Model Development

Robert Clements
MSDS Program
University of San Francisco

We'll Start with Model Development and Talk About the Data Later

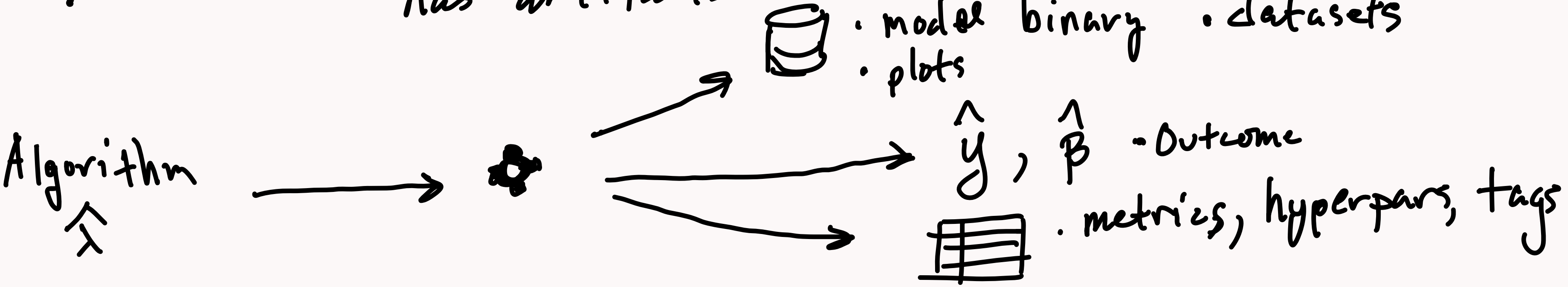
- Goal: to learn how to approach the development of a machine learning model like a scientist.
- How: in the lab we will grab some freely available ML datasets and use a popular tool called MLFlow to conduct "experiments".
- Note: we are ***not*** going to be all that concerned with the actual model that we build. That is not the point here.

Experiment Tracking

How do YOU keep track of your models during model development?

Doing Experiments

- Not talking about Experimentation (A/B testing, Multi-armed Bandit)
- Main idea:
  - Each model trained is a run of an experiment, w/ a specific treatment (hyperpars, dataset, etc.)
  - Each run has an outcome or response (model, predictions)
  - " " has artifacts and metadata



What Should We Track in our Experiments?

- Hyperparameters
- Runtime – operational
- Performance
- Algorithm
- Datasets
- Feature importance
- Explanations

Examples

- You randomly split your data into a training and testing set using a random seed:
  - You should log the seed and version your training, validation and testing sets.
- You do hyperparameter tuning to find the optimal model:
  - You should log all results from each model (metric, performance charts, model weights, hyperparameter values) for each combination of hyperparameter values.
- You decide to try a different algorithm:
  - You should log all results of this model (metric, performance charts, model weights, hyperparameter values) to compare with your other models.

Why do we need a new tool to do this for us?

- It is difficult, and prone to error, to track results by porting them over to a spreadsheet.

Iteration	Training	Validation	Testing	Model-ID	algorithm	mtry	ntree	AUC
1	dataA	valA	testA	rf1.1	rf	3	50	0.65
2	dataA	ValA	testA	rf1.2	rf	4	50	0.652
3	dataA	valA	testA	rf1.3	rf	5	50	0.652
4	dataA	ValA	testA	rf2.1	rf	3	70	0.651
5	dataA	valA	testA	rf2.2	rf	4	70	0.652
6	dataA	ValA	testA	rf2.3	rf	5	70	0.65
7	dataA	valA	testA	rf3.1	rf	3	90	0.651
8	dataA	ValA	testA	rf3.2	rf	4	90	0.653
9	dataA	valA	testA	rf3.3	rf	5	90	0.654
10	dataA	ValA	testA	rf4.1	rf	3	100	0.654
11	dataA	valA	testA	rf5.1	rf	3	120	0.655
12	dataA	ValA	testA	rf5.2	rf	4	120	0.6551
13	dataA	valA	testA	rf5.3	rf	5	120	0.6551
14	dataA	ValA	testA	rf5.4	rf	5	150	0.655
15	dataA	valA	testA	rf5.5	rf	5	200	0.65

Why do we need a new tool to do this for us?

