Machine Learning: What are We Doing Here?

An Introduction to lolpop

An Inquiry

I am not sure if there are other companies like this, but we don't have any CI/CD or SDLC system or culture in our company (Like none, seriously! People share their script through Teams chat and fall into this chaos everytime with the different versions of scripts.). I've been trying to bring in some good tech practices (Like having docstring and comments on top of very basic CI and CD practices using GitHub and GitHub actions), but it's tough going when you're one of the only two people who get why GitHub and good CI/CD is crucial. 😅
So, I'm at this point where I'm in a hurry to find a new place where I can start real ML and software dev work as most of my time is spent on fixing problems with these issues.

And a response

What you're describing is, I think, much more common than not. In truth there are very few companies out there that are truly "tech-centered", and would have super good software engineering practices that newcomers would come into and be impressed and feel empowered to learn a lot about 'proper' practices.

Small Teams Struggle

Large Teams Struggle

We've been led to believe this is important

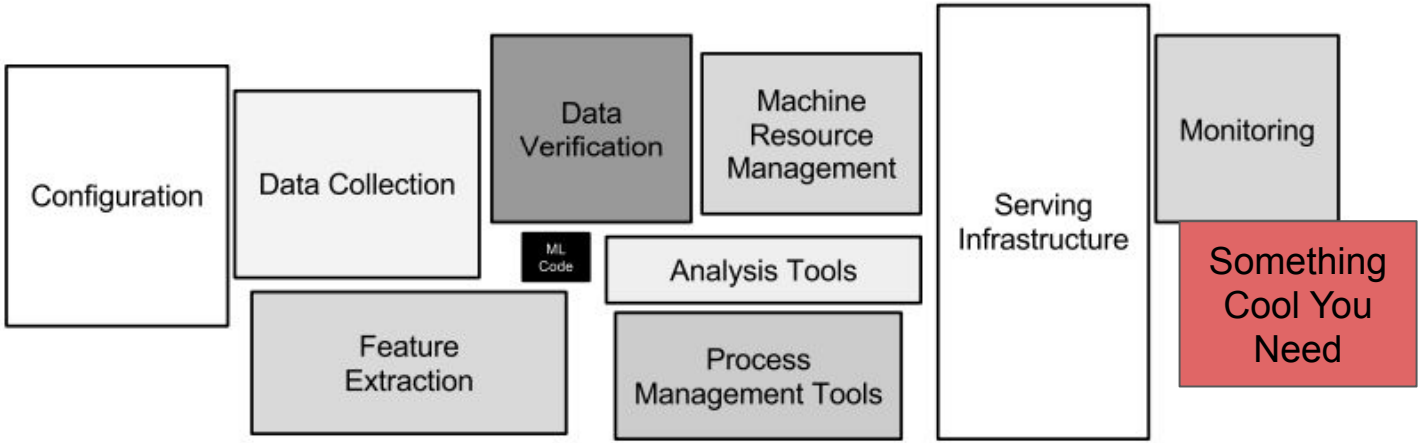


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

It's Not

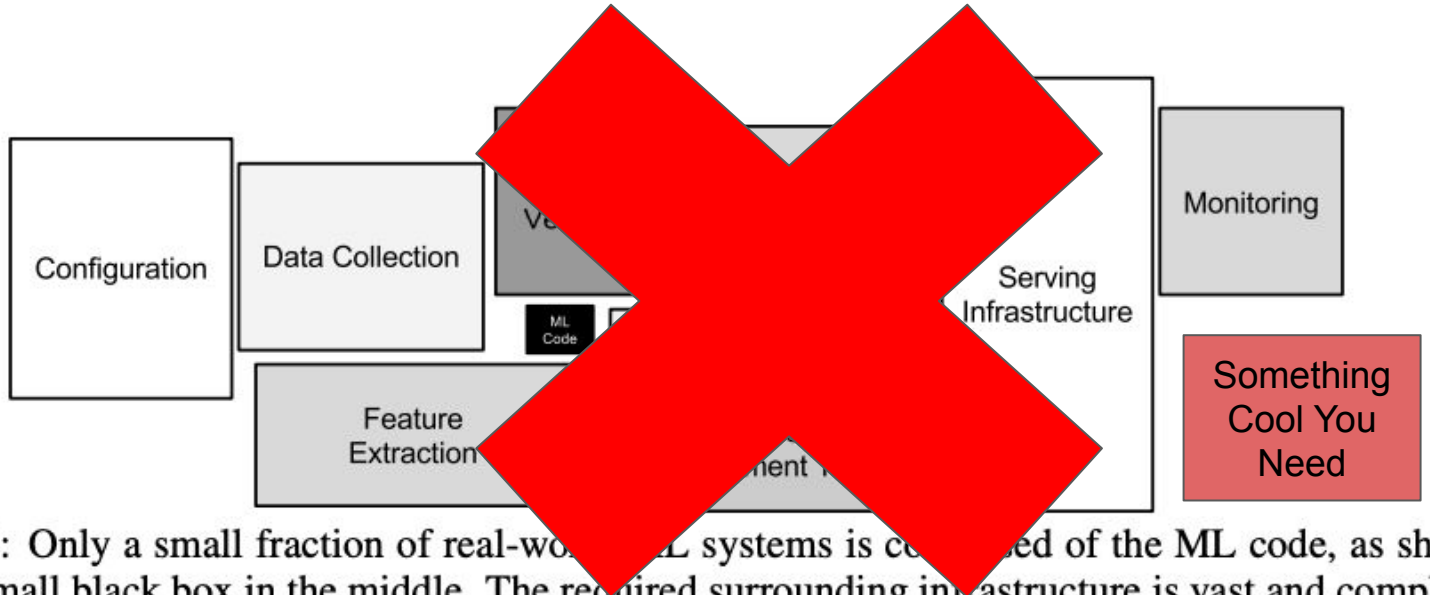


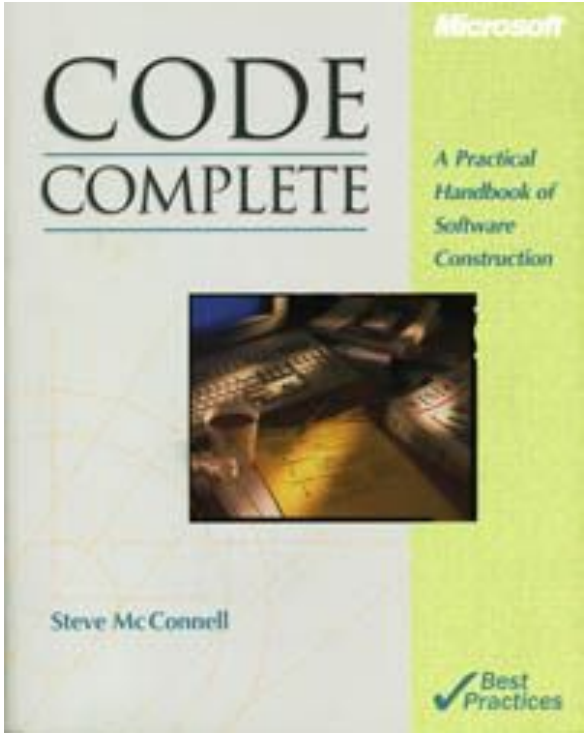
Figure 1: Only a small fraction of real-wo L systems is c ed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

