DVC: data versioning and ML experiments on top of Git

Dmitry Petrov, PhD

@FullStackML

dmitry@iterative.ai

Hello

Dmitry Petrov

PhD in Computer Science
Twitter: @FullStackML

Creator of
DVC.org project

Co-Founder & CEO > Iterative.AI > San Francisco, USA

ex-Data Scientist > Microsoft (BingAds) > Seattle, USA

ex-Head of Lab > St. Petersburg Electrotechnical University > Russia

1

DVC principles

DVC principle: MLOps is a part DevOps

1. Automation: command line first
2. Compatible with Git (not a replacement)
3. Compatible with CI/CD

What DVC does?

1. Data versioning
2. ML models versioning
3. ML pipeline versioning
4. Experiment tracking
    a. Model checkpoints

DVC principles

1. Git is a foundation
2. Use storage directly – speed!
    - S3, Azure Blob, GCS, GDrive, FTP
3. No services
    - Git server (like GitHub) + S3
    - Distributed (as Git)
4. Compatible with Git and Git-ecosystem
    - GitHub/GitLab/BitBucket issue trackers
    - CI/CD

DVC users

1. ML researcher & ML engineer
    - Create ML models
    - Tune models
2. DevOps & Engineers
    - Productionize models
    - Manage data and labels

DVC first step

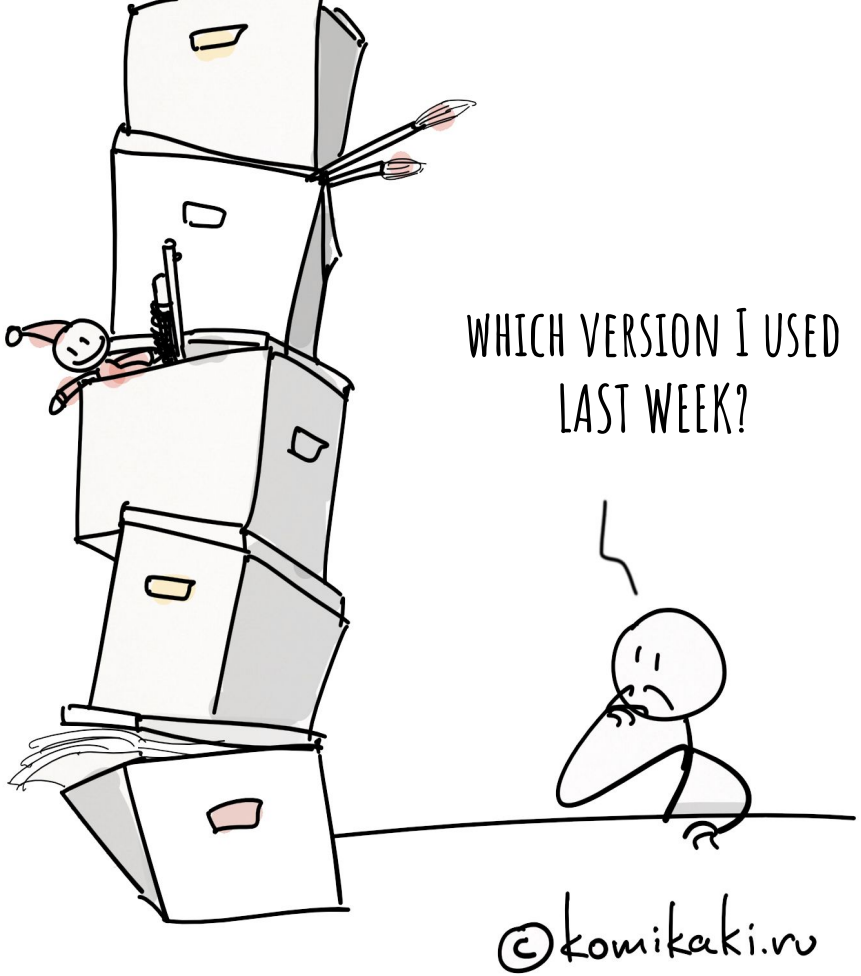
WebSite and Docs: **https://dvc.org**

Source code: **https://github.com/iterative/dvc/**

```
$ pip install dvc


# Init in a project
$ cd my-project
$ dvc init
```

Define storage - data remote

```
$ dvc remote add -d mys3 s3://dmpetrov/test
$ git add .dvc/config
```