- It is difficult, and prone to error, to track results by porting them over to a spreadsheet.

- You don't commit all code changes
- Older tools aren't useful for comparing models more deeply

Experiment Tracking Tools

- MLFlow
- Weights + Biases
- Deepchecks(?)
- Neptune $$
- Comet $$

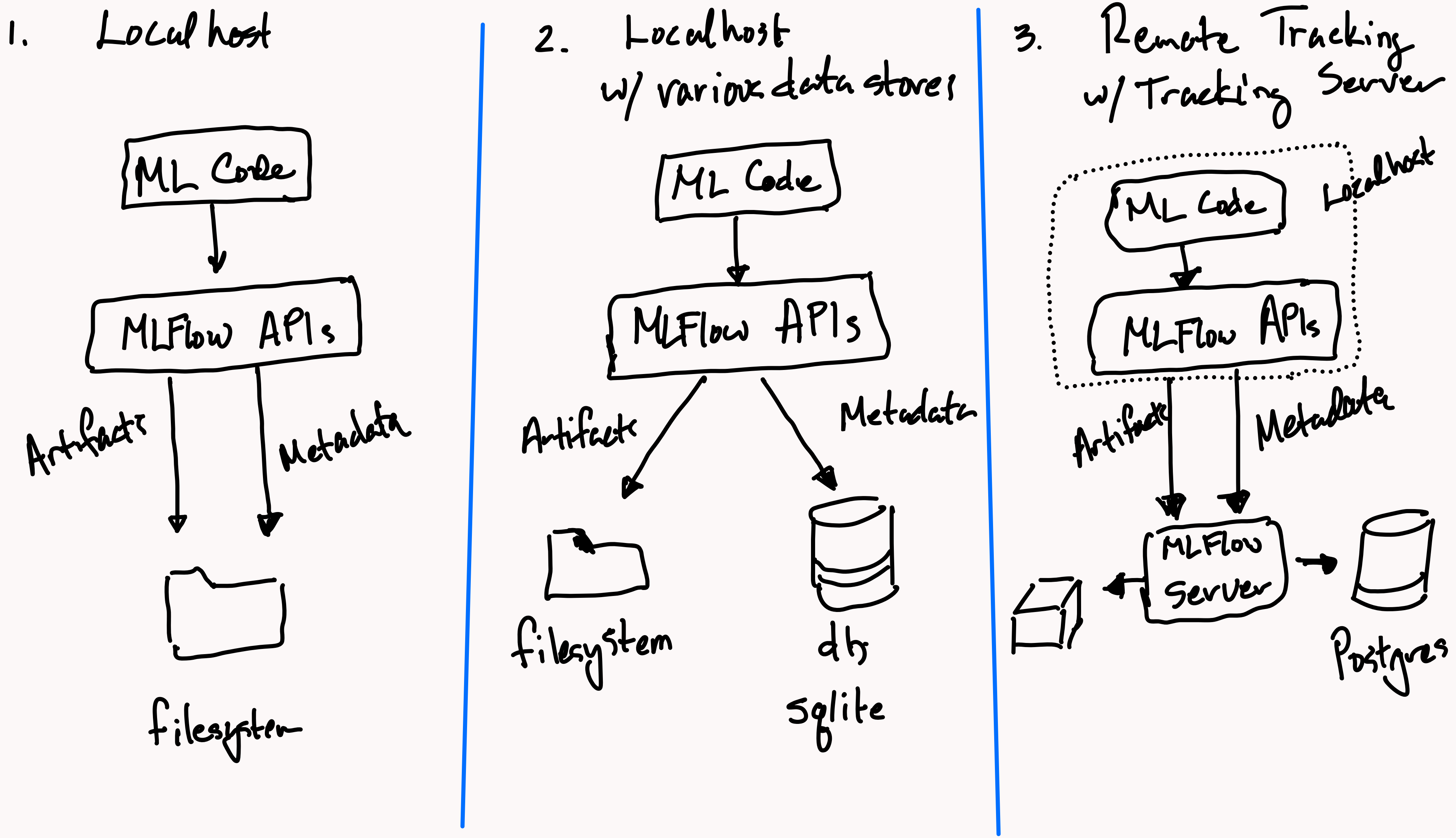
MLFlow

- MLFlow is a tool that promises to do four things:
  - Track experiments
  - Package projects
  - Store artifacts
  - Deploy models

