks
  - https://papermill.readthedocs.io/

Software engineering best practises

- Statically type your code
  - mypy (Dropbox), pytype (Google), pyre (Facebook), pyright (Microsoft)
  - You can use them simultaneously
- Lint your code
  - flake8, black
- Test your code
  - pytest
  - Even Jupyter notebooks: pytest --nbval
    - https://github.com/computationalmodelling/nbval

Software engineering best practises

- Manage your dependencies
  - E.g. the code won't survive next major update of dependent libs:

Raw | Blame

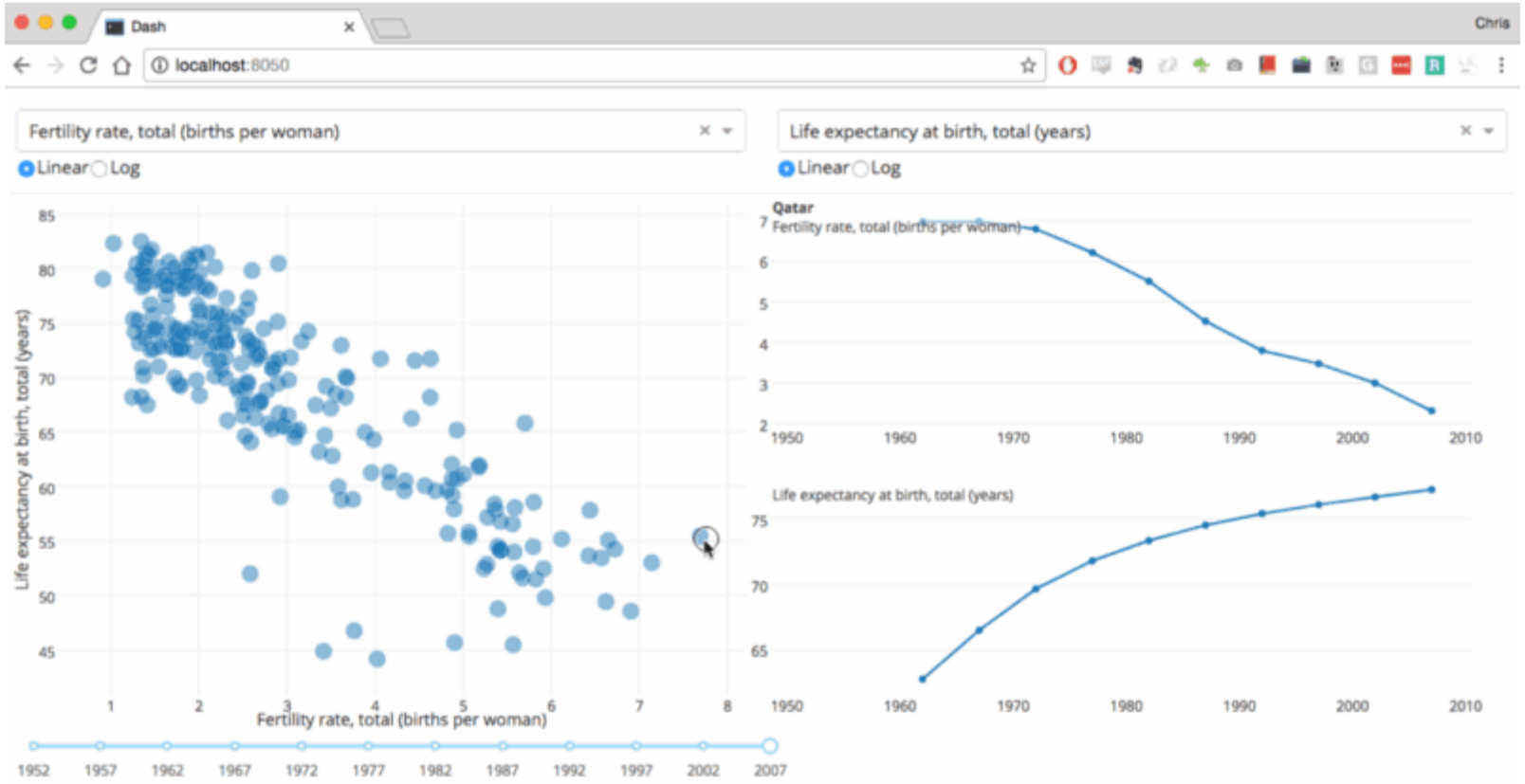
5 lines (5 sloc) | 62 Bytes

```
pandas>=0.25.0
numpy>=1.16.5
scipy>=0.14.0
matplotlib
requests
```