We’ve built lots of good tools

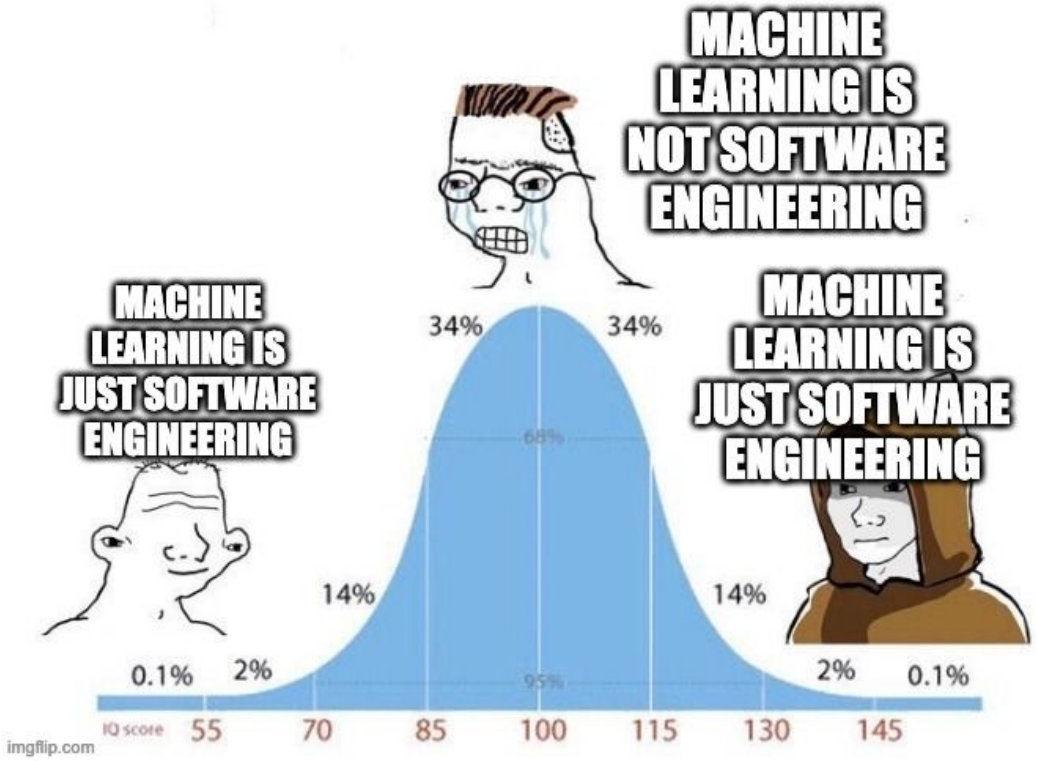
But, probably too many tools

We need to think about the process

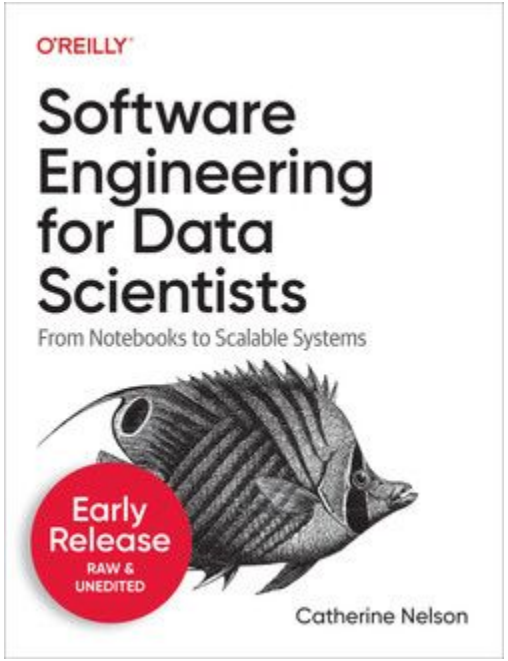
SWE figured this out decades ago

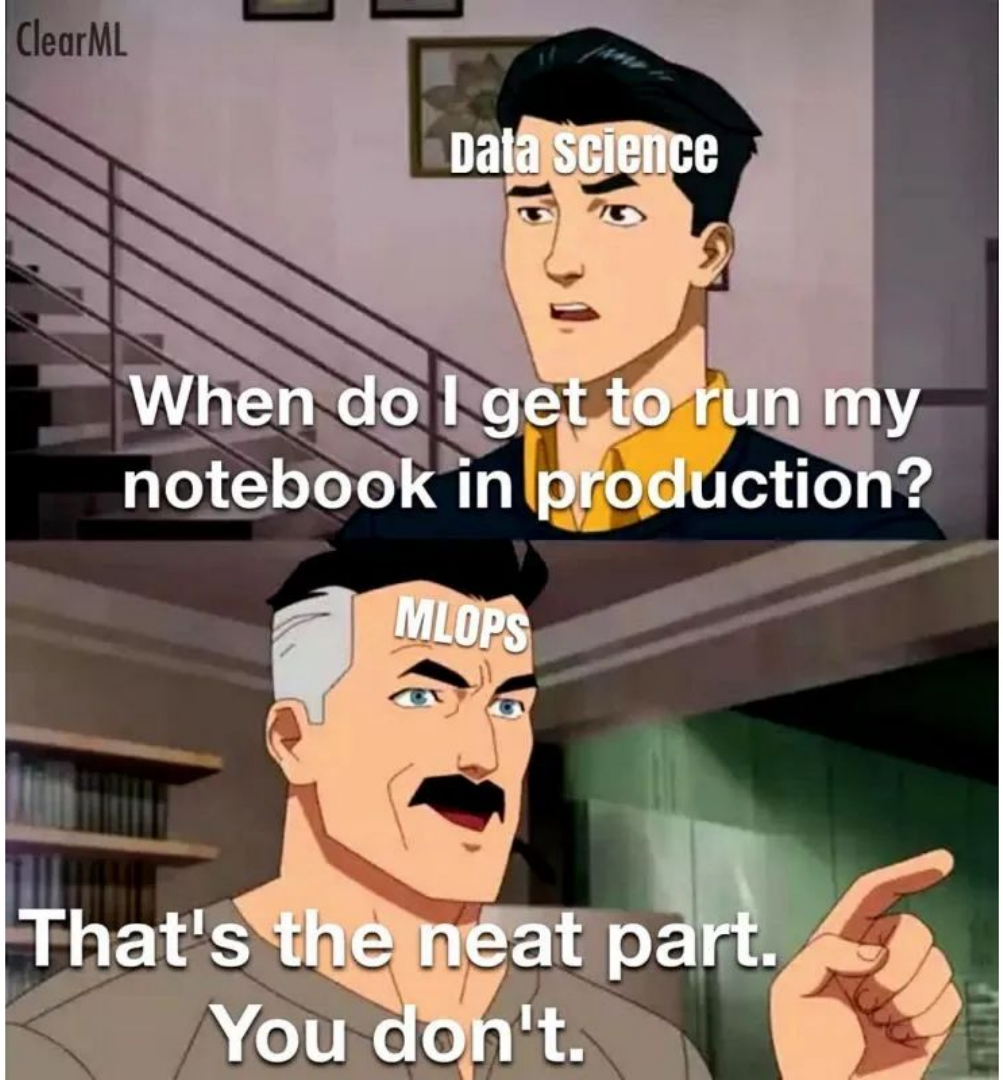


