Data Versioning
Towards Reproducibility in Machine Learning

Nicolás Eiris
Machine Learning Engineer
Tryolabs

Tryolabs

- We build custom **AI solutions**
- **70+** team members
- **12+ years** of experience
- Served more than **150 clients**

Trusted by

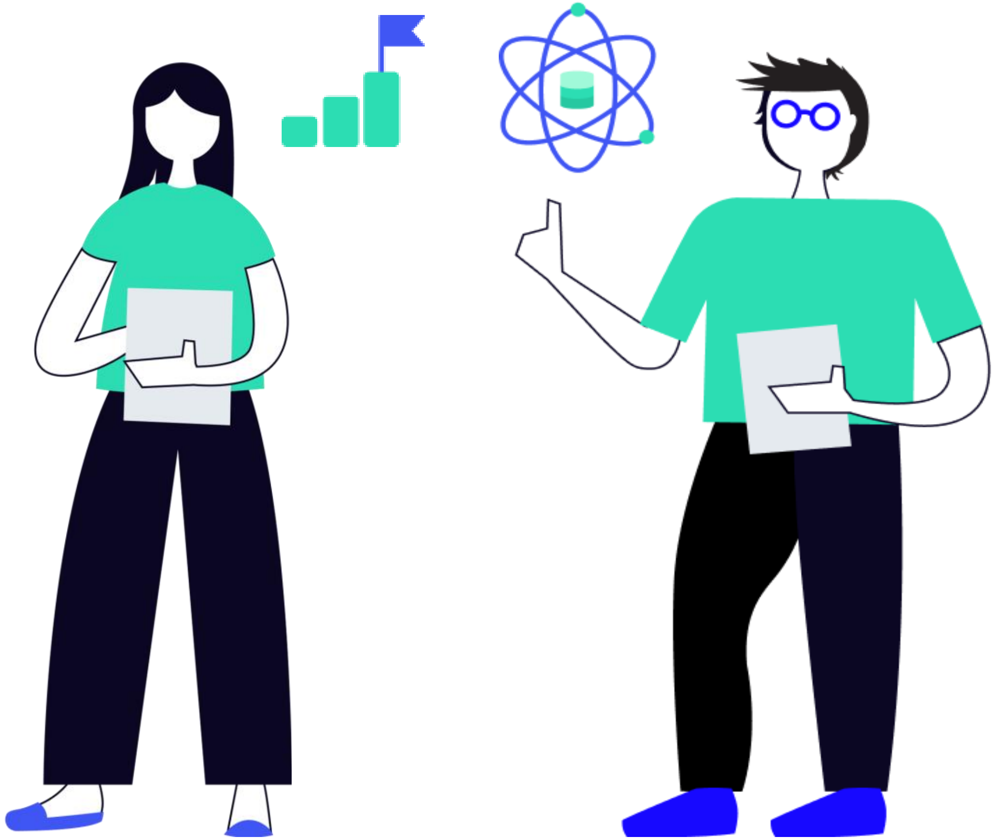
TheRealReal
Allianz Global Investors
GRUBHUB
CFL
HALLIBURTON
SES

Agenda

1. **Main pain points in ML workflows**
2. **Useful open source tool**
3. **Takeaways**
4. **References**