- pyenv+poetry, pipenv

Visualise the result

- Jupyter notebook
- Tensorboard
- Dash Plotly



https://medium.com/plotly/introducing-dash-5ecf7191b503

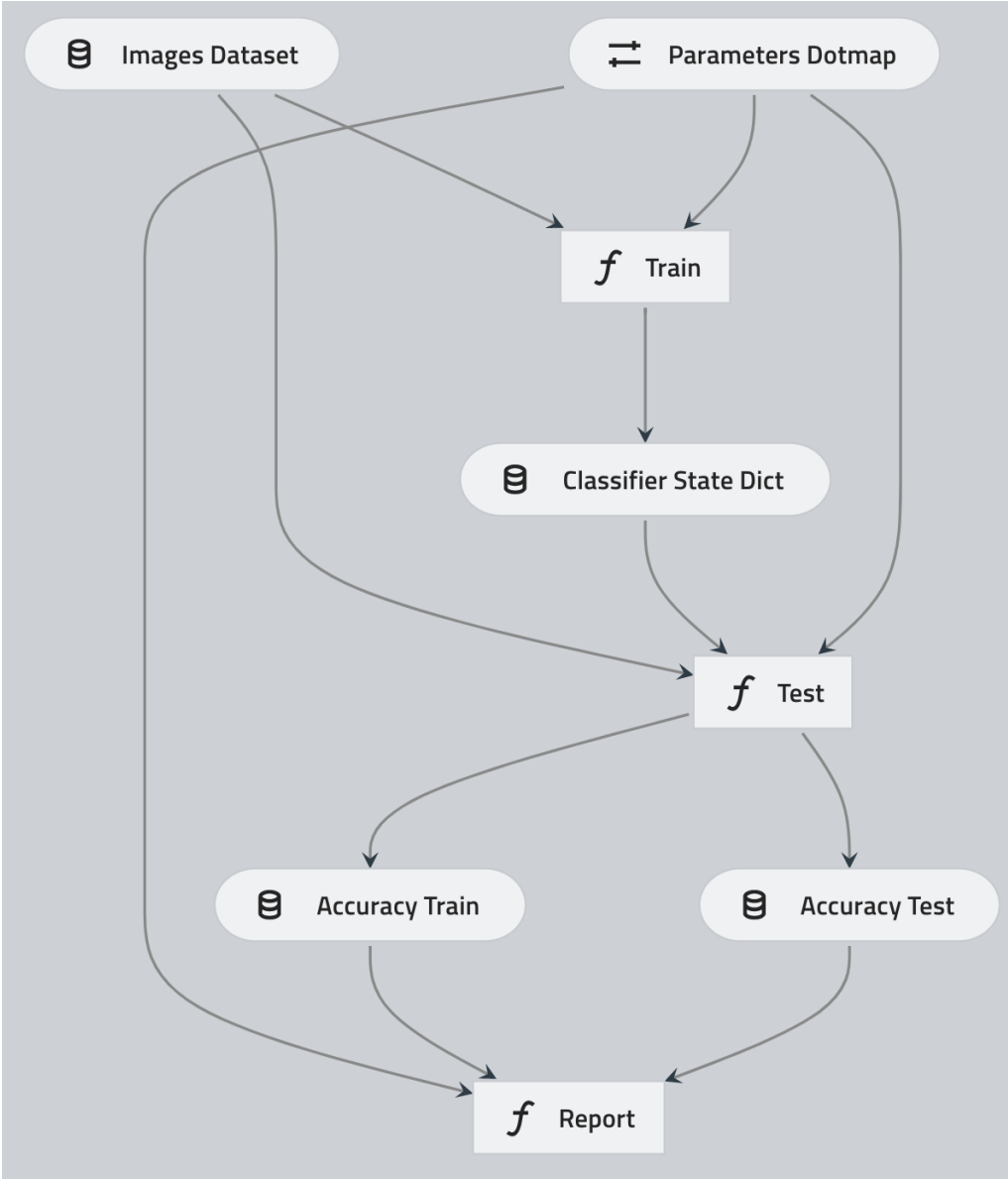
IDE support of notebooks

- VSCode
- JetBrains PyCharm & DataSpell
- nbdev is a library that allows you to develop a python library in Jupyter Notebooks, putting all your code, tests and documentation in one place
  - https://github.com/fastai/nbdev

Pipeline and DAG

What are pipelines and DAGs?

- DAG is for “directed asyclic graph”
- ML pipeline is a DAG
- It has input, output and code inside



Why DAGs?

- DAG is a mathematical abstraction of a pipeline
- ML frameworks expect pipeline decomposition to DAG
- In return they give
  - Pipeline visualisation
  - Pipeline parallelisation
  - Resume execution after fail
  - Run every node on different hardware
  - Convert your DAG to another system

Kedro

- Developed by QuantumBlack (McKinsey)
- open-source Python framework for creating reproducible, maintainable and modular data science code
- Features
  - Project structure
  - Data catalogue
  - Pipeline abstraction
  - Coding standards best practises
  - Flexible deployment
- Command Line Interface
- Addons

