To Production and Beyond:
How to Manage the Machine Learning Lifecycle with mlflow

...

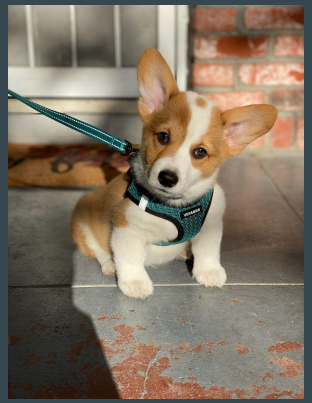
Amanda Moran
Solutions Architect, Databricks

Agenda

- What is the machine learning lifecycle?
- Why should I care about this?
- What is MLflow?
- What is model tracking?
- How to build a reproducible project?
- How to create models that can be run anywhere?

A Little About Amanda ...

- MS Computer Science, BS Biology
- Previously: HP, Teradata, DataStax, Esgyn
- PMC and **Apache Committer** on Apache Trafodion
- Instructor for Udacity Data Engineering Nanodegree



Machine Learning Development is Complex

Why is ML Dev Different than Software Dev?

Traditional Software

Goal: Meet a functional specification

Quality depends only on code

Typically pick one software stack

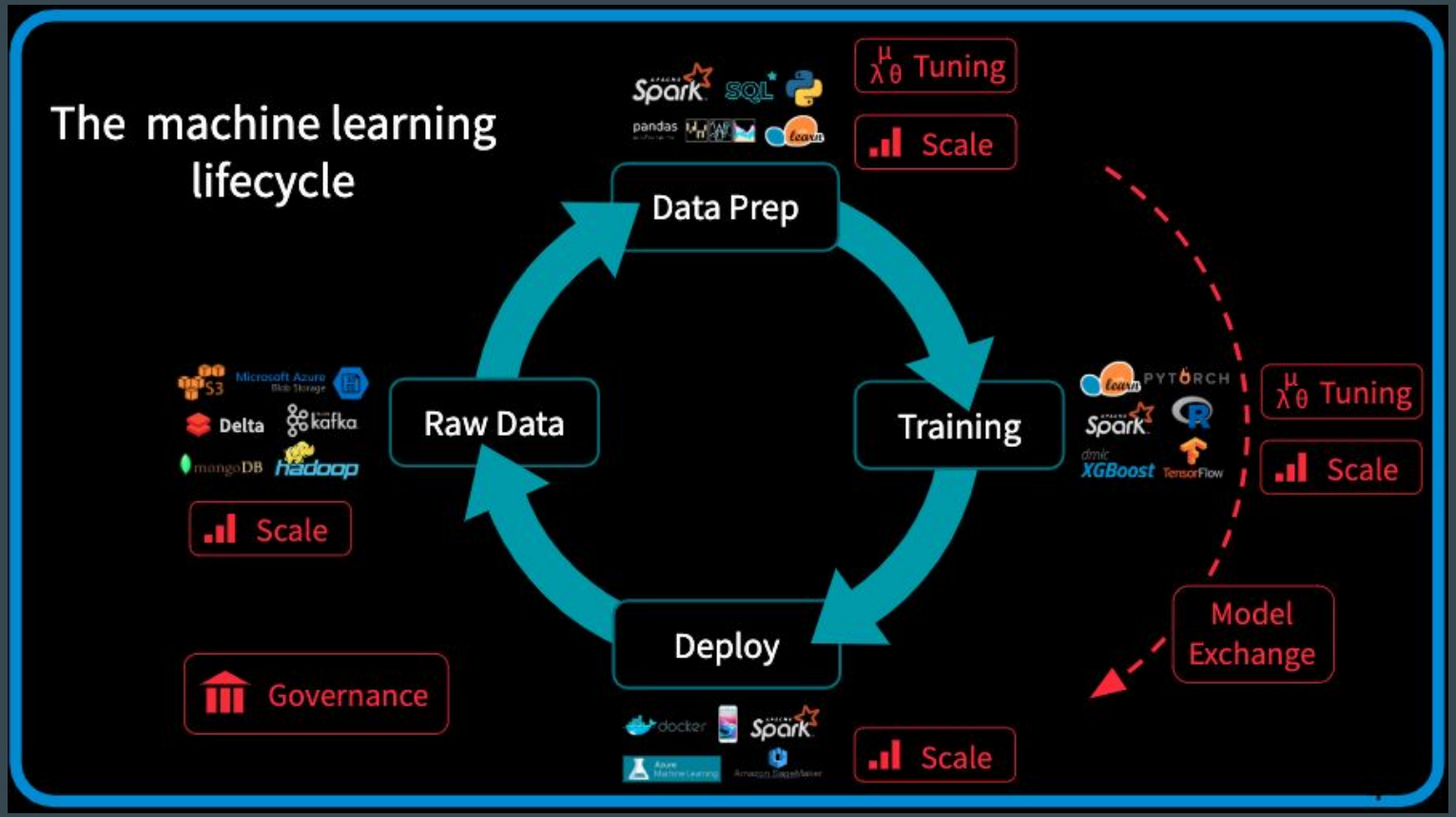
Machine Learning

Goal: Optimize a metric (e.g., accuracy)
- Constantly experiment to improve it

Quality depends on input data
and tuning parameters

Compare + combine many libraries,