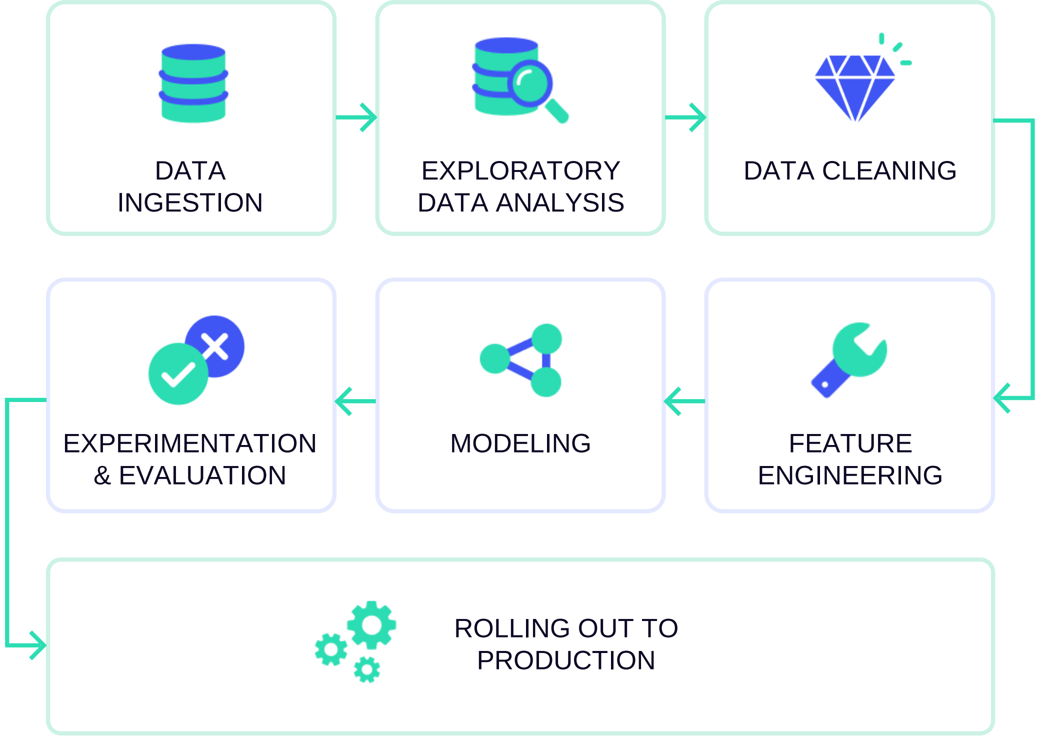
Dilemma in ML development

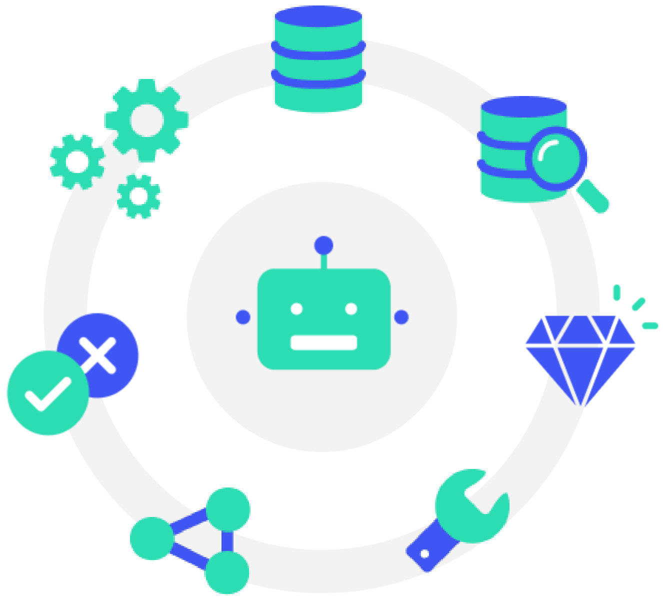
Building everything **manually** from scratch vs. using a **tool to support** the development phase (from collecting data to deploying on the edge).



