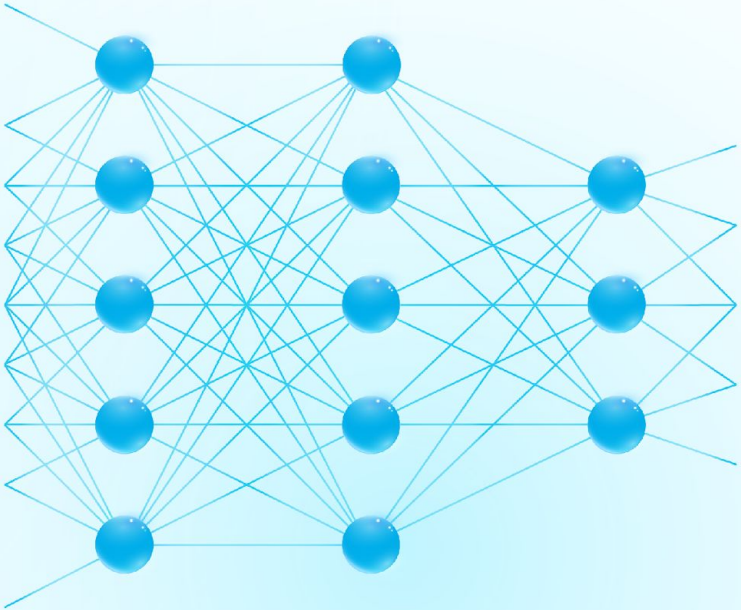
DATA SCIENCE SUMMIT

MACHINE LEARNING EDITION

Turn VS Code into a One-Stop Shop for ML Experiments

Eryk Lewinson



Organizatorem wydarzenia jest Fundacja Academic Partners

ACADEMIC PARTNERS

A bit about me…

- Hi, I'm Eryk 👋
- Sr. Data Scientist at *bol* (🇳🇱 e-commerce platform). I'm working on forecasting problems.
- I write data science articles on Medium.
- I published two editions of *Python for Finance Cookbook* (Packt)
- Hobbies: reading 📖, video games 🎮

Agenda

1. Why track experiments?
2. Why DVC?
3. How it works?
4. Demo 💻

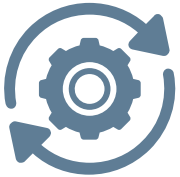
A typical ML workflow

Source: https://commons.wikimedia.org/wiki/File:CRISP-DM_Process_Diagram.png

Sweet spot of ML experiments

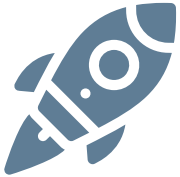
Fast experimentation

OR



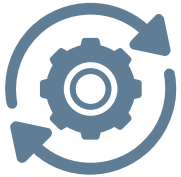
Reproducibility

Sweet spot of ML experiments



Fast experimentation

~~OR~~
AND



Reproducibility

A real-life horror story

“I remember that about 15 months ago, we ran an experiment that achieved a much better score than our current model. Can you quickly find out what we did back then?”

- Made-up colleague

Reproducibility approach #1

Reproducibility approach #2

timestamp	model_type	max_depth	n_estimators	accuracy
2023-11-10 12:10	rf	10	10	0.56
2023-11-10 12:30	rf	15	100	0.61
2023-11-15 15:15	lgbm	10	42	0.62
2023-12-03 9:00	rf	31	100	0.55

Reproducibility approach #2 cntd.

timestamp	model_type	git commit	model	data	max_depth	n_estimators	accuracy	recall
2023-11-10 12:10	rf	341and8	rf.joblib	dataset_v1.csv	10	10	0.56	0.4
2023-11-10 12:30	rf	129any8	rf_1.joblib	dataset_v1.csv	15	100	0.61	0.45
2023-11-15 15:15	lgbm	472hab3	lgbm.joblib	dataset_v1.csv	10	42	0.62	0.47
2023-12-03 9:00	rf	876hsk1	rf_2.joblib	dataset_final.csv	31	100	0.55	0.3

Some of the available solutions

mlflow

CLEARML

Weights & Biases

DVC

comet

Valohai

TensorBoard

neptune.ai

polyaxon

Why DVC?