Demo

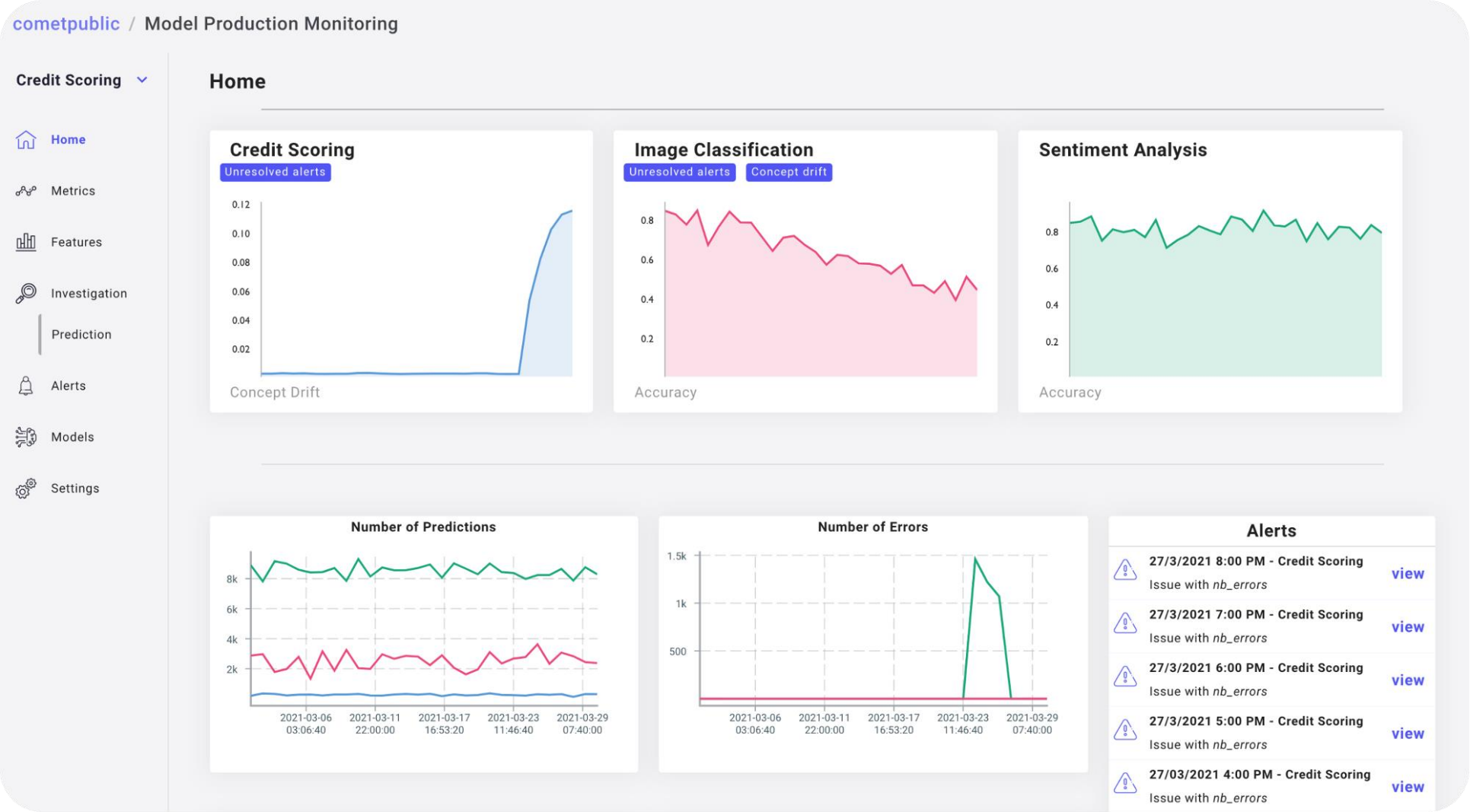
Comet.ml

- track, compare, explain and optimize experiments and models across the model's entire lifecycle

$ pip install comet-ml

```
from comet_ml import Experiment
experiment = Experiment(project_name="my-project", workspace="my-workspace")
experiment.log_parameters(params_dict)
```