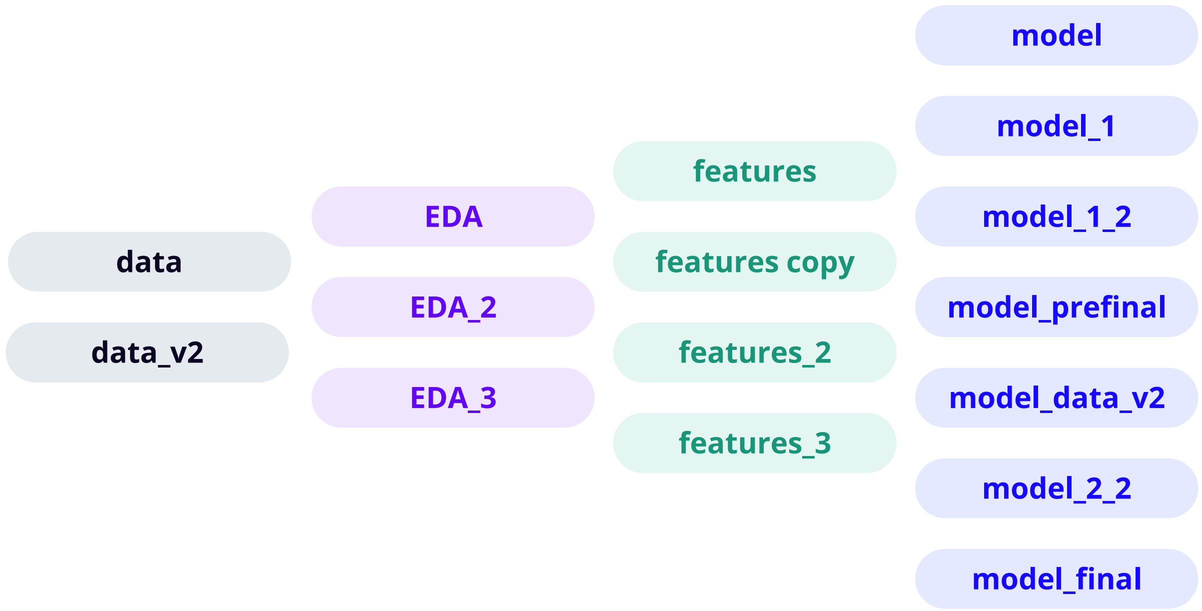
Main pain points in ML workflows

Standard ML workflow



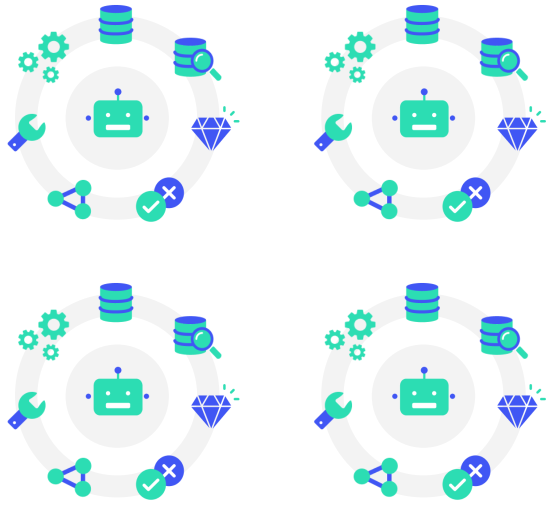


ML pipeline in practice



*EDA = Exploratory data analysis

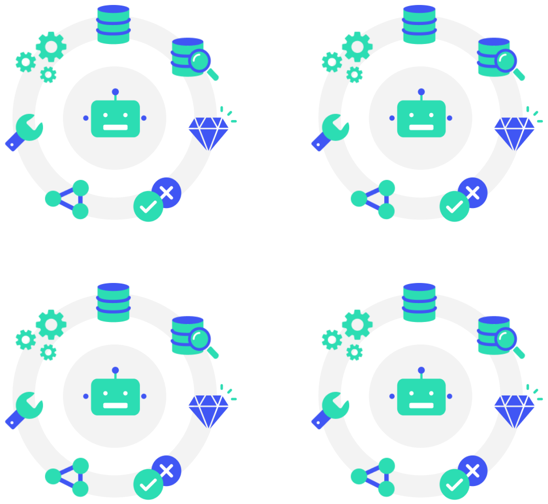
Main pain points in ML workflows



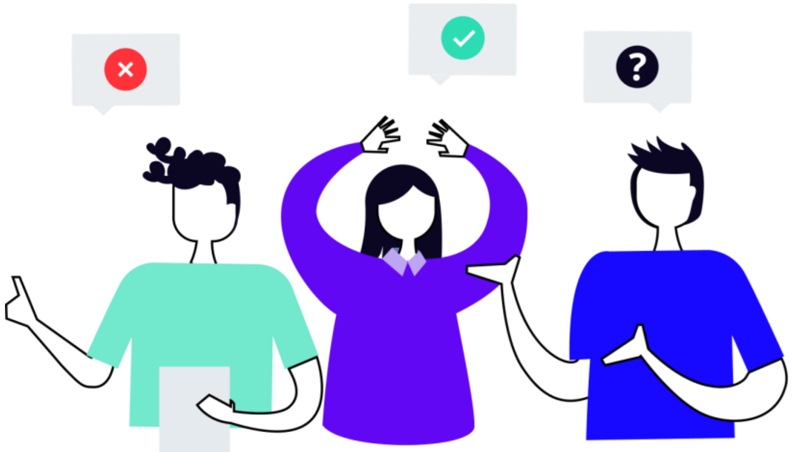
1. Reproducibility

- Teamwork
- Usually ad-hoc processes
- Productivity bottleneck
- Challenges
  - Changes in data
  - Hyperparams inconsistency
  - Randomness
  - Manual and ad-hoc execution of experiments