- DVC is open-source
- You are already (🤞) tracking code with Git
- DVC builds on top of Git
- Works with different storage providers (AWS, GCP, Azure, GDrive, local)
- No metric server needed
- DVC's CLI is similar to Git's
- VS Code extension

```
git init          dvc init
git checkout      dvc checkout
git add           dvc add
git push          dvc push
git pull          dvc pull
git fetch         dvc fetch
git diff          dvc diff
```

Reproducible ML experiments

- Code

Reproducible ML experiments

+

- Code

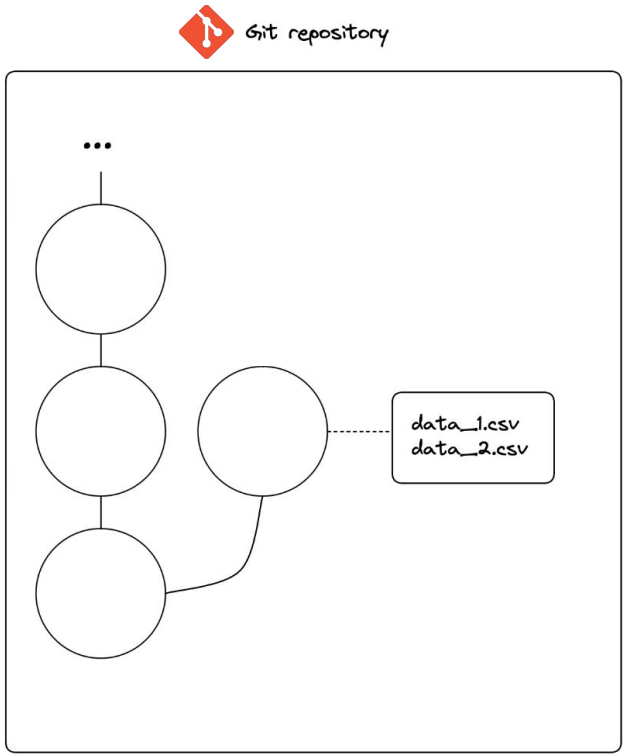
- Data
- Hyperparameters
- Metrics
- Models
- Plots
- Other artifacts

Reproducible experimentation with DVC 101

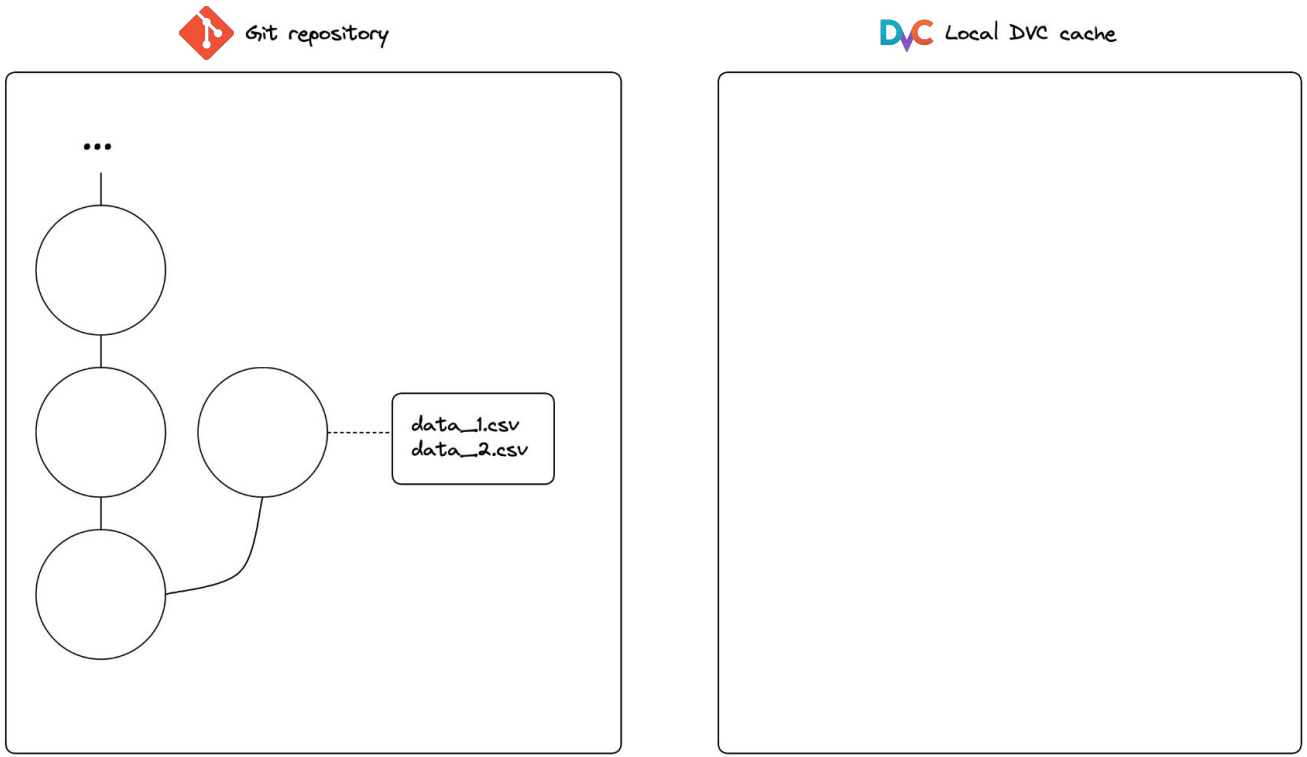
Key components:

1. Versioning data and artifacts
2. DVC pipelines
3. Experiments

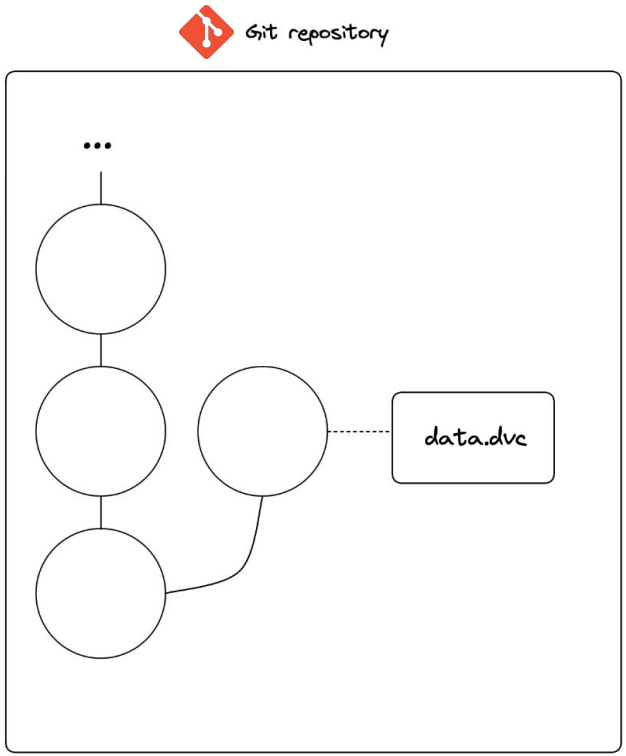
Versioning data and artifacts



Versioning data and artifacts



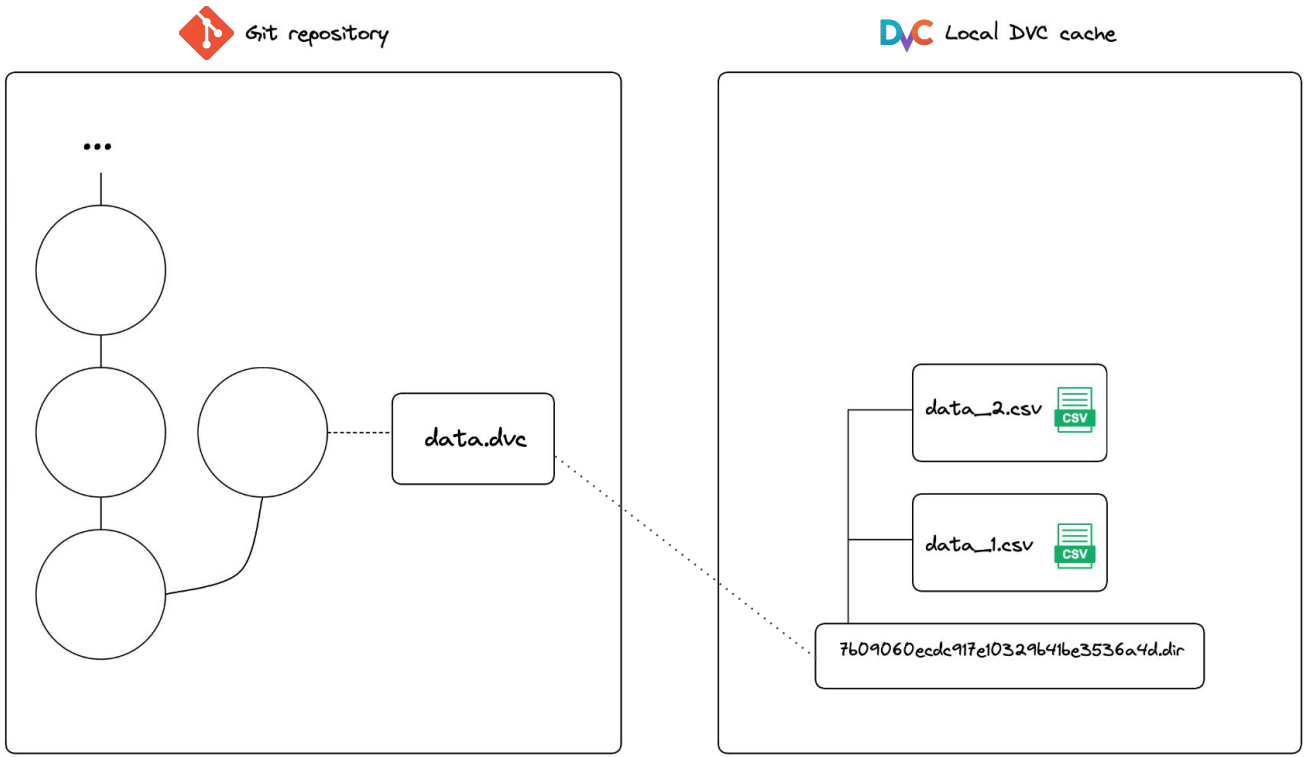
