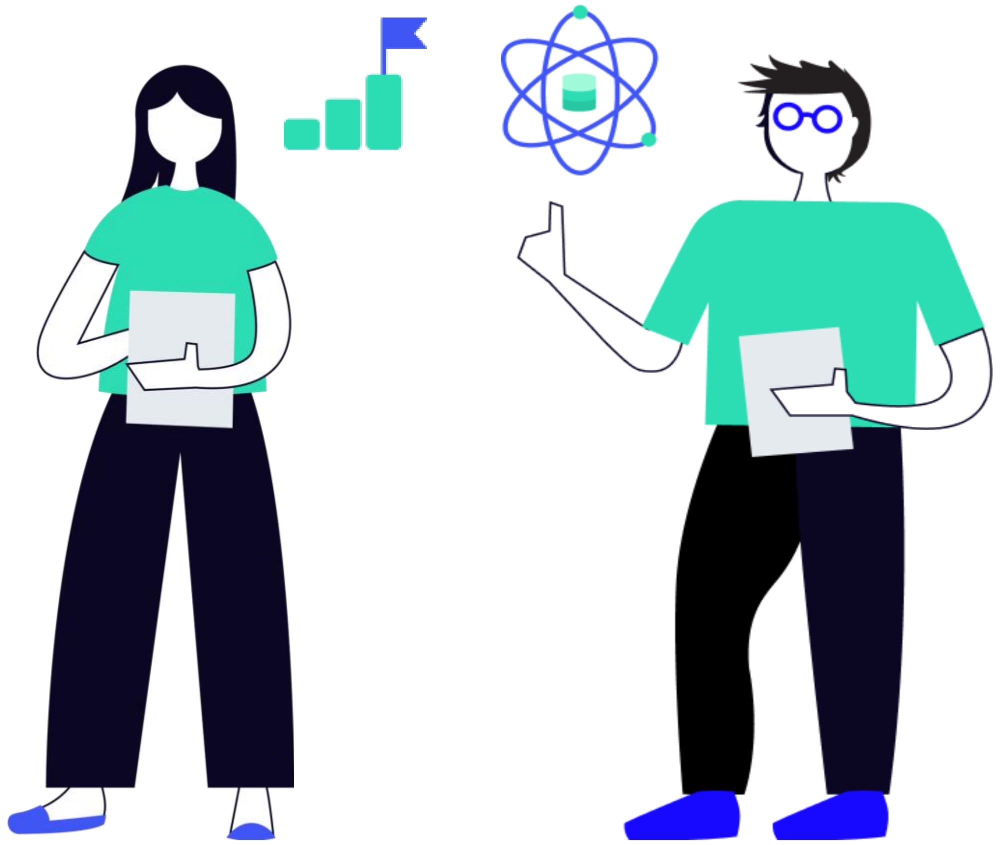
Dilemma in ML development

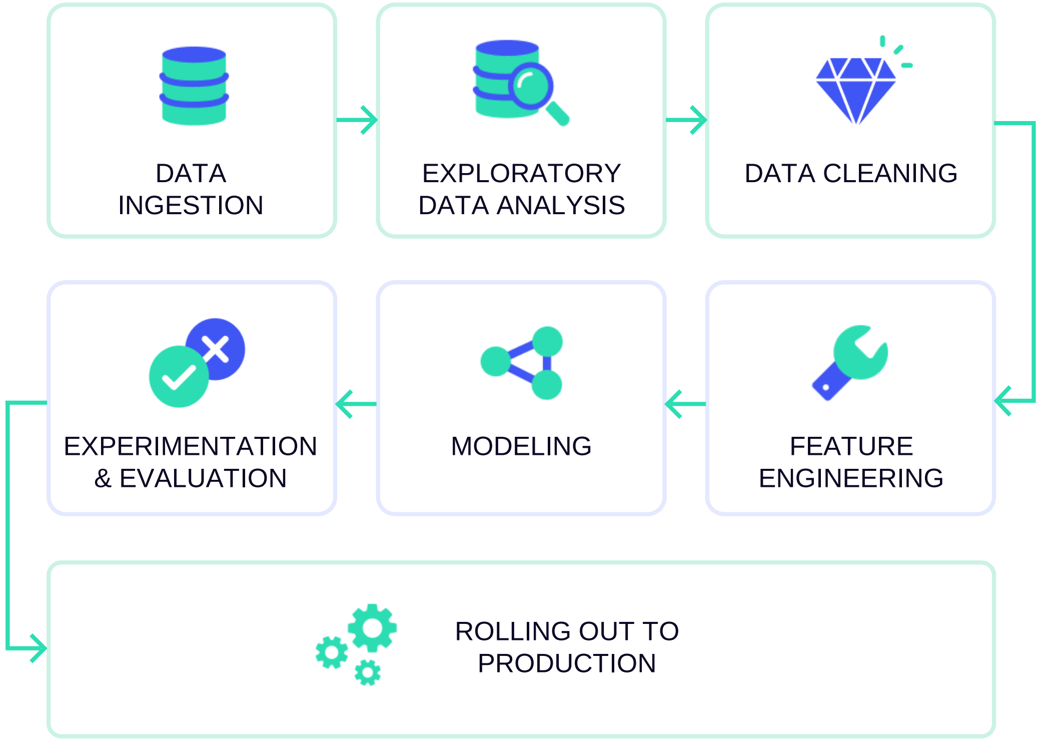
Building everything **manually** from scratch vs. using a **tool to support** the development phase (from collecting data to deploying on the edge).



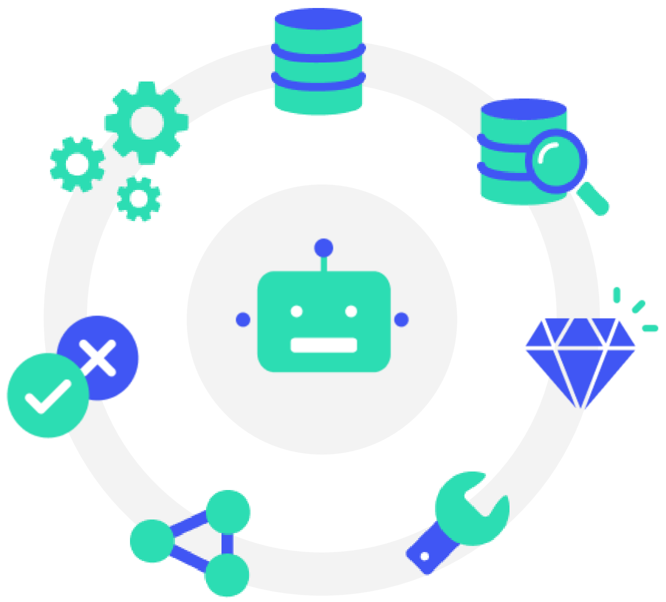
Main pain points in ML workflows

tryo·labs

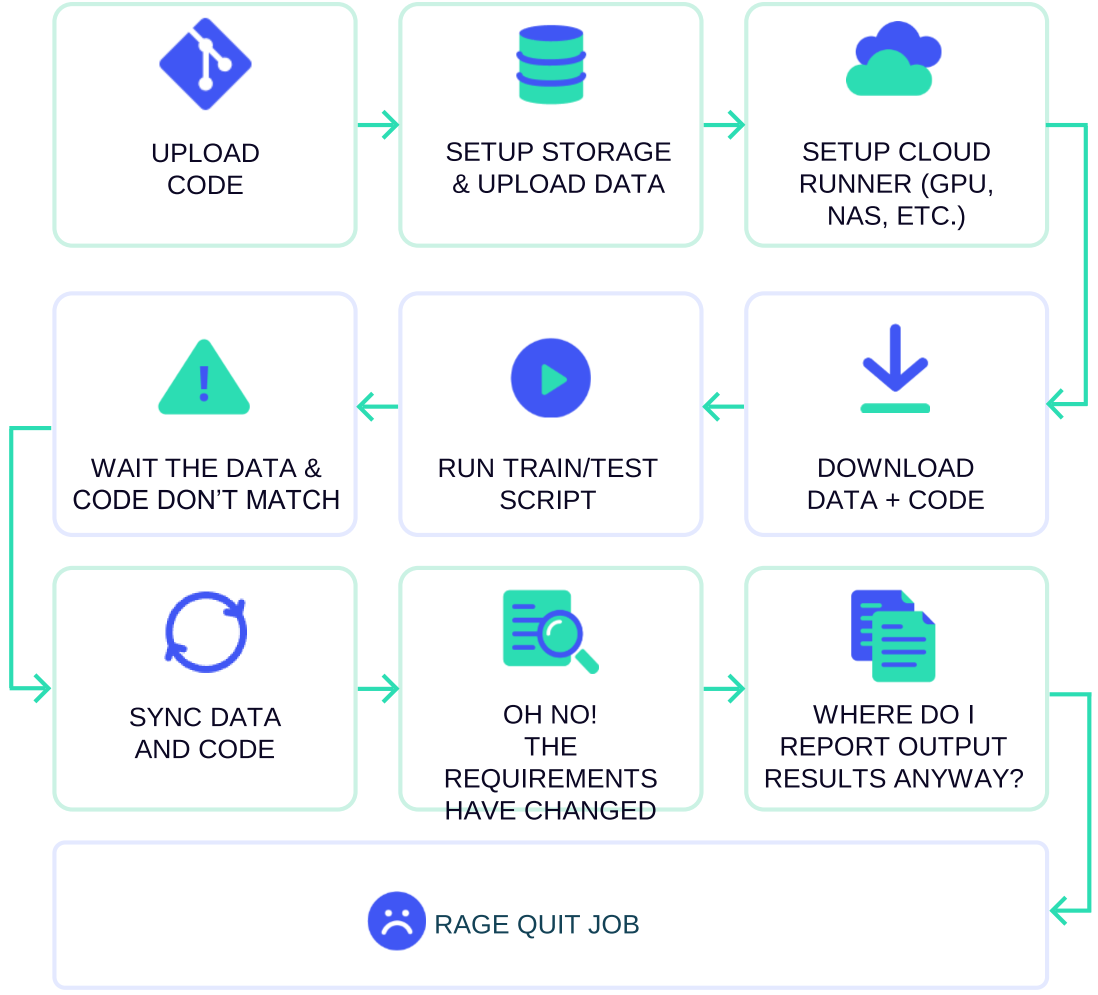
Standard ML workflow

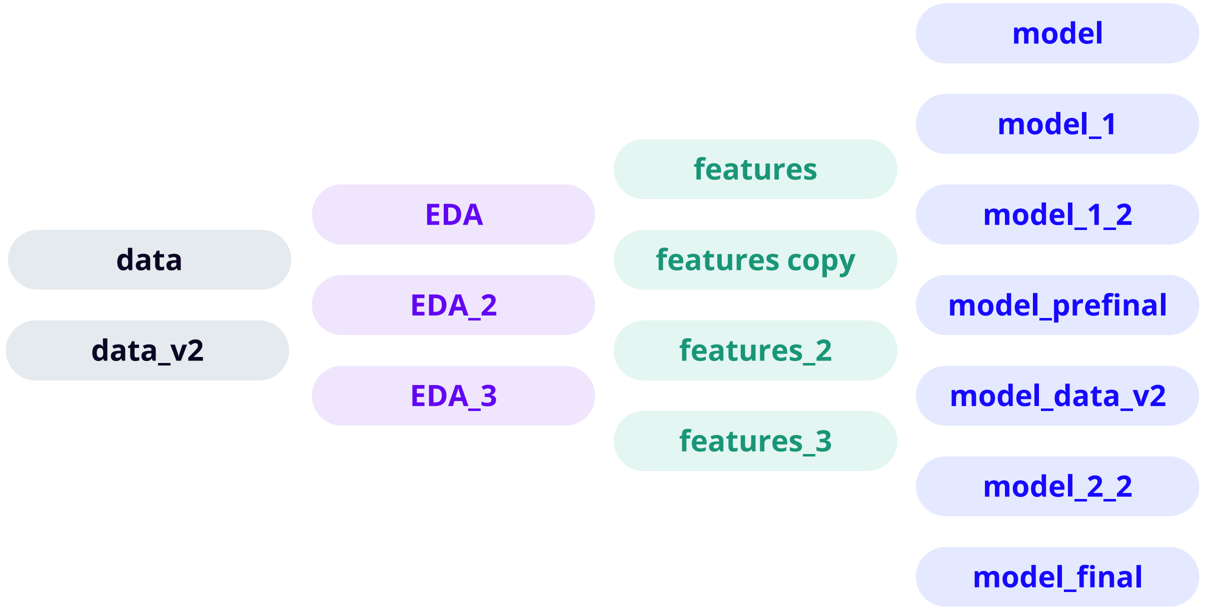


DATA INGESTION
EXPLORATORY DATA ANALYSIS
DATA CLEANING
FEATURE ENGINEERING
MODELING
EXPERIMENTATION & EVALUATION
ROLLING OUT TO PRODUCTION



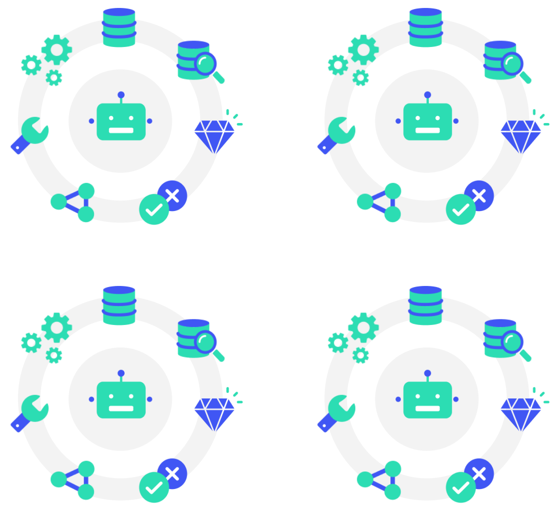
ML pipeline in practice


UPLOAD CODE
SETUP STORAGE & UPLOAD DATA
SETUP CLOUD RUNNER (GPU, NAS, ETC.)
DOWNLOAD DATA + CODE
RUN TRAIN/TEST SCRIPT
WAIT THE DATA & CODE DON'T MATCH
SYNC DATA AND CODE
OH NO! THE REQUIREMENTS HAVE CHANGED
WHERE DO I REPORT OUTPUT RESULTS ANYWAY?
RAGE QUIT JOB



*EDA = Exploratory data analysis

Main pain points in ML workflows

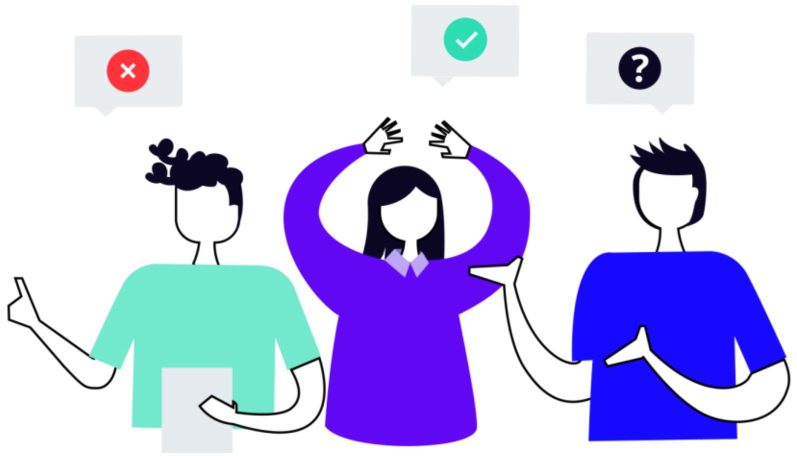


1. Reproducibility

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

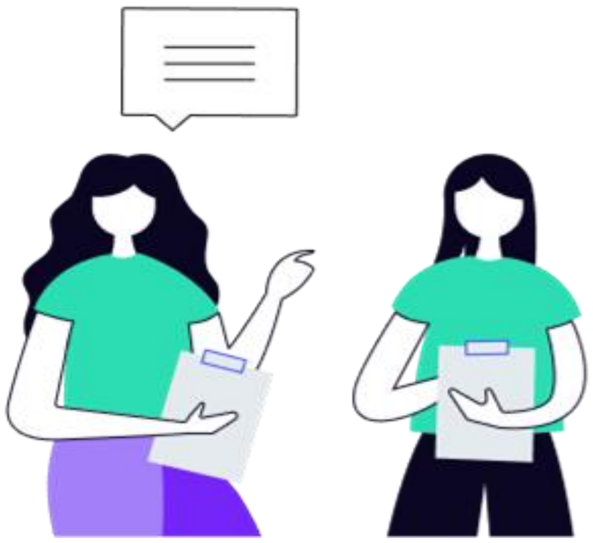
Main pain points in ML workflows

1. Reproducibility



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

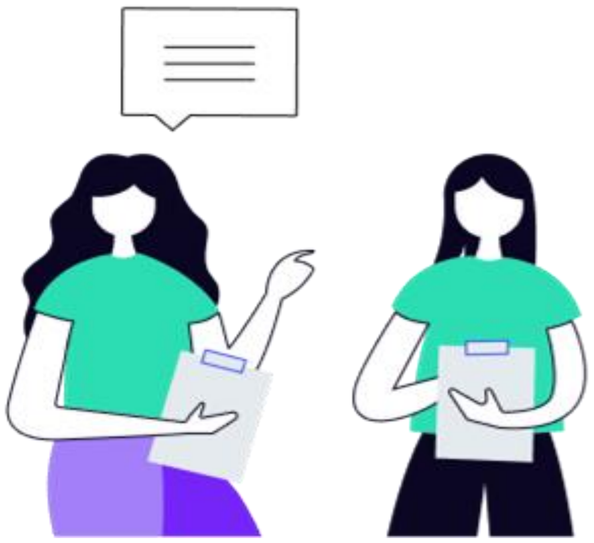
Main pain points in ML workflows



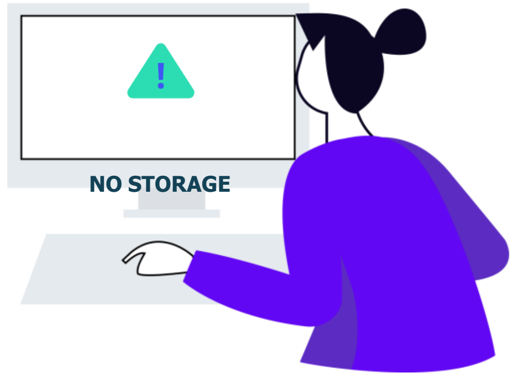
2. Data sharing

- 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

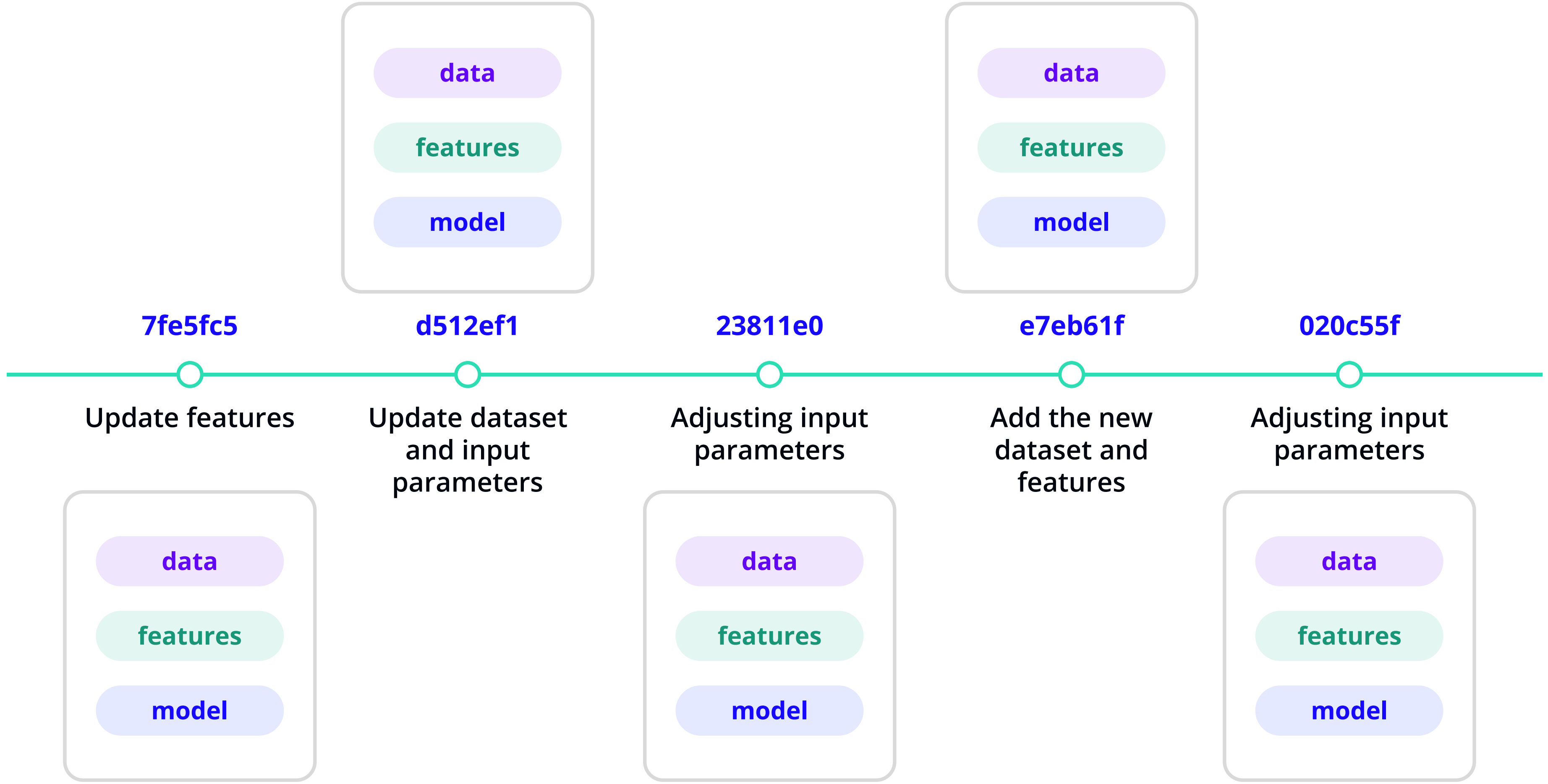
Sharing experiments, models, and results in a simple way

Stop worrying about source code and data association

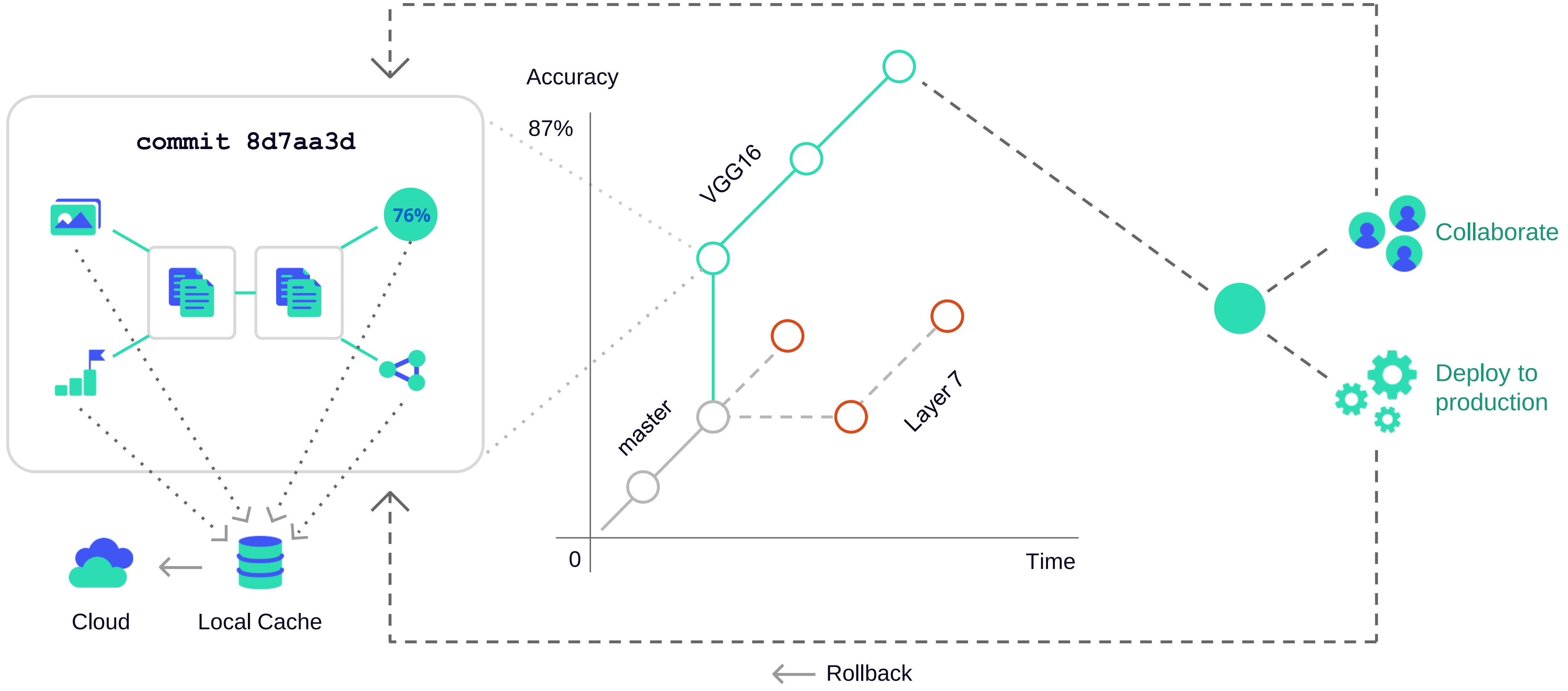
Useful open source tool:

DATA VERSION CONTROL

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`

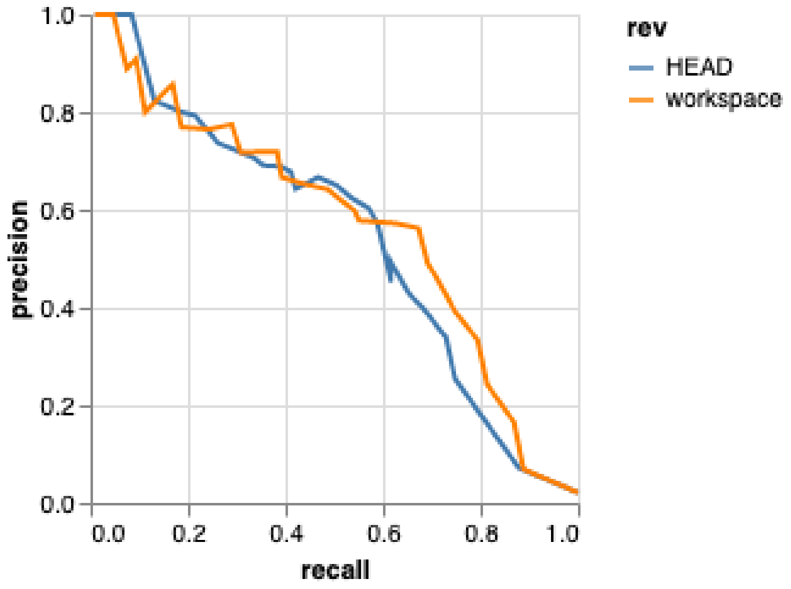
```
$ dvc dag
         +---------+
         | prepare |
         +---------+
              *
              *
              *
        +-----------+
        | featurize |
        +-----------+
         **        **
       **            *
      *               **
+-------+               *
| train |    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

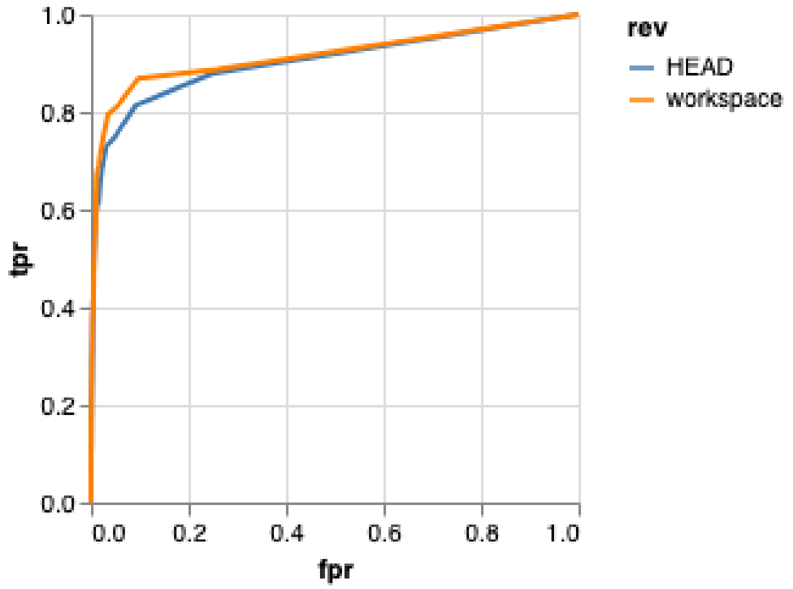
```
stages:
  build:
    cmd: python train.py
    deps:
      - features.csv
    outs:
      - model.pt
    metrics:
      - accuracy.txt:
          cache: false
    plots:
      - auc.json:
          cache: false
```

Metrics differences

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

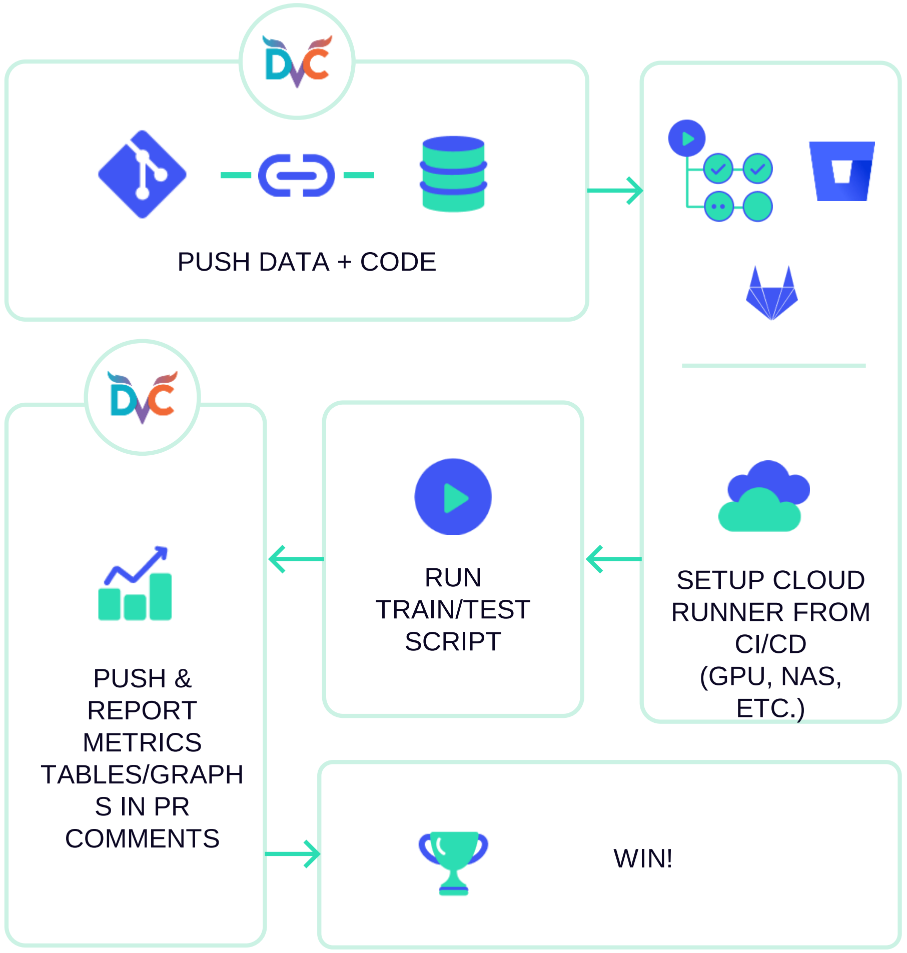
```
$ dvc metrics diff
Path         Metric     HEAD     workspace  Change
scores.json  avg_prec   0.52048  0.55259    0.03211
scores.json  roc_auc    0.9032   0.91536    0.01216
```





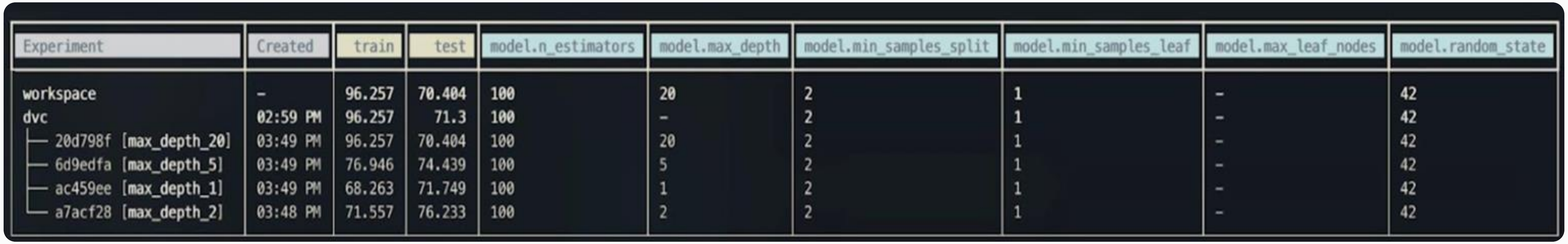
Continuous integration

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


PUSH DATA + CODE
SETUP CLOUD RUNNER FROM CI/CD (GPU, NAS, ETC.)
RUN TRAIN/TEST SCRIPT
PUSH & REPORT METRICS TABLES/GRAPHS IN PR COMMENTS
WIN!

SOURCE: WWW.DVC.COM

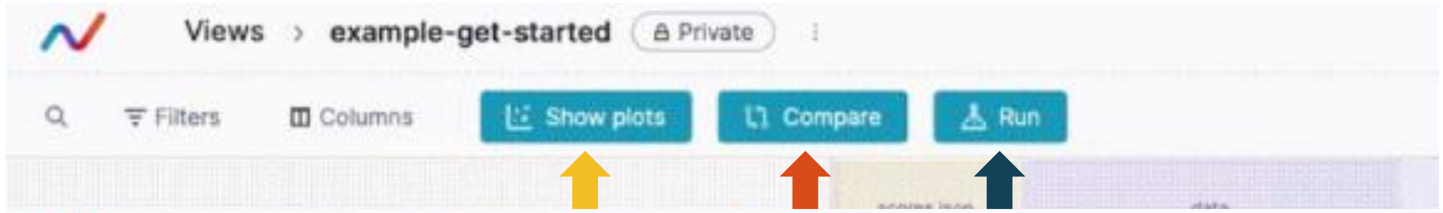
Experiments batch execution



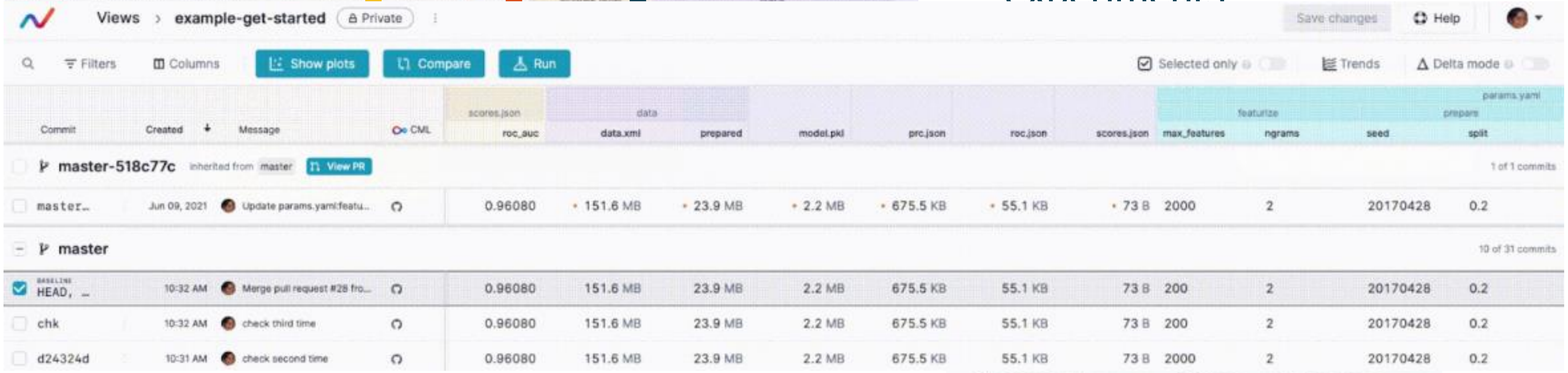
```
 Experiment                      Created    train    test    model.n_estimators  model.max_depth  model.min_samples_split  model.min_samples_leaf  model.max_leaf_nodes  model.random_state

 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."

UI does not have to be built from scratch

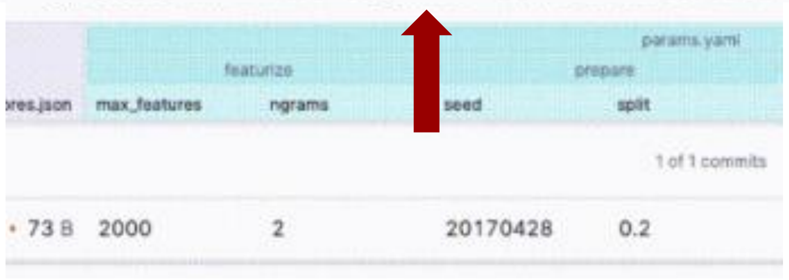


- Show plots for selected experiments



				scores.json	data						params.yaml featurize		params.yaml prepare	
Commit	Created	Message	CML	roc_auc	data.xml	prepared	model.pkl	prc.json	roc.json	scores.json	max_features	ngrams	seed	split
master-518c77c inherited from master View PR														1 of 1 commits
master…	Jun 09, 2021	Update params.yaml:featu…		0.96080	151.6 MB	23.9 MB	2.2 MB	675.5 KB	55.1 KB	73 B	2000	2	20170428	0.2
master														10 of 31 commits
BASELINE HEAD, …	10:32 AM	Merge pull request #28 fro…		0.96080	151.6 MB	23.9 MB	2.2 MB	675.5 KB	55.1 KB	73 B	200	2	20170428	0.2
chk	10:32 AM	check third time		0.96080	151.6 MB	23.9 MB	2.2 MB	675.5 KB	55.1 KB	73 B	200	2	20170428	0.2
d24324d	10:31 AM	check second time		0.96080	151.6 MB	23.9 MB	2.2 MB	675.5 KB	55.1 KB	73 B	2000	2	20170428	0.2

- Generate trend charts



	params.yaml featurize		params.yaml prepare	
…res.json	max_features	ngrams	seed	split
				1 of 1 commits
73 B	2000	2	20170428	0.2

SOURCE: WWW.DVC.COM 22