Main pain points in ML workflows

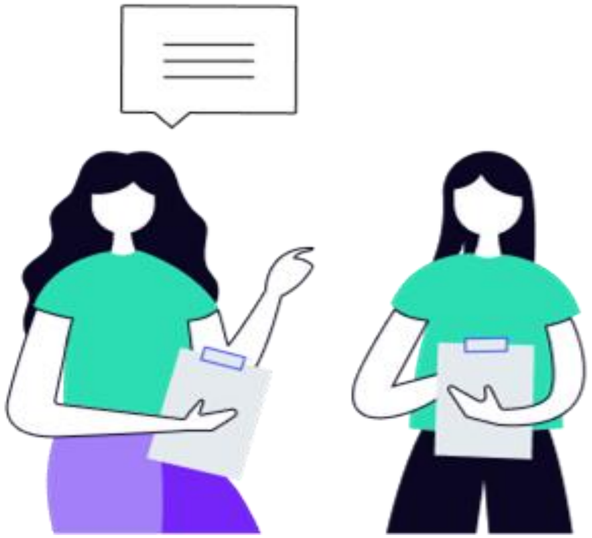


1. Reproducibility



"Changes are uploaded, please run all the notebook again."

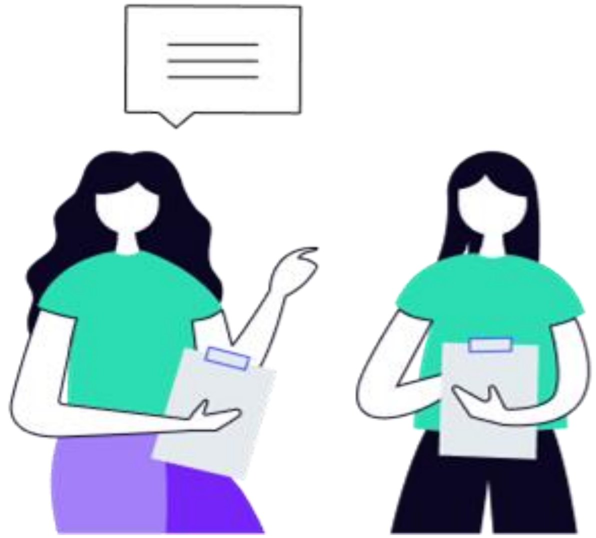
Main pain points in ML workflows



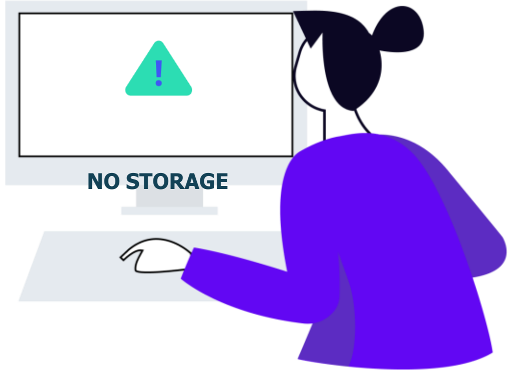
2. Data sharing

- Complex READMEs on how to gather data from remote storage
- Security and data privacy risks
- Manual versioning of dataset changes

Main pain points in ML workflows



2. Data sharing



"I wish I could automate this process…"

Main pain points in ML workflows

3. Experiments execution & tracking

- Experiments setup traceability challenges
- Inefficient results comparison & evaluation
- Manual process:
  - Spreadsheet
  - Github (metadata files)
  - Tracking tools (big learning curve)