Comet.ml



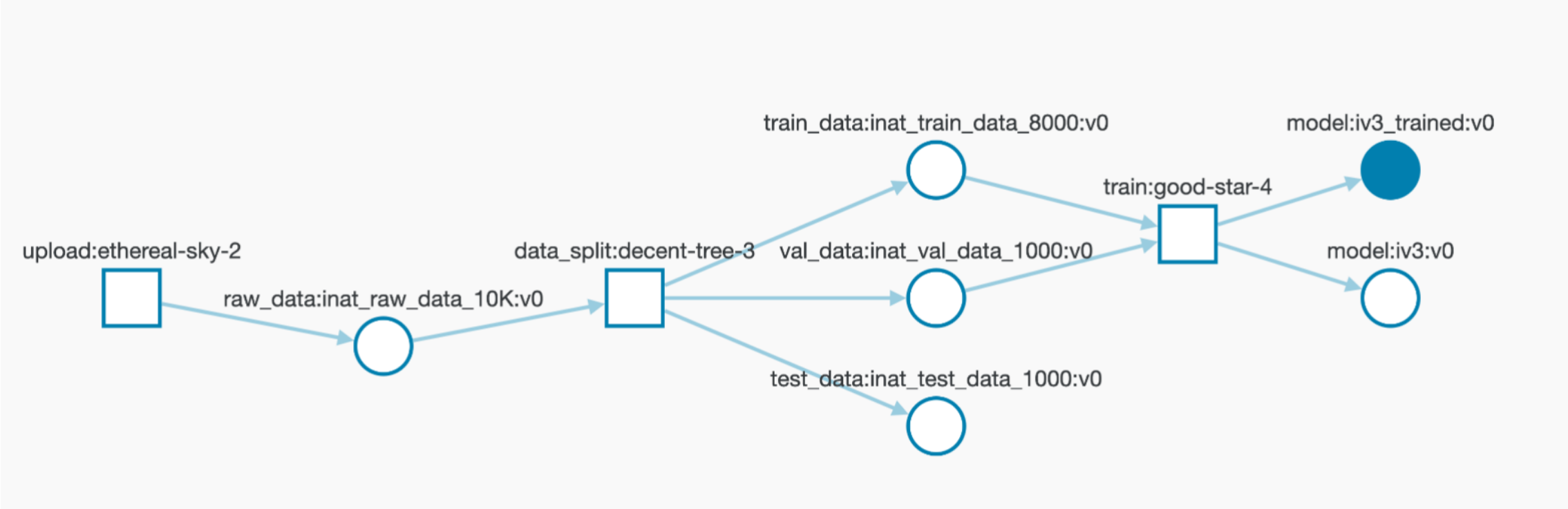
https://www.comet.ml/

WandB.io

- Mostly the same as comet.ml
- Additional features
  - Reports generator
  - Artifacts
    - https://wandb.ai/wandb/arttest/reports/Intro-to-W-B-Artifacts--VmlldzozNTAzMDM

WandB.io

```
run = wandb.init(project=PROJECT_NAME, job_type="train", config=config_defaults)

train_data = run.use_artifact(train_at, type='train_data’)

trained_model_artifact = wandb.Artifact(MODEL_NAME, type="model",
                                        description="trained inception v3", metadata=dict(cfg))
```



MLFlow

- Developed by Databricks
- MLflow is an open source platform to manage the ML lifecycle