Versioning data and artifacts



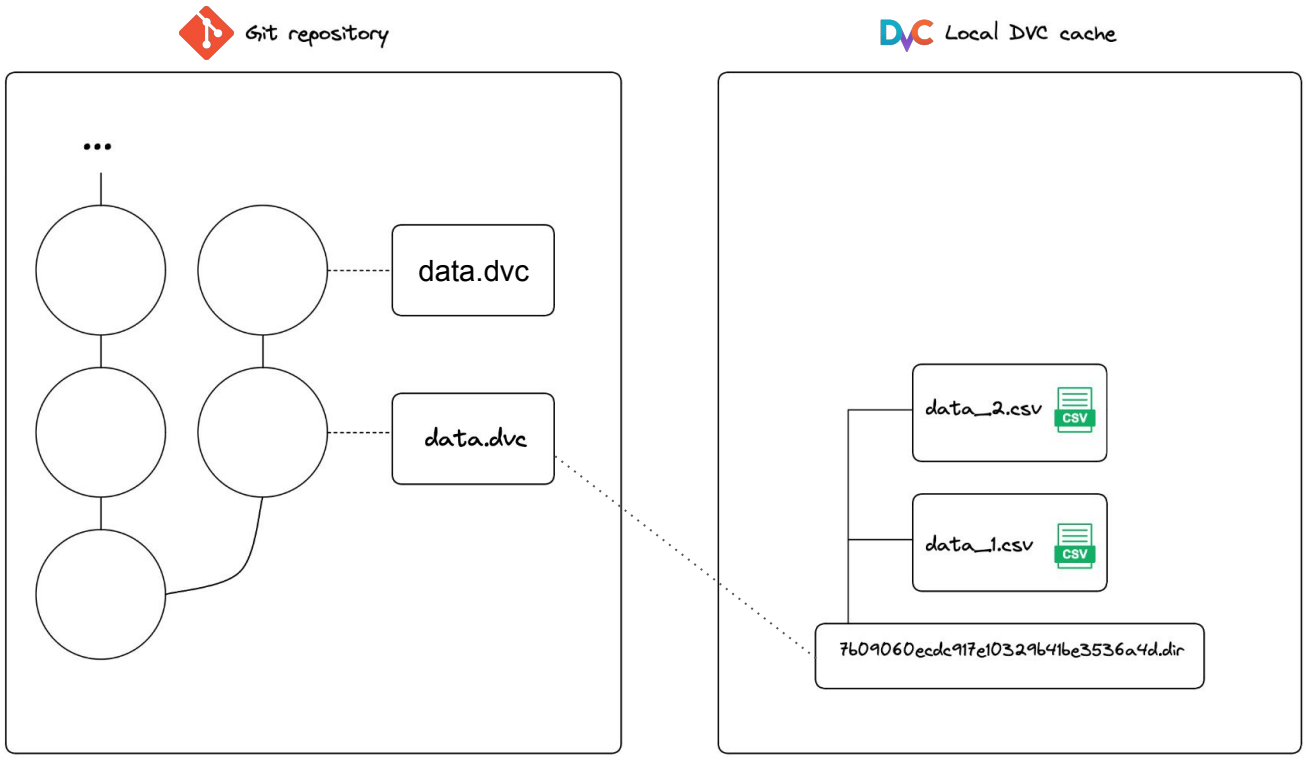
data.dvc

```
outs:
- md5: 7b09060ecdc917e10329b41be3536a4d.dir
  size: 73839
  nfiles: 2
  hash: md5
```