models & algorithms for the same task

What is the Machine Learning Lifecycle?



Machine Learning Challenges

- Zoo of software tools
- Tracking & reproducing results
- Productionizing models
- Scaling

Introducing mlflow

To Production and Beyond

What is MLflow?

- Open source project to manage the ML lifecycle:
  - Experimentation
  - Reproducibility
  - Deployment
- Open Sourced by Databricks in 2018

MLflow Components

mlflow Tracking

Record and query experiments: code, data, config, results

mlflow Projects

Packaging format for reproducible runs on any platform

mlflow Models

General model format that supports diverse deployment tools

MLflow Components

API to record results and experiment models, along with what code version, configs and results. This adds structure for reproducibility to your models' experiments

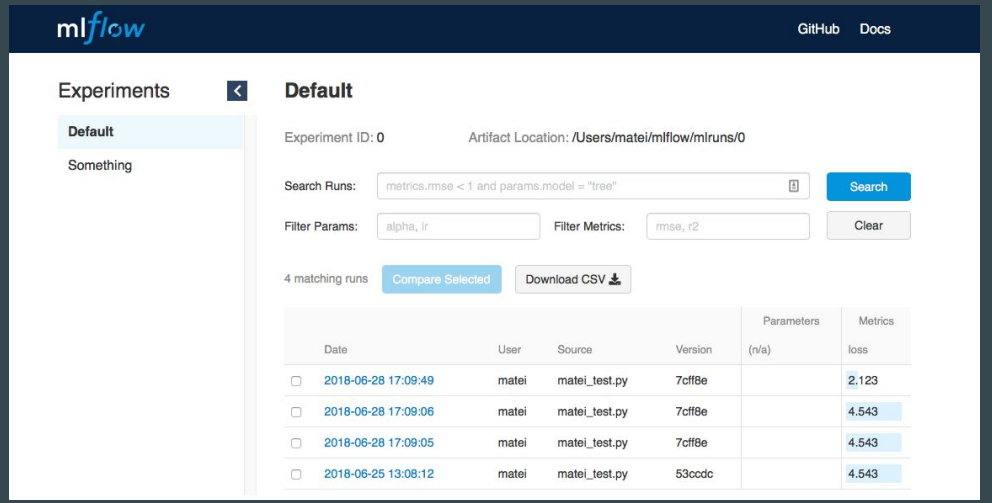
MLflow Tracking Concepts

Parameters: key-value inputs to your code

Metrics: numeric values (can update over time)

Artifacts: arbitrary files, including models

Source: what code ran?