MLflow Tracking

Record and query experiments: code, data, config, and results

Read more

MLflow Projects

Package data science code in a format to reproduce runs on any platform

Read more

MLflow Models

Deploy machine learning models in diverse serving environments

Read more

Model Registry

Store, annotate, discover, and manage models in a central repository

Read more

https://mlflow.org/

MLFlow

- MLFlow works as a server. Can be run locally

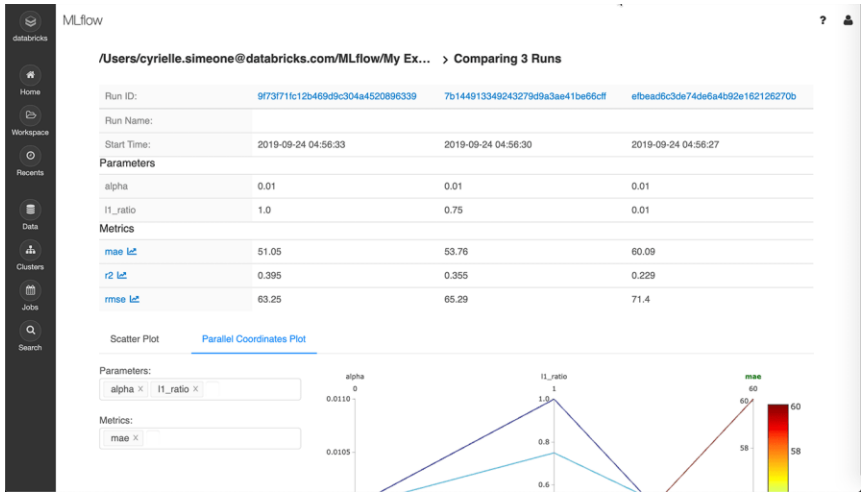
  mlflow ui

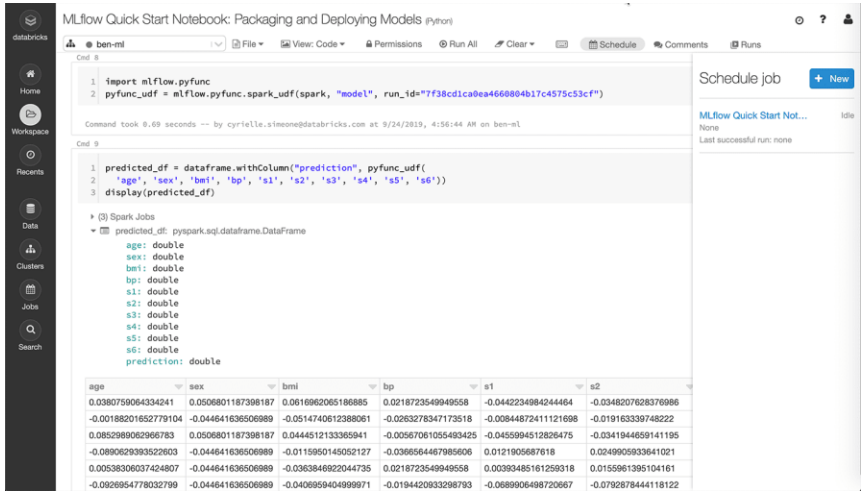
  view it at http://localhost:5000

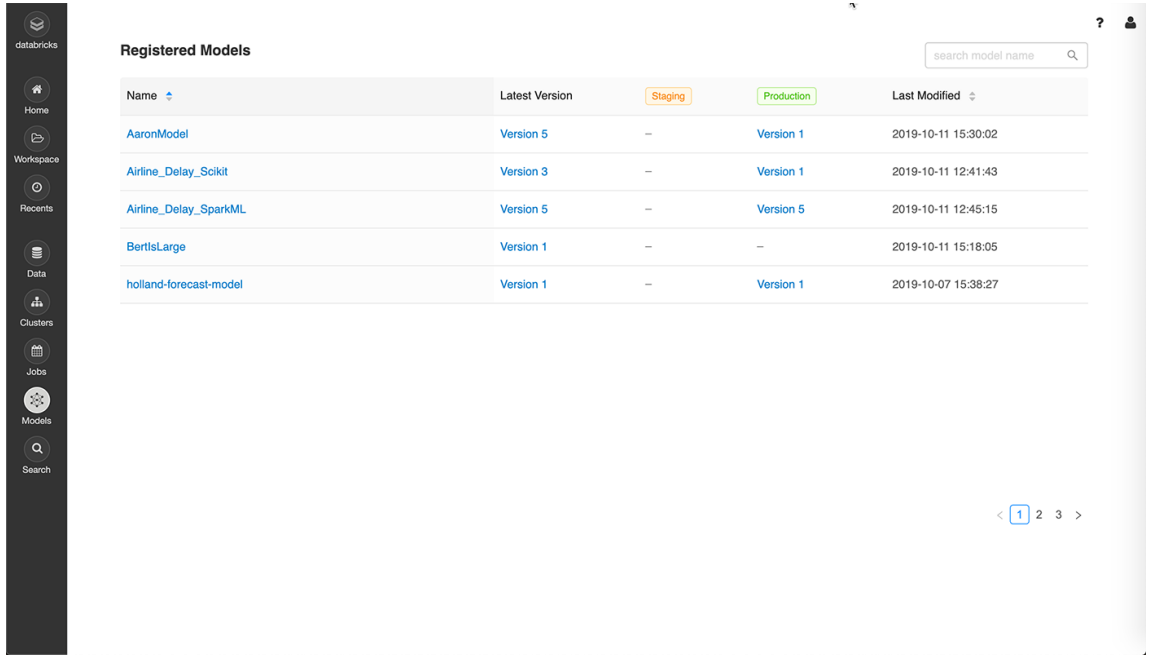
- Add Python library that sends pipeline info to it

  ```
  $ pip install mlflow
  with mlflow.start_run():
      mlflow.log_param("x", 1)
      mlflow.log_metric("y", 2)
  ```

MLFlow







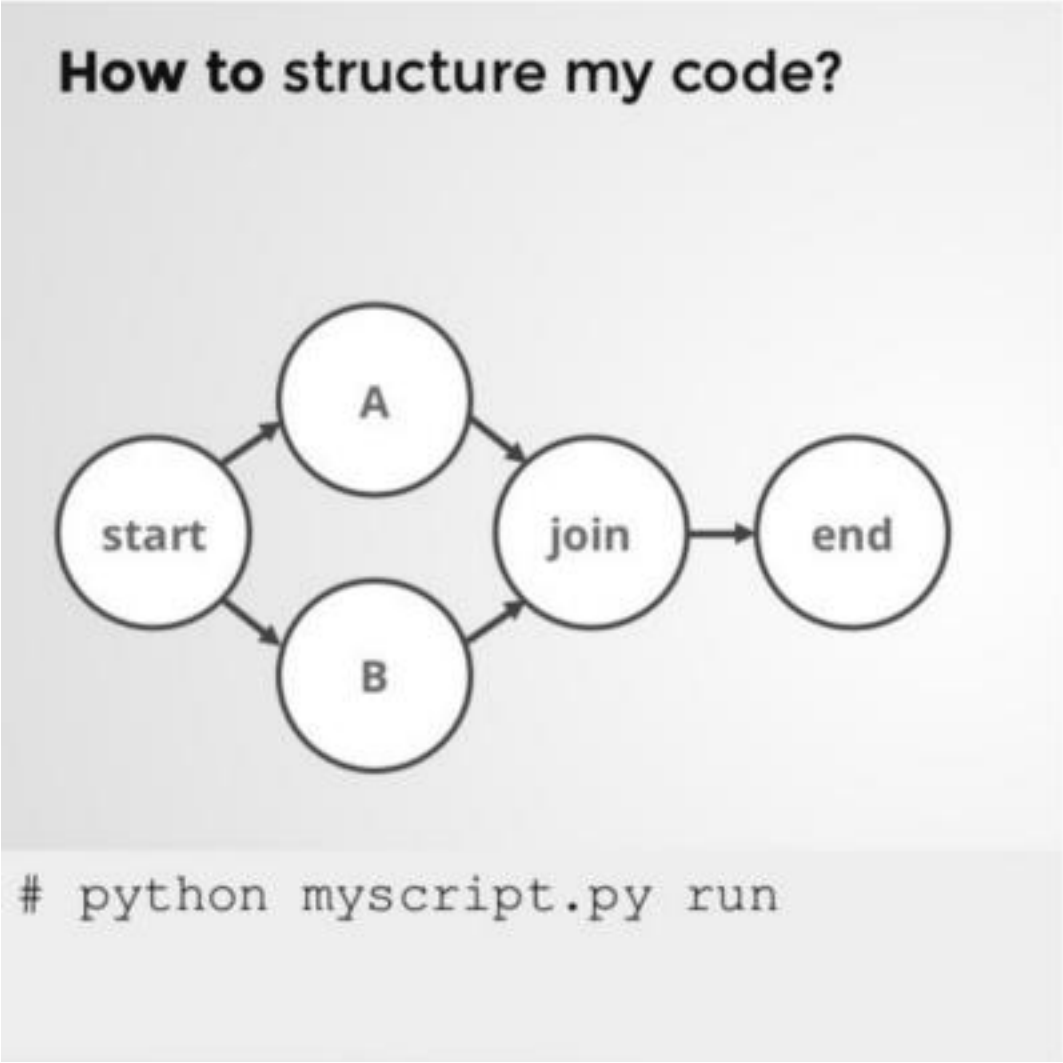
https://databricks.com/product/managed-mlflow