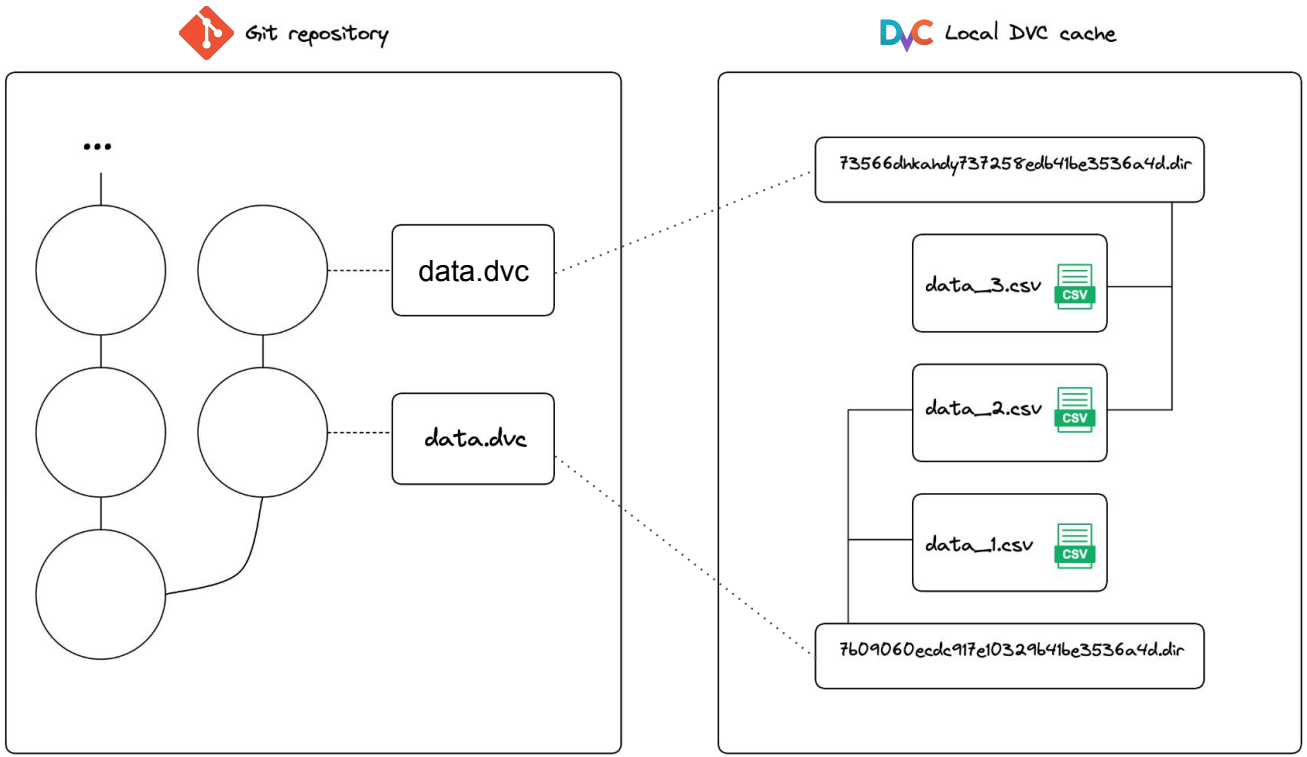
Versioning data and artifacts



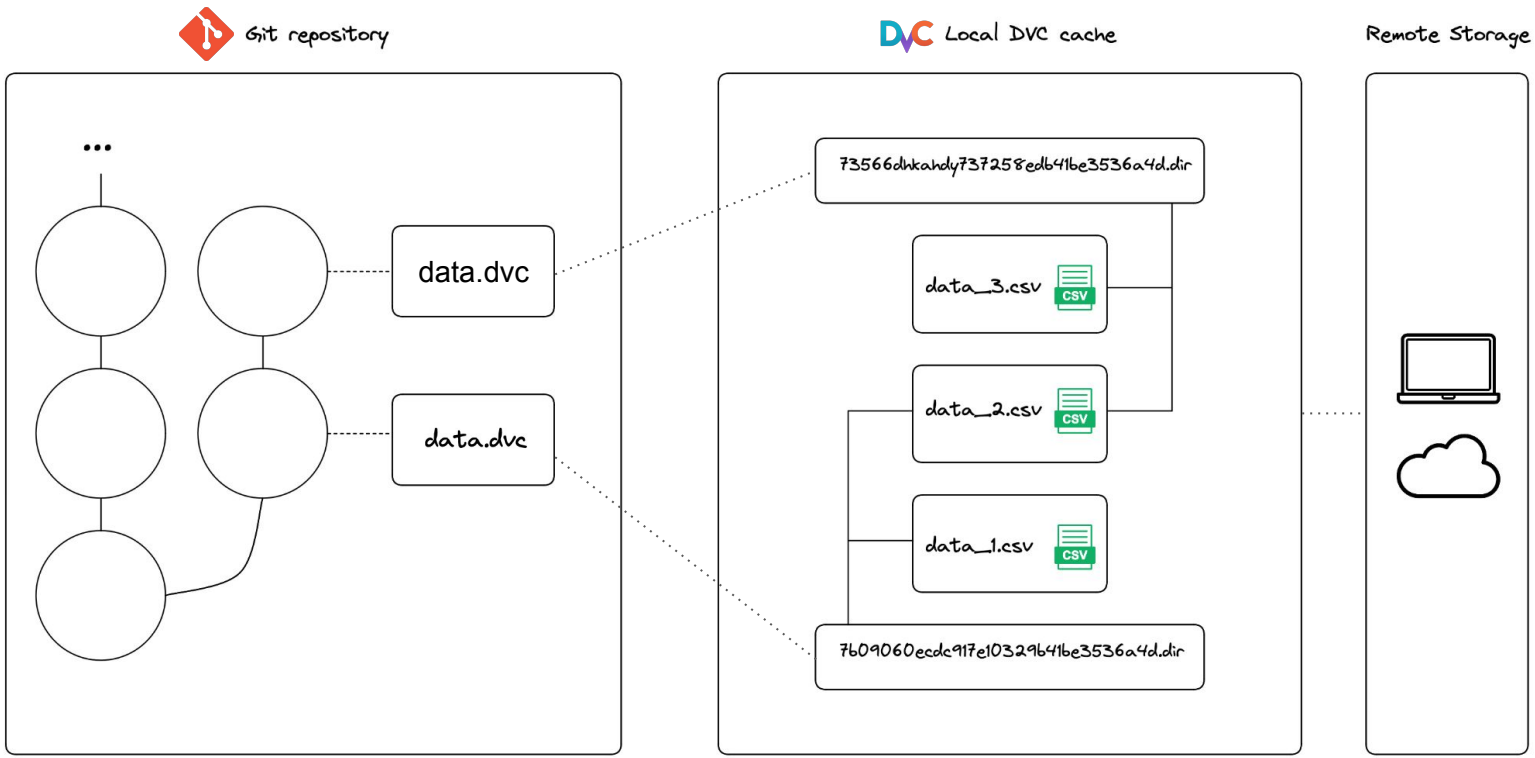
Versioning data and artifacts



Versioning data and artifacts



Versioning data and artifacts



DVC pipelines

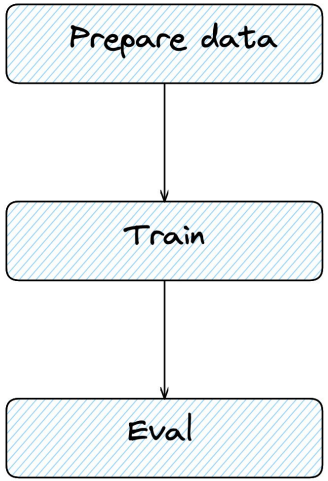
```yaml
stages:
  prepare_data:
    cmd: python src/prepare_data.py
    deps:
    - src/prepare_data.py
    - data/raw
    params:
    - prepare_data
    outs:
    - data/processed
  train:
    cmd: python src/train.py
    deps:
    - src/train.py
    - data/processed
    params:
    - train
    outs:
    - models/model.joblib
  eval:
    cmd: python src/evaluate.py
    deps:
    - src/evaluate.py
    - data/processed
    - models/model.joblib
    metrics:
    - metrics.json
```

dvc.yaml

DVC pipelines

```yaml
stages:
  prepare_data:
    cmd: python src/prepare_data.py
    deps:
    - src/prepare_data.py
    - data/raw
    params:
    - prepare_data
    outs:
    - data/processed
  train:
    cmd: python src/train.py
    deps:
    - src/train.py
    - data/processed
    params:
    - train