2

Data versioning

Add data to your project

```
$ dvc add data/   # creates mata-file data.dvc

$ cat data.dvc
outs:
- md5: b8f4d5a78e55e88906d5f4aeaf43802e.dir
  size: 41149064
  nfiles: 1800
  path: data
```

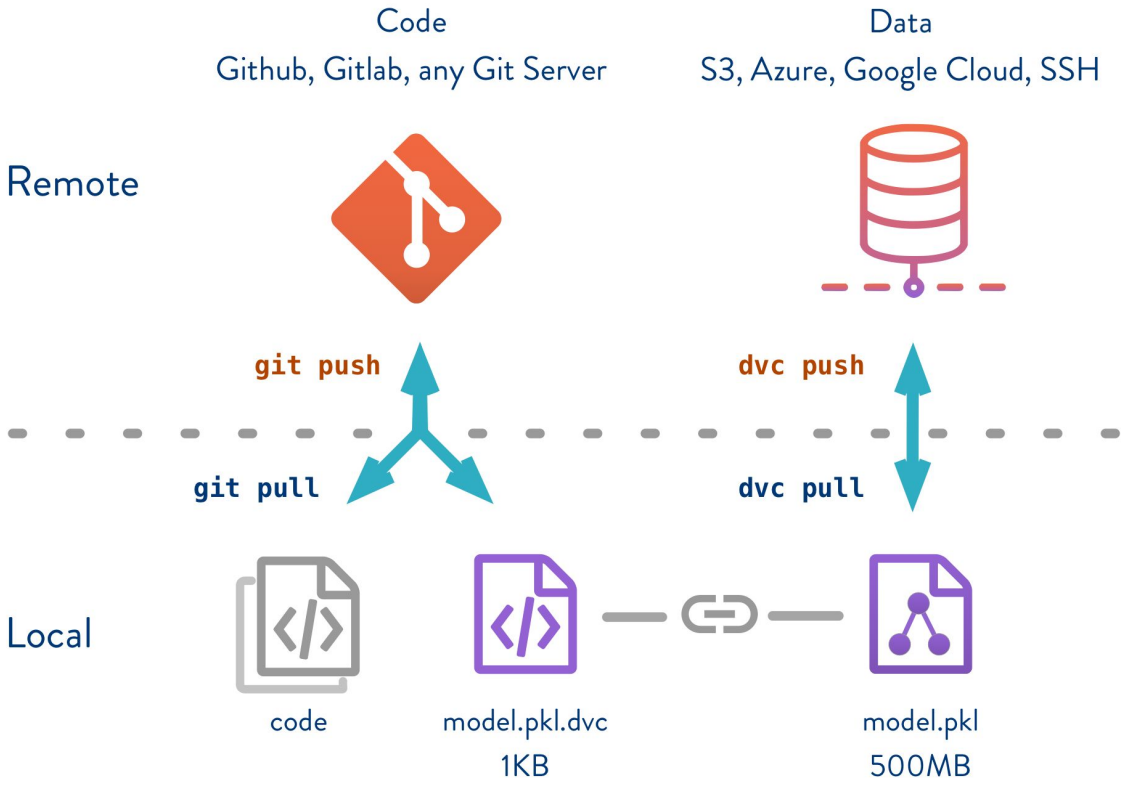
Commit meta-data

```
$ cat data.dvc
outs:
- md5: b8f4d5a78e55e88906d5f4aeaf43802e.dir
  size: 41149064
  nfiles: 1800
  path: data

$ git add data.dvc
$ git commit -m “1st dataset”
$ dvc push  # push dataset to the storage
```

Transfer projects

```
$ git clone https://github.com/dmpetrov/test
$ cd test
$ ls
…
data.dvc

$ dvc pull            # transfers data
$ du -sh data
43M       data.tsv
```

DVC: data meta-info



3

ML pipeline versioning

Define pipeline and metrics file

```
$ dvc run -n my_train          \
      -d data/ -d train.py     \   # dependencies
      -o model.h5              \   # model file
      -M metrics.json          \   # model metrics file
      python train.py          \   # command to run

$ cat metrics.json
{"step": 2, "loss": 2.142695903778076,
 "accuracy": 0.2135166748523712,
  "val_loss": 2.102356910705564,
  "val_accuracy": 0.25060001015663147
}
```