MLflow Tracking API

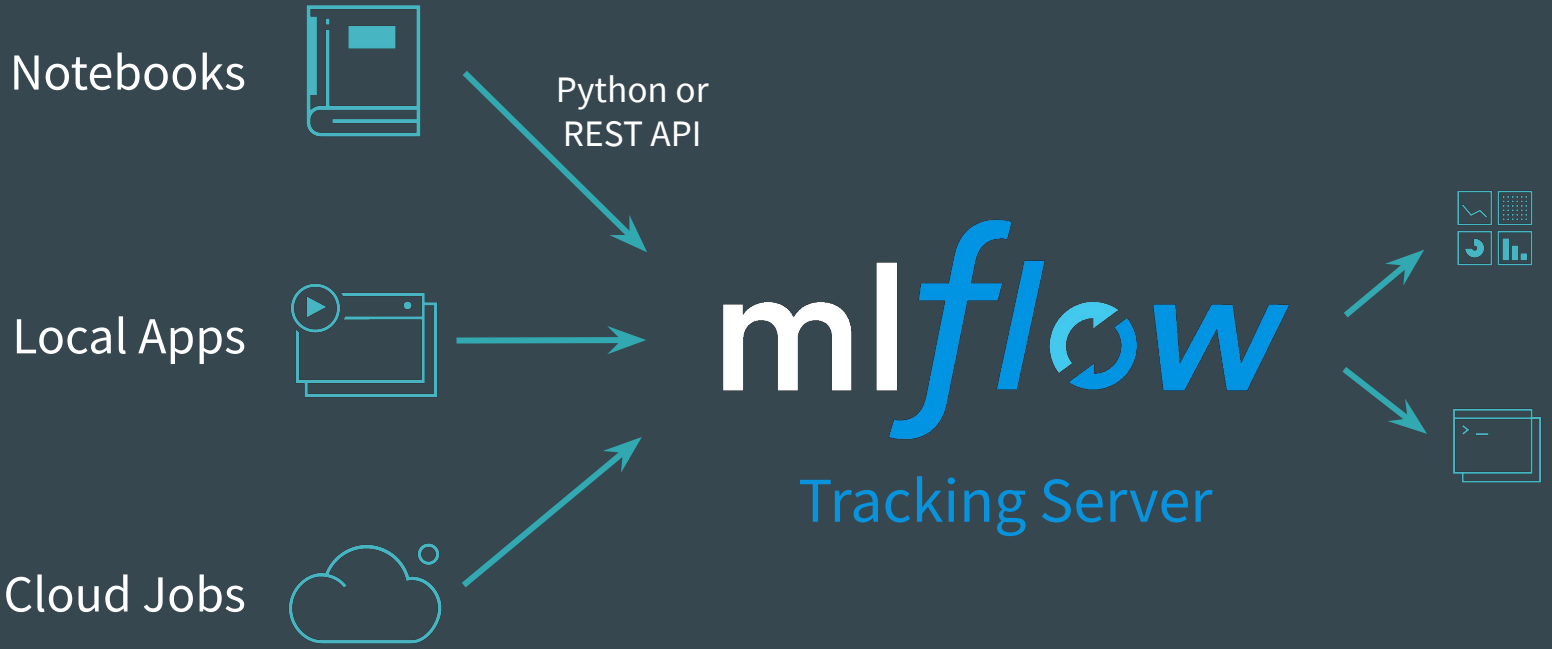
```
import mlflow

 # log model’s tuning parameters

with mlflow.start_run():
  mlflow.log_param("layers", layers)
  mlflow.log_param("alpha", alpha)

  # log model’s metrics
  mlflow.log_metric("mse", model.mse())
  mlflow.log_artifact("plot", model.plot(test_df))
```

MLflow Tracking



MLflow Tracking Demo