Metaflow

- Developed by Netflix
- Python/R library that helps scientists and engineers build and manage real-life data science projects

$ pip install metaflow

Metaflow

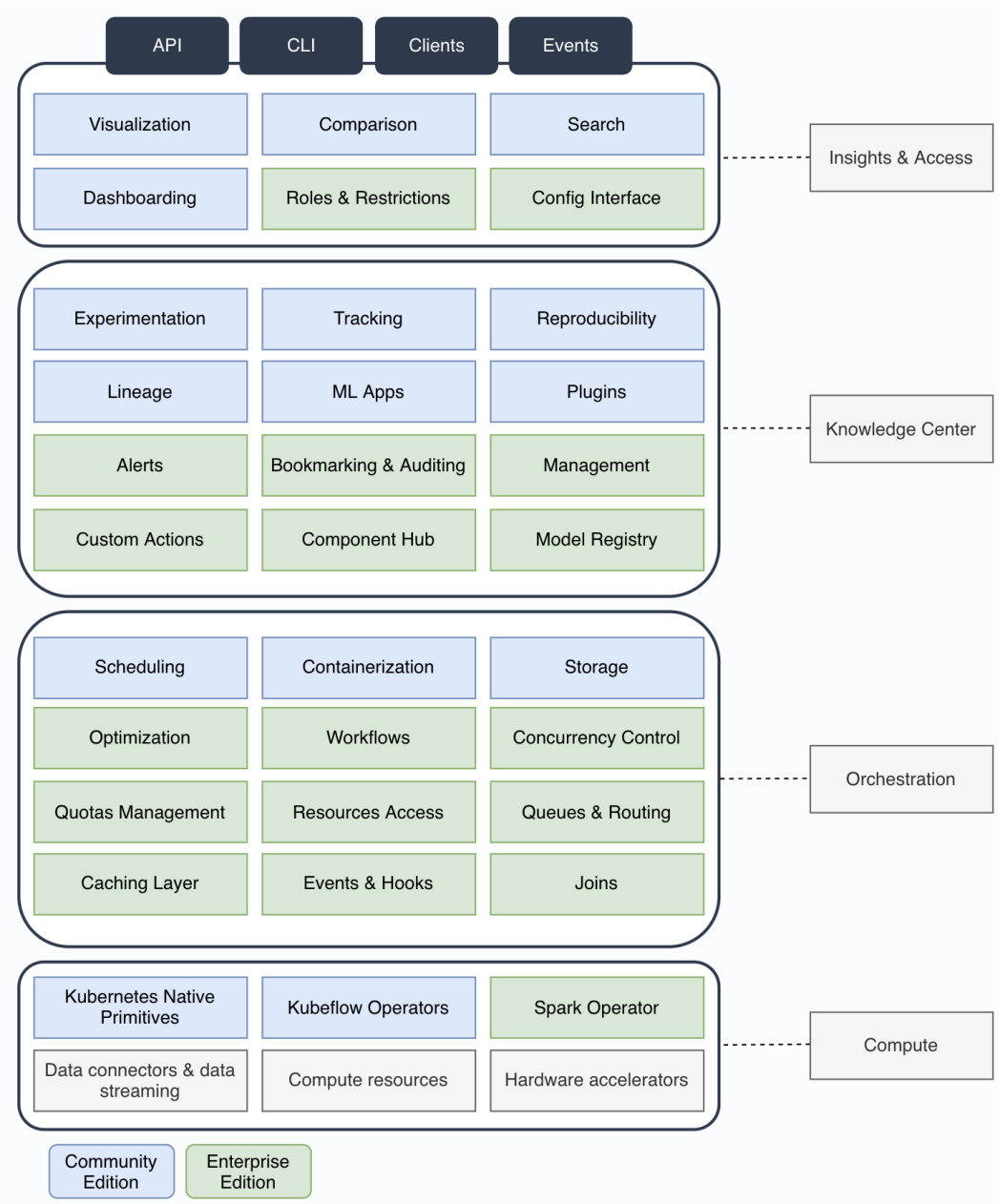



```python
from metaflow import FlowSpec, step

class MyFlow(FlowSpec):

  @step
  def start(self):
    self.next(self.a, self.b)

  @step
  def a(self):
    self.next(self.join)

  @step
  def b(self):
    self.next(self.join)

  @step
  def join(self, inputs):
    self.next(self.end)

MyFlow()
```

https://ychung38.medium.com/use-python-metaflow-to-build-a-perfect-data-science-project-9458ef1b5a30