Ideal development experience

Structured pipeline composed by interdependent steps

Easily **adding files or directories** to a remote repository

Sharing experiments, models, and results in a simple way

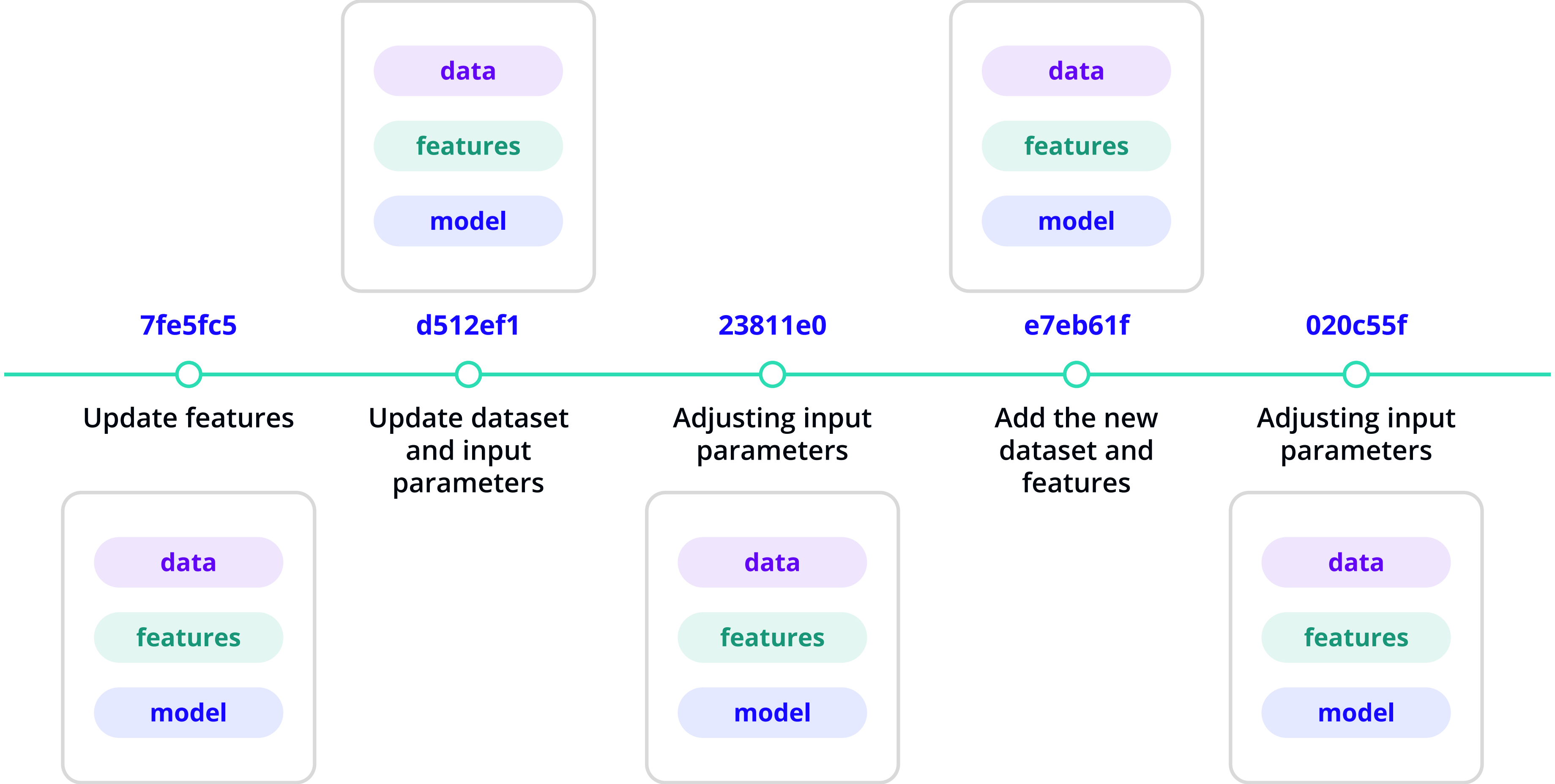
Stop worrying about source code and data association

Useful open source tool:

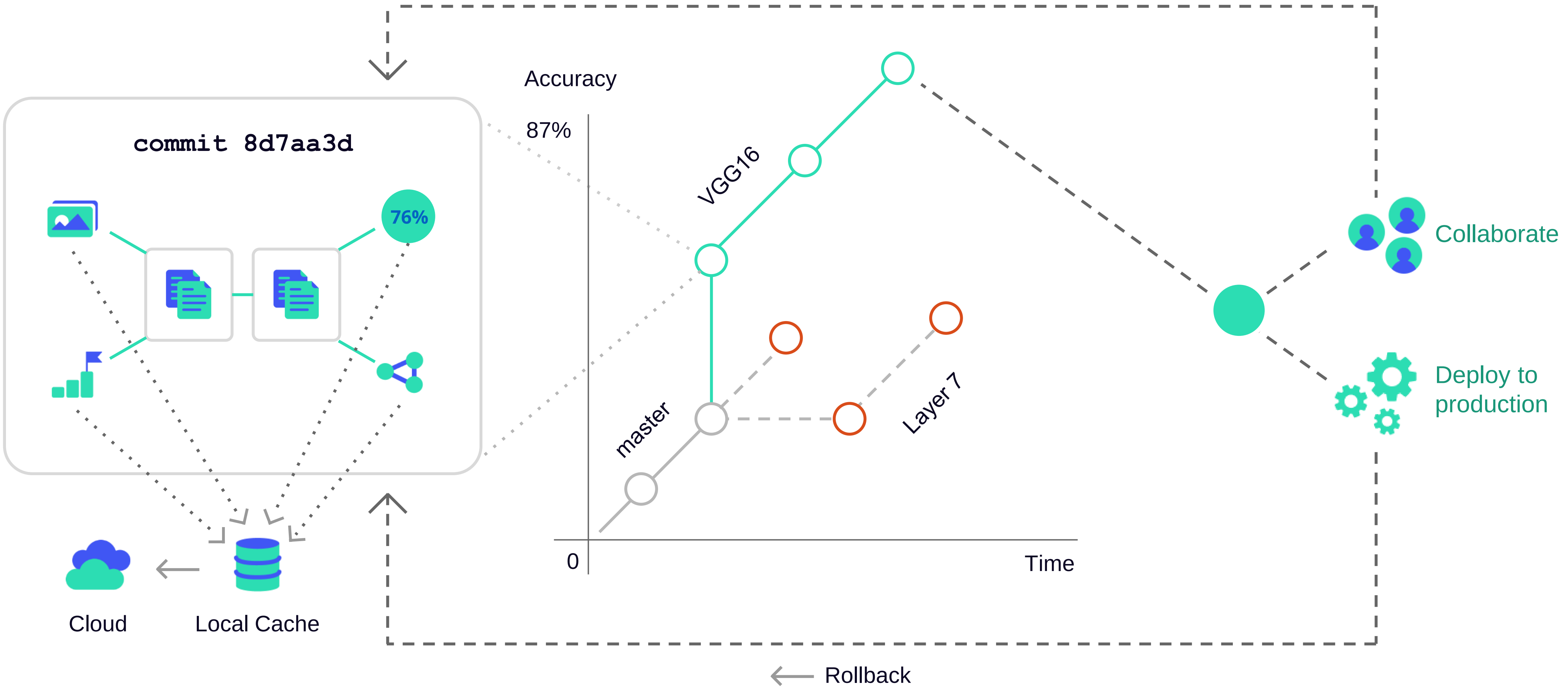
DATA VERSION CONTROL

tryo•labs

DVC high-level overview



DVC high-level overview



Main features

- Git-compatible
- Storage agnostic
- Reproducible
- Low friction branching
- ML pipeline framework
- Language & framework agnostic
- Track failures
- Experiments & metrics tracking

Pipelines

- **Pipelines** composed by interdependent **steps**
  - Dependencies
  - Code to execute
  - Outputs

- Additional pipeline **visualization** command `dvc dag`

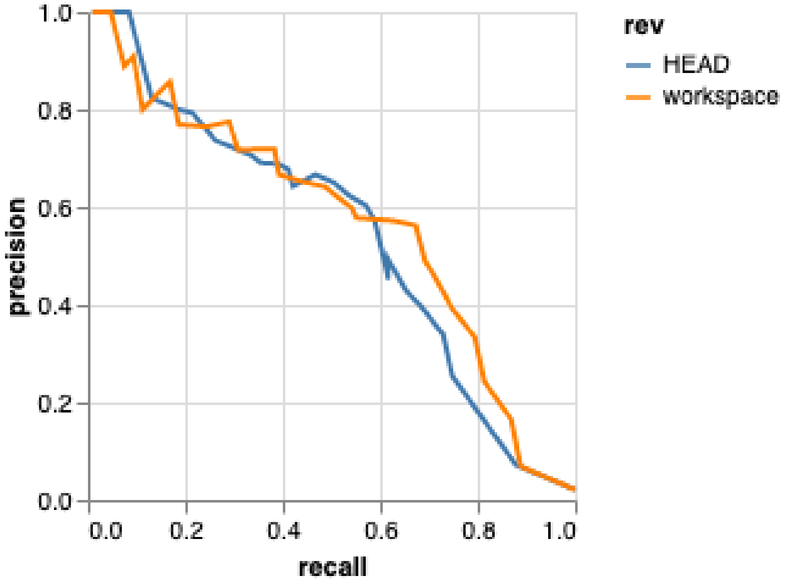
```
$ dvc dag
         +---------+
         | prepare |
         +---------+
              *
              *
              *
        +-----------+
        | featurize |
        +-----------+
          **       **
        **           *
       *              **
+-------+               *
| train |              **
+-------+             *
          **       **
            **   **
              * *
        +----------+
        | evaluate |
        +----------+
```