Historically, a debatable topic in ML



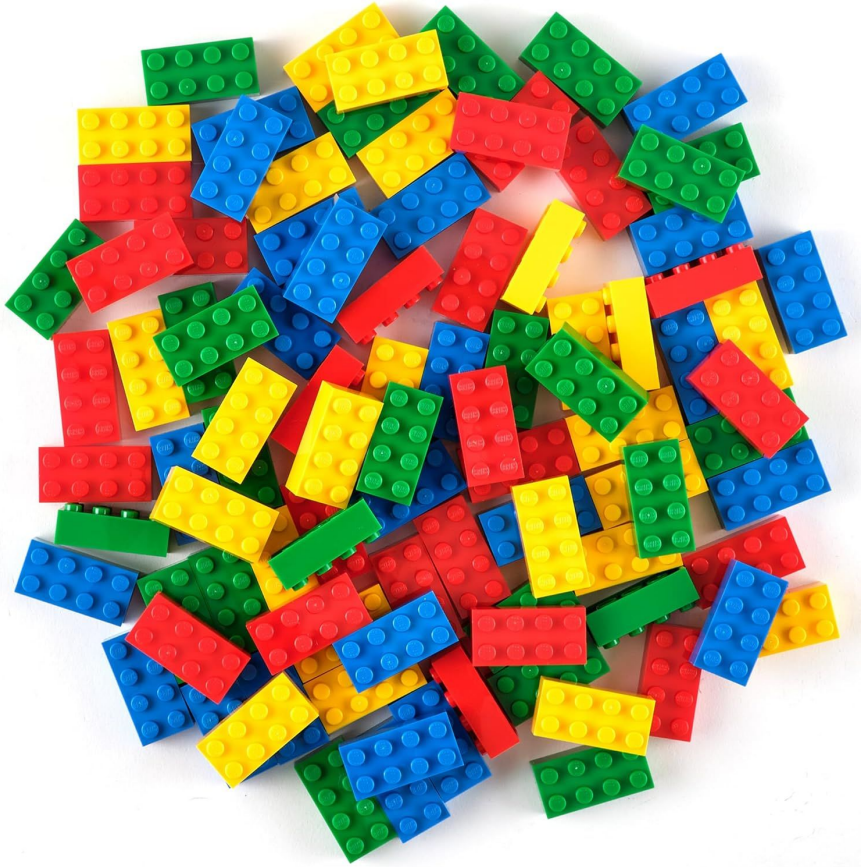
We’re making Some Progress




ClearML
Data Science
When do I get to run my notebook in production?
MLOPS
That's the neat part.
You don't.

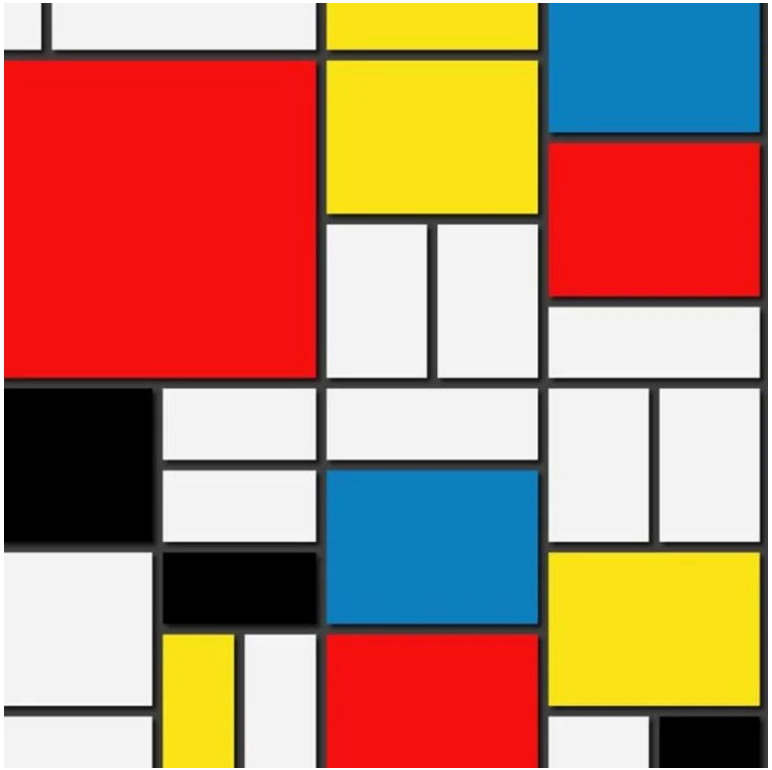
Core SWE for ML Principles

Modularity



Composition

Abstraction



Extensibility

What you should care about

1. Getting things done.
2. Building robust dev to prod pipelines.
3. Not being locked into tech. (i.e. one vendor doesn’t rule)