Polyaxon

- https://polyaxon.com/
- Built on Kubernetes cluster
  - CLI which knows the ML specifics
- Many parts are commercial



Polyaxon

22 lines (20 sloc) | 620 Bytes

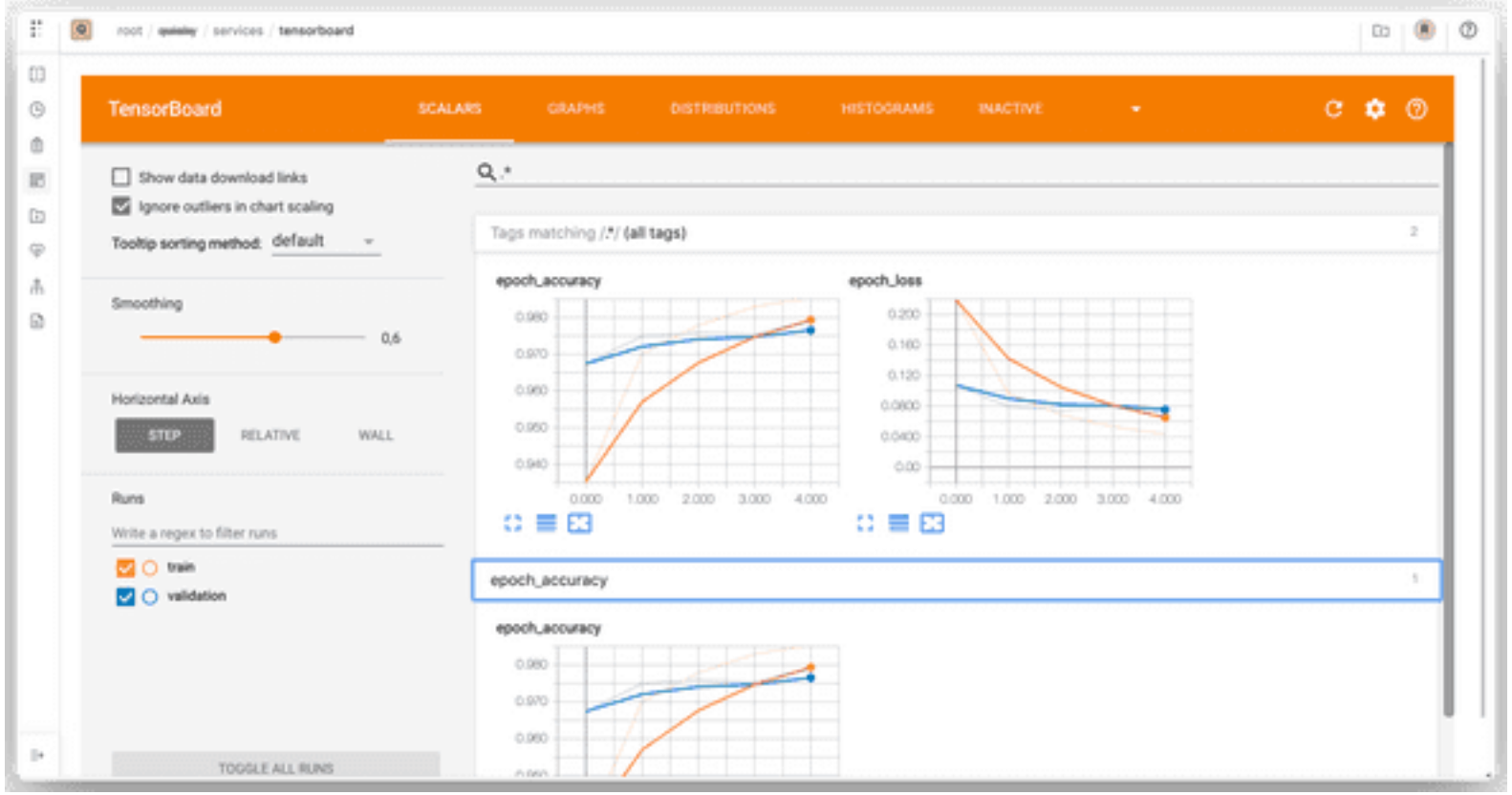
Raw Blame

```yaml
version: 1.1
kind: component
name: iris-classification
tags: ["streamlit", "app"]

inputs:
- name: uuid
  isOptional: true
  type: str

run:
  kind: service
  ports: [8501]
  rewritePath: true
  init:
  - git: {"url": "https://github.com/polyaxon/polyaxon-examples"}
  - artifacts: {"files": ["{{ uuid }}/assets/model/iris-model.joblib"]}
  container:
    image: polyaxon/polyaxon-contrib
    workingDir: "{{ globals.artifacts_path }}/polyaxon-examples/in_cluster/sklearn/iris"
    command: [streamlit, run, app.py]
    args: ["--", "--model-path={{ globals.artifacts_path }}/{{ uuid }}/assets/model/iris-model.joblib"]
```