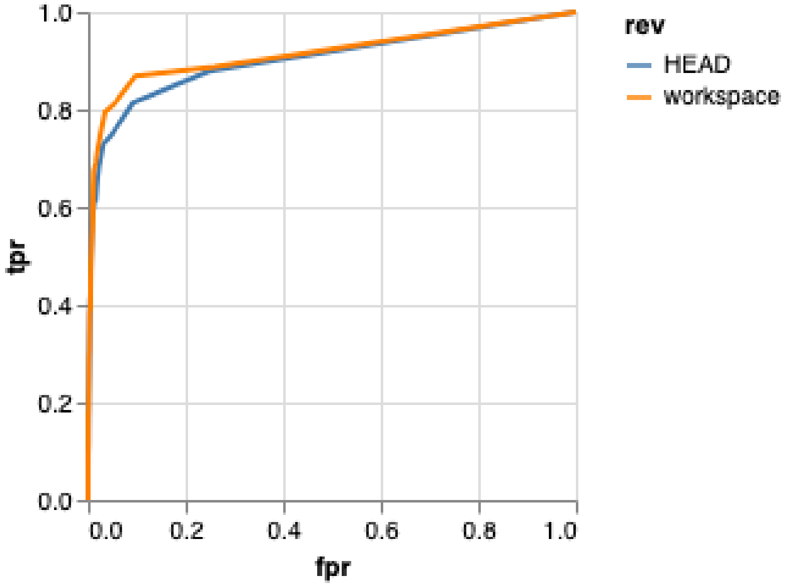
```
stages:
  build:
    cmd: python train.py
    deps:
      - features.csv
    outs:
      - model.pt
    metrics:
      - accuracy.txt:
          cache: false
    plots:
      - auc.json:
          cache: false
```

Metrics differences

Smooth comparison process:
numeric and **graphic** visualization

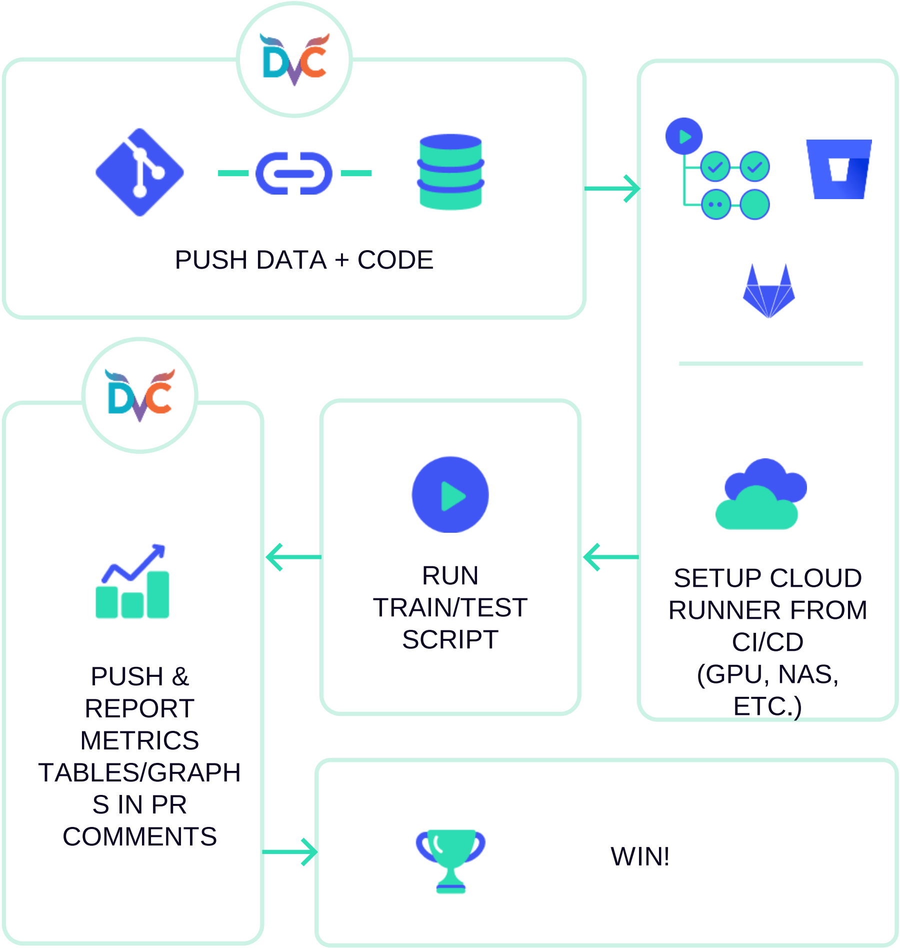
```
$ dvc metrics diff
Path         Metric     HEAD     workspace  Change
scores.json  avg_prec   0.52048  0.55259    0.03211
scores.json  roc_auc    0.9032   0.91536    0.01216
```