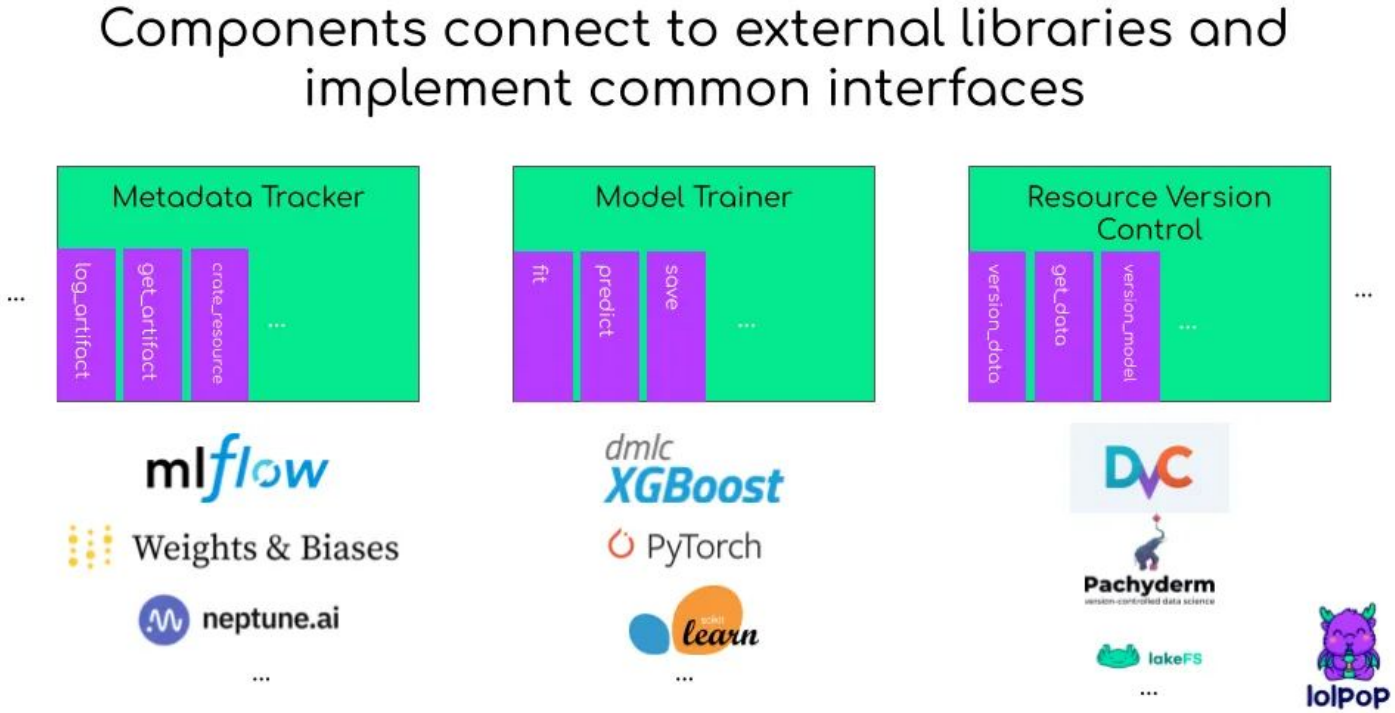
Introducing lolpop!

lolpop is an open software engineering framework for machine learning workflows.