Polyaxon

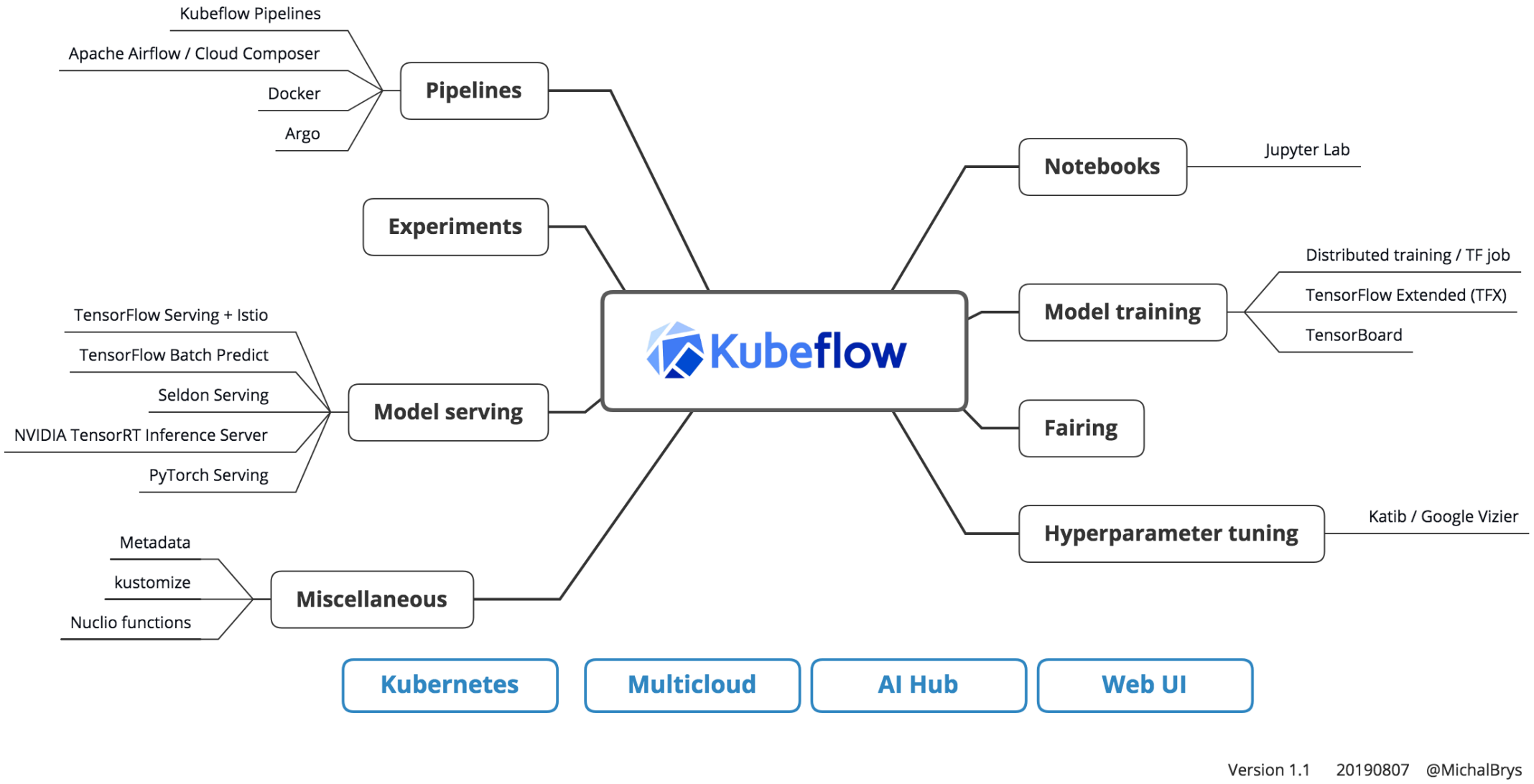
$ polyaxon run --url=https://raw.githubusercontent.com/polyaxon/simple.yaml –l

$ polyaxon run --hub tensorboard:single-run -P uuid=UUID -w



https://polyaxon.com/docs/intro/quick-start/

Kubeflow



https://medium.com/@michal.brys/kubeflow-a-machine-learning-toolkit-for-kubernetes-d8686f6c91b6

Kubeflow

- Comprehensive platform to manage ML workflows
- Aims to deploy ML workflow to Kubernetes
- It’s huge. Requires extensive administration

Recap. A typical workflow

- Prototype at Jupyter notebook. It's a good way to start
- Move it to a server
  - To share it or to get powerful hardware
- Convert to code
  - To extract reusable parts or to compose a library
- Visualize it
  - Comet.ml, WandB, DashPlotly
- Express pipeline to DAG
  - To use Kedro, Metaflow, etc.
- Deliver to production cluster

Conclusion. A typical workflow

- **Reproducibility**
  - **Deliver** code, data, model and results to end user is **important**
  - There are plenty of software to achieve this

Thank you!

Interested in collaboration?

alex_myltsev

hse_lambda

Alexander Myltsev