Continuous integration

- Automatically check data version
- Benchmark new model against previously deployed models
- Metrics diff & interactive plots in Pull Requests
- Re-train & refine in the cloud



SOURCE: WWW.DVC.COM

Experiments batch execution

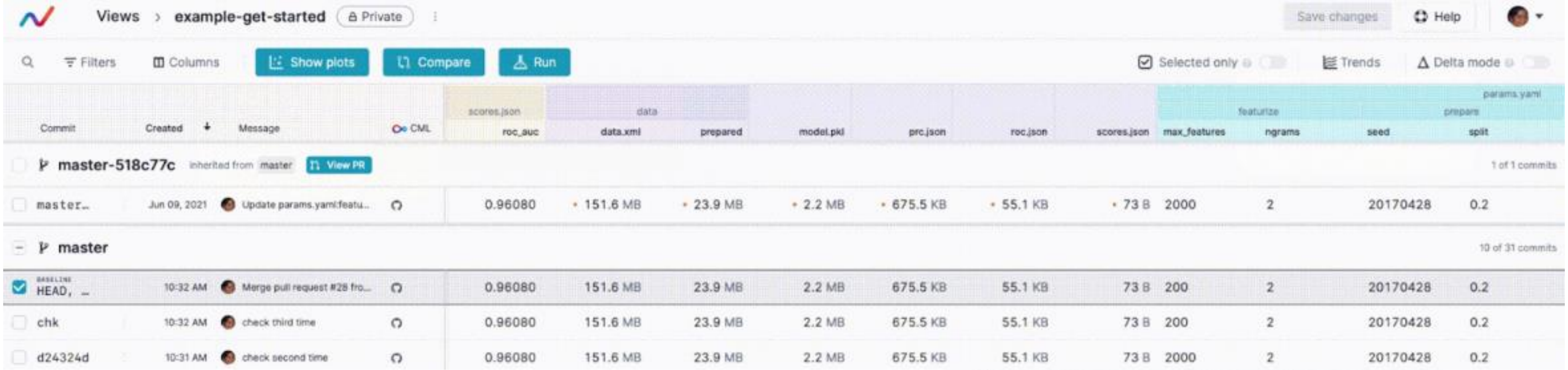
Experiment	Created	train	test	model.n_estimators	model.max_depth	model.min_samples_split	model.min_samples_leaf	model.max_leaf_nodes	model.random_state
workspace	-	96.257	70.404	100	20	2	1	-	42
dvc	02:59 PM	96.257	71.3	100	-	2	1	-	42
├── 20d798f [max_depth_20]	03:49 PM	96.257	70.404	100	20	2	1	-	42
├── 6d9edfa [max_depth_5]	03:49 PM	76.946	74.439	100	5	2	1	-	42
├── ac459ee [max_depth_1]	03:49 PM	68.263	71.749	100	1	2	1	-	42
└── a7acf28 [max_depth_2]	03:48 PM	71.557	76.233	100	2	2	1	-	42

"I can't believe the number of hours saved by queuing and executing experiments in parallel."

tryo·labs

UI does not have to be built from scratch

- Show plots for selected experiments



- Generate trend charts

SOURCE: WWW.DVC.COM

Takeaways

Takeaways

Adopting a **development support tool** across the entire ML workflow may be crucial for the **success** of a project.

Stop reinventing the wheel for **common ML challenges**.

Boost developer's productivity by enabling them to focus on coding.

Integrating DVC tool **favors quality attributes** such as maintainability, scalability, and security.

Support **end-to-end experience**, from EDA to production.

Takeaways

Reproducibility

With a couple of commands, **replicate the environment state** from other team members (without re-executing all the pipeline or experiment).

Experiments

Quickly run multiple **experiments in parallel** with various ways of visualizing and comparing results.

Data sharing

Data and source code association out-of-the-box, with a wide variety of remote storage options.

Takeaways - tool vs. from scratch

We learned that for most of the cases, using an all-in-one framework **like DVC** alleviates the work **vs. manually dealing** with Reproducibility, Experimentation, and Data sharing **tasks.**

Resources

DVC documentation

https://dvc.org/doc

Platform to quickly get-started with DVC
https://katacoda.com/dvc/courses/get-started

Norfair - Tryolabs object tracking open-source library

https://github.com/tryolabs/norfair

Reproducibility in machine learning

https://towardsdatascience.com/reproducible-machine-learning-cf1841606805

Thank you!

tryo•labs