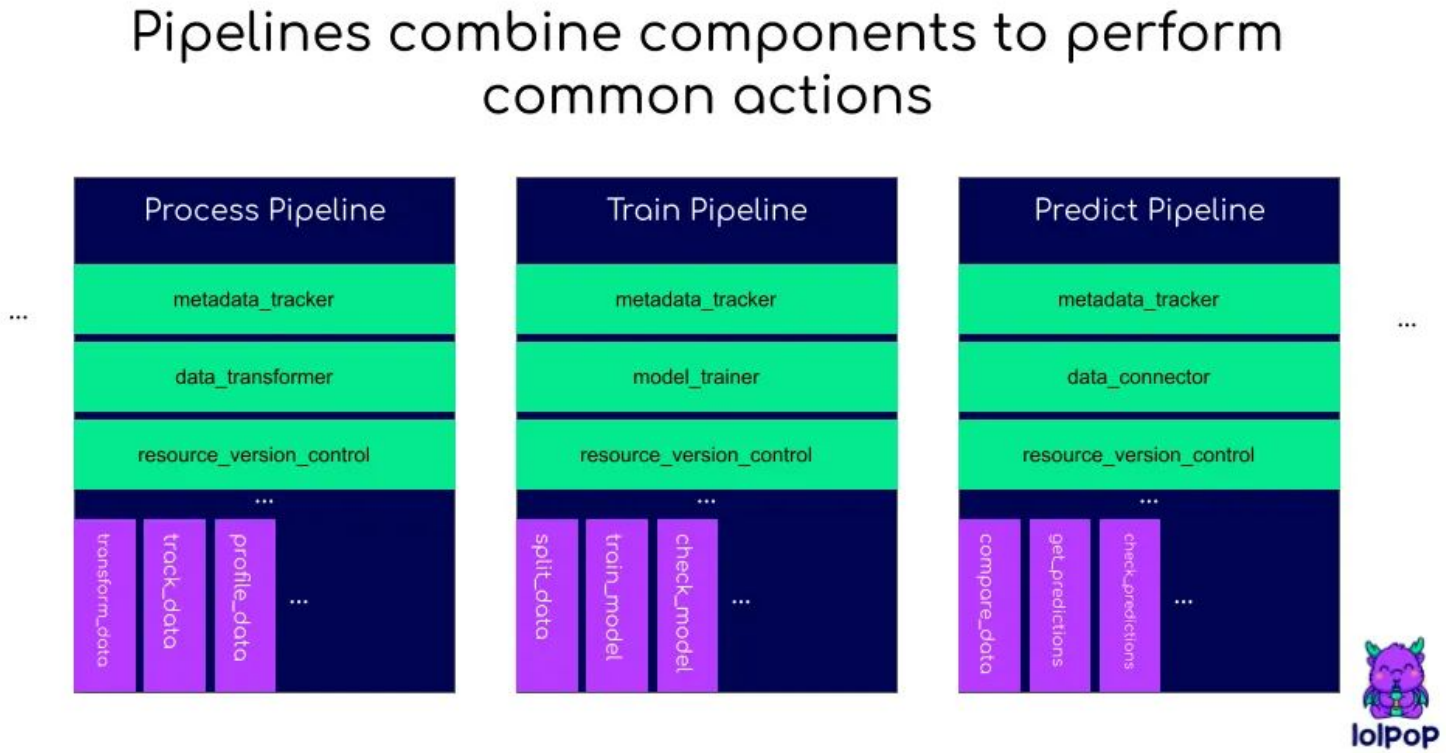
It it meant to help DS/MLE/SWE build and run ML workflows quickly and easily while following SWE best practices.

I.E. we've thought about some of the hard stuff so you can focus on the "fun stuff".