    outs:
    - models/model.joblib
  eval:
    cmd: python src/evaluate.py
    deps:
    - src/evaluate.py
    - data/processed
    - models/model.joblib
    metrics:
    - metrics.json
```

dvc.yaml



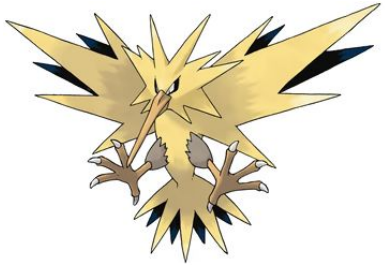
Experiments with

There’s quite a lot of Pokémon

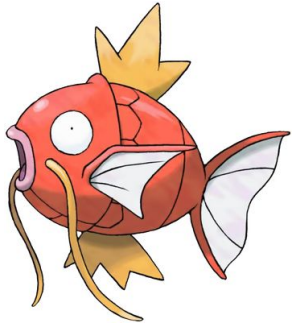
Source: https://bit.ly/pokedex_image

Our task: classify legendary Pokémon

	Pokedex Number	Name	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Legendary	Generation	Type 1	Type 2	Mega Evolution
0	1	Bulbasaur	318	45	49	49	65	65	45	False	1	Grass	Poison	False
1	2	Ivysaur	405	60	62	63	80	80	60	False	1	Grass	Poison	False
2	3	Venusaur	525	80	82	83	100	100	80	False	1	Grass	Poison	False
3	3	Mega Venusaur	625	80	100	123	122	120	80	False	1	Grass	Poison	True
4	4	Charmander	309	39	52	43	60	50	65	False	1	Fire	NaN	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1190	1006	Iron Valiant	590	74	130	90	120	60	116	False	9	Fairy	Fighting	False
1191	1007	Koraidon	670	100	135	115	85	100	135	True	9	Fighting	Dragon	False
1192	1008	Miraidon	670	100	85	100	135	115	135	True	9	Electric	Dragon	False
1193	1009	Walking Wake	590	99	83	91	125	83	109	True	9	Water	Dragon	False
1194	1010	Iron Leaves	590	90	130	88	70	108	104	True	9	Grass	Psychic	False



Legendary: True



Legendary: False

Source: https://pokemondb.net/

Demo time 💻

Wrapping up

- Git + DVC + VS Code = One-stop shop for ML experimentation
  - **Git** handles versioning for code, configuration, and small text files (metadata pointers).
  - **DVC** facilitates the creation of ML pipelines and experiments, while also managing the versioning of artifacts such as data, models, plots, metrics, etc.
  - **VS Code** serves as a user-friendly interface for experiment management and visualizations.

Thanks for listening!

Let’s stay in touch:

- linkedin.com/in/eryklewinson
- medium.com/@eryk-lewinson
- https://github.com/erykml

GitHub repo of this project:

- https://bit.ly/dss_ml_2024

My book:

- https://bit.ly/pff_2

FEEDBACK

Turn VS Code into a One-Stop Shop for ML Experiments

Eryk Lewinson

http://ml.dssconf.pl/user.html#!/lecture/DSSML24-0c8f/rate