- MLFlow is a python library:
  - Needs a backend store
  - Can run locally or on a server

MLFlow Architecture

- Where should everything be recorded?
- Options:


1. Localhost
ML Code
MLFlow APIs
Artifacts
Metadata
filesystem
2. Localhost
w/ various data stores
ML Code
MLFlow APIs
Artifacts
Metadata
filesystem
db
sqlite
3. Remote Tracking
w/ Tracking Server
Localhost
ML Code
MLFlow APIs
Artifacts
Metadata
MLFlow
Server
Postgres

Experiment Tracking in MLFlow

```
Import mlflow

with mlflow.start_run():
     …code…

     mlflow.set_tag(“Tag_name”, “Tag value”)
     mlflow.log_params(params)
     mlflow.log_metric(“metric_name”, metric_value)
     mlflow.end_run()

OR

     mlflow.autolog() # can customize what gets logged
     mlflow.sklearn.autolog() # specific to sklearn
```

Experiment Tracking in MLFlow

```
Import mlflow

with mlflow.start_run():
     …code…

     mlflow.set_tag("Tag_name", "Tag value")
     mlflow.log_params(params)
     mlflow.log_metric("metric_name", metric_value)
     mlflow.end_run()

OR

     mlflow.autolog() # can customize what gets logged
     mlflow.sklearn.autolog() # specific to sklearn
```

Autologging exists for all of these

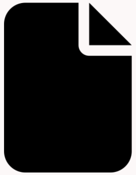
- Fastai
- Gluon
- Keras/TensorFlow
- LangChain
- LlamaIndex
- LightGBM
- OpenAI
- Paddle
- PySpark
- PyTorch
- Scikit-learn
- Spark
- Statsmodels
- XGBoost

MLFlow Experiment Tracking Demo

Artifact Tracking and Model Registry

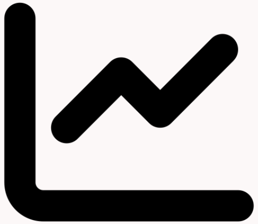
What are artifacts?

- An artifact is any output file that you’d like to store from the runs of your experiments:



Model File

Data



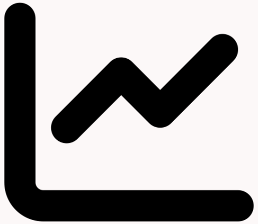
Plots/images

What are artifacts?

- An artifact is any output file that you'd like to store from the runs of your experiments:

Model File

Data



Plots/images

What are artifacts?

- An artifact is any output file that you'd like to store from the runs of your experiments:

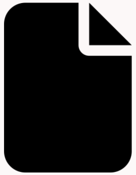
Model File

Data

Plots/images

What are artifacts?

- An artifact is any output file that you'd like to store from the runs of your experiments:



Model File

Data

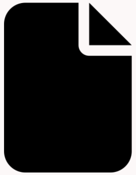
Plots/images

Artifact Tracking in MLFlow

```
Import mlflow

with mlflow.start_run():
      …code…

      mlflow.set_tag(“Tag_name”, “Tag value”)
      mlflow.log_params(params)
      mlflow.log_metric(“metric_name”, metric_value)

      mlflow.log_artifacts(“path/to/artifact”)

      mlflow.end_run()
```

What is a model registry?

- A model registry is where you store and register your models.



Model File

Data

Plots/images

MLFlow Artifact Tracking and Model Registry Demo

Experiment Tracking Lab