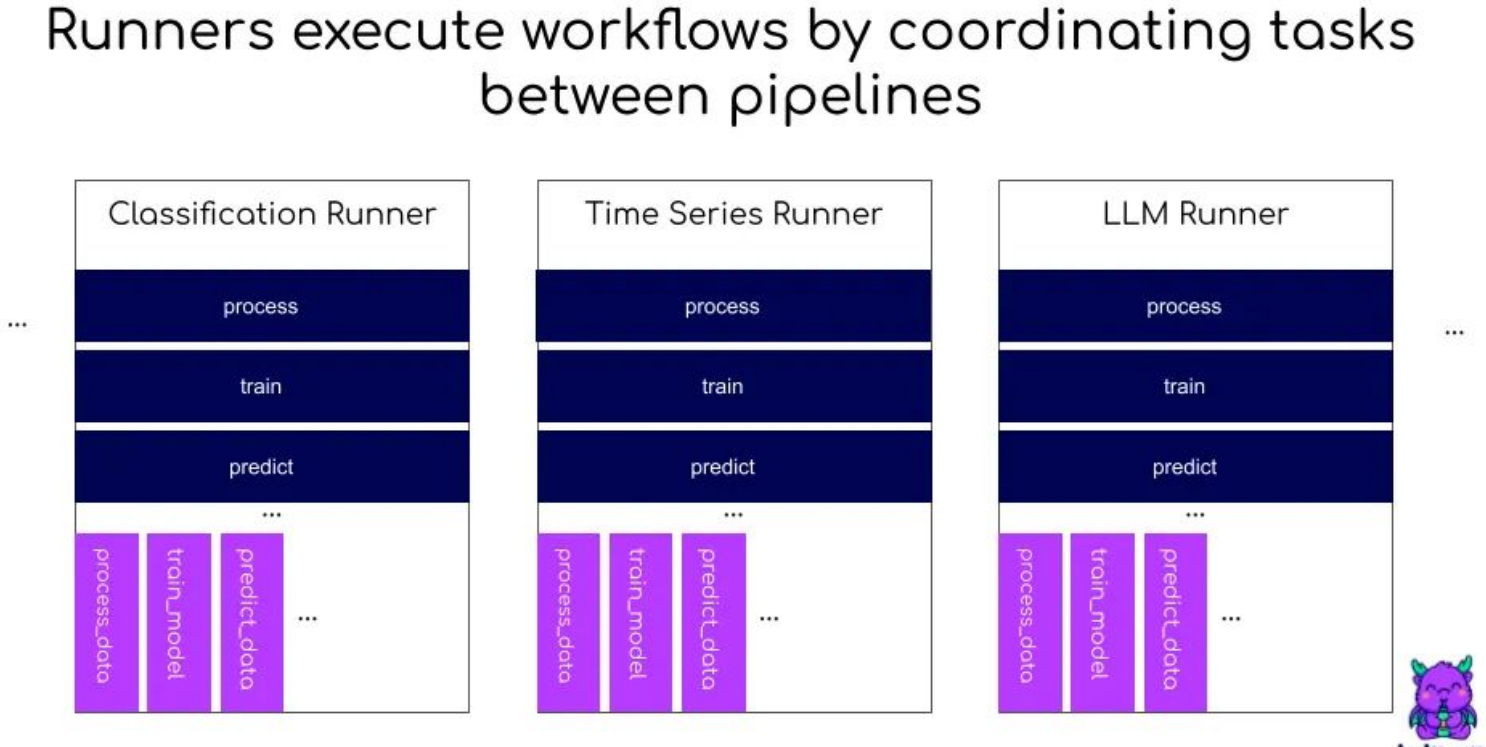
Modularity


Components connect to external libraries and implement common interfaces
Metadata Tracker
log_artifact
get_artifact
create_resource
...
Model Trainer
fit
predict
save
...
Resource Version Control
version_data
get_data
version_model
...
...
mlflow
Weights & Biases
neptune.ai
...
dmlc XGBoost
PyTorch
scikit learn
...
DVC
Pachyderm
lakeFS
...
lolPoP

Composition



Composition Pt. 2



Abstraction

my_training_pipeline.py

```
from lolop.pipeline import BasePipeline

class MyTrainingPipeline(BasePipeline):
    ...
    def train_model(self, data, *args, **kwargs):
        model = self.model_trainer.fit(data)
        return model
    ...
```

dev.yaml

```
#runner config
pipelines:
  train: MyTrainingPipeline
  ...

#pipelines config
train:
  components:
    model_trainer: CatBoostTrainer
  model_trainer:
    config:
        training_params:
            iterations: 2
            depth: 2
            learning_rate: 1
            loss_function: RMSE
...
```

Extensibility

```
from lolpop.comonent import BaseComponent
from catboost import CatBoostRegressor, CatBoostClassifier

class CatBoostModelTrainer(BaseComponent):

    def __init__(problem_type=None, params={}, *args, **kwargs):
        super().__init__(*args, **kwargs)

        if problem_type == "classification":
            self.model = CatBoostClassifier(**params)
        elif problem_type == "regression":
            self.model = CatBoostRegressor(**params)

    def fit(self, data, *args, **kwargs):
        self.model.fit(data["X_train"], data["y_train"])

        return self.model

    ...
```

And more!

- Built-in Testing Framework
- Use Gen-AI to generate documentation/tests
- Generalizable Integration Framework
- Extensible CLI
- Integration packaging
- Workflow Orchestration Support
- Project templating
- Everything is extensible.

- Open Source – free to use forever.

Where to find lolpop

Github: https://github.com/jordanvolz/lolpop

pyPI: pip install lolpop

Docs: https://lolpop.readthedocs.io/en/latest/

Launch blog: https://medium.com/@jordan_volz/introducing-lolpop-c78581d6ee1f

Roadmap: https://medium.com/@jordan_volz/announcing-lolpop-v0-2-0-6667df573aad

Get involved – looking for contributions, robust examples, etc.