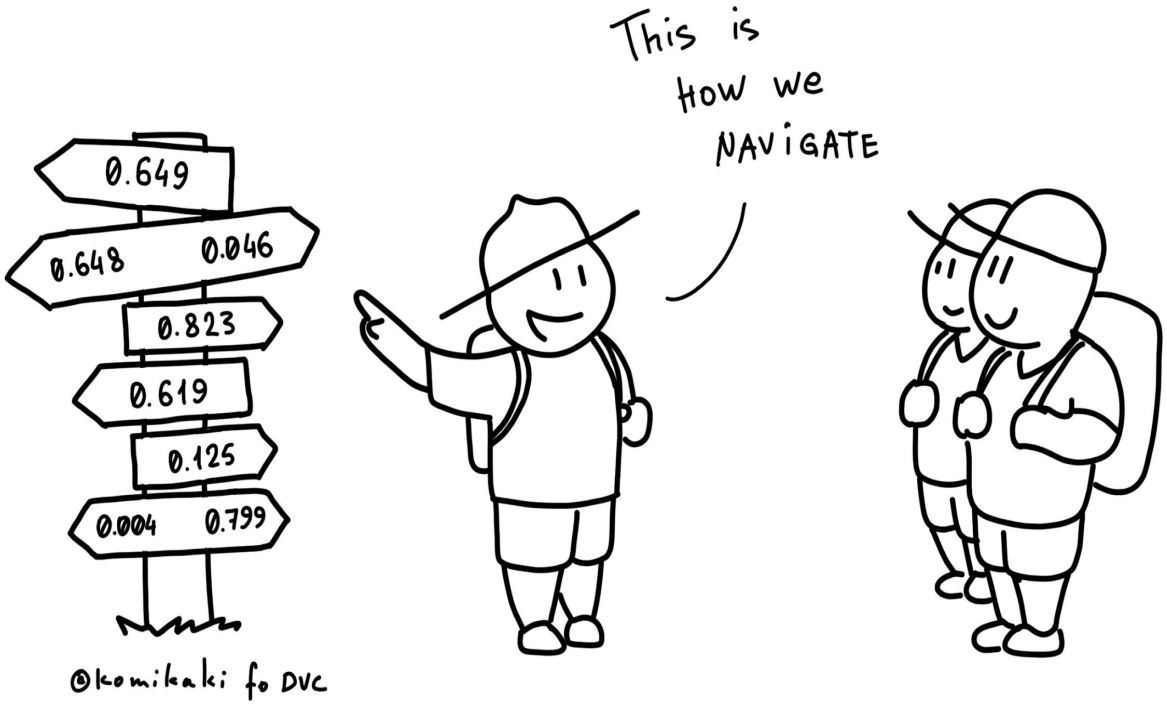
Pipeline meta-file

```
$ cat dvc.yaml
stages:
  my_train:
    cmd: python train.py
    deps:
    - data
    - train.py
    outs:
    - model.h5
    metrics:
    - metrics.json:
        cache: false
```

Reproduce pipeline

```
$ dvc repro
'data.dvc' didn't change, skipping
Stage 'cats-and-dogs' didn't change, skipping
Data and pipelines are up to date.
```

4
Model versioning



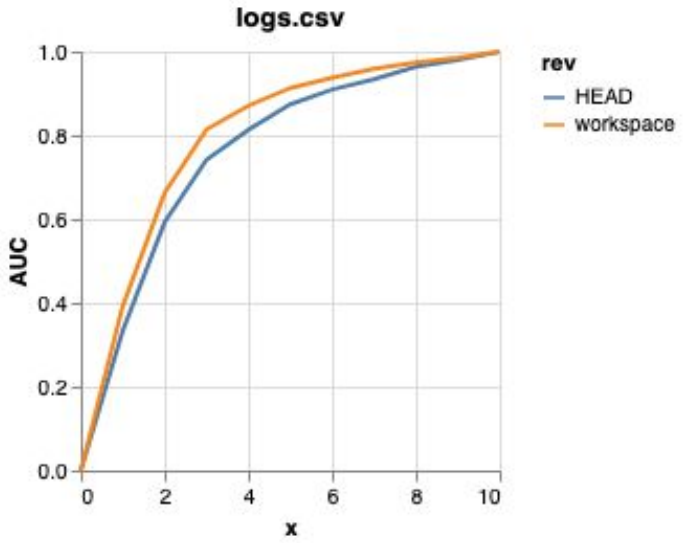
Metrics diff

```
$ dvc metrics diff
Path                Metric        Value         Change
metrics.json        accuracy      0.213516      0.023528
metrics.json        loss          2.142695      -0.382401

Also:
$ dvc metrics diff HEAD^^
$ dvc metrics diff master
$ dvc metrics diff v2.1 v2.3
...
```

Plots file

```
$ dvc plots diff
file:///Users/dmitry/
src/exp4/plots.html
```

→



Also:

```
$ dvc plots diff HEAD^^
$ dvc plots diff master
$ dvc plots diff v2.1 v2.3
```

...

Getting ML model (to production)

```
$ dvc list https://github.com/iterative/myproject
model.h5
train.py
...

$ dvc get https://github.com/iterative/myproject model.h5
$ du -sh model.h5
4.7M   model.h5
```

Getting ML model (to production) - Python API

```python
import dvc.api

with dvc.api.open(
        'get-started/data.xml',
        repo='https://github.com/iterative/dataset-registry'
        ) as fd:
    # ... fd is a file descriptor that can be processed normally.
```

5

Integration and compatibility CI/CD

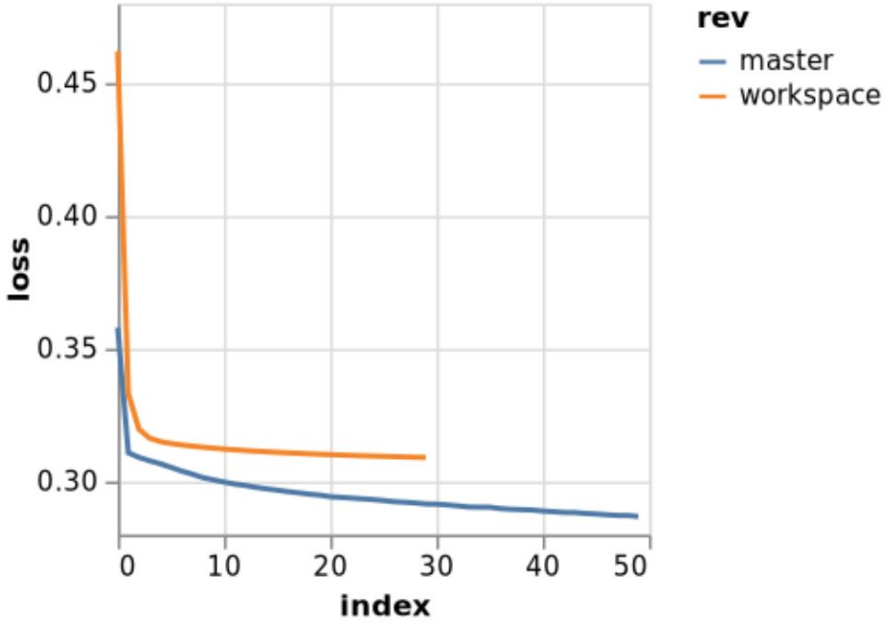
CML – Continuous Machine Learning

CML is CI/CD for Machine Learning Projects.

https://cml.dev/

github-actions bot commented on 591b626 on Jul 2, 2020

Path	Metric	Value	Change
metrics.json	accuracy	0.86944	-0.00292
metrics.json	precision	0.86923	0.00504
metrics.json	recall	0.87185	-0.01382



GitHub Action CI/CD workflow

```
dvc pull data
pip install -r requirements.txt
dvc repro

git fetch --prune
dvc metrics diff --show-md master >> report.md

dvc plots diff \
     --target loss.csv --show-vega master > vega.json
vl2png vega.json | cml-publish --md >> report.md
cml-send-comment report.md
```

5

ML experiment in DVC (demo)

ML experimentation challenges

1. Too many experiments: 10s, 100s, 100s
2. Git commit overhead
3. Models checkpoints (deep learning)

Conclusion

Conclusion: MLOps is a part DevOps

1. DevOps principles and CLI enable the entire ecosystem of tools: Git, GitHub/GitLab, CI/CD, clouds.
2. ML experimentation can be efficiently done in Git.

Thank you!

Dmitry Petrov, PhD

Twitter: @FullStackML

Tutorial: dvc.org/doc/use-cases/versioning-data-and-model-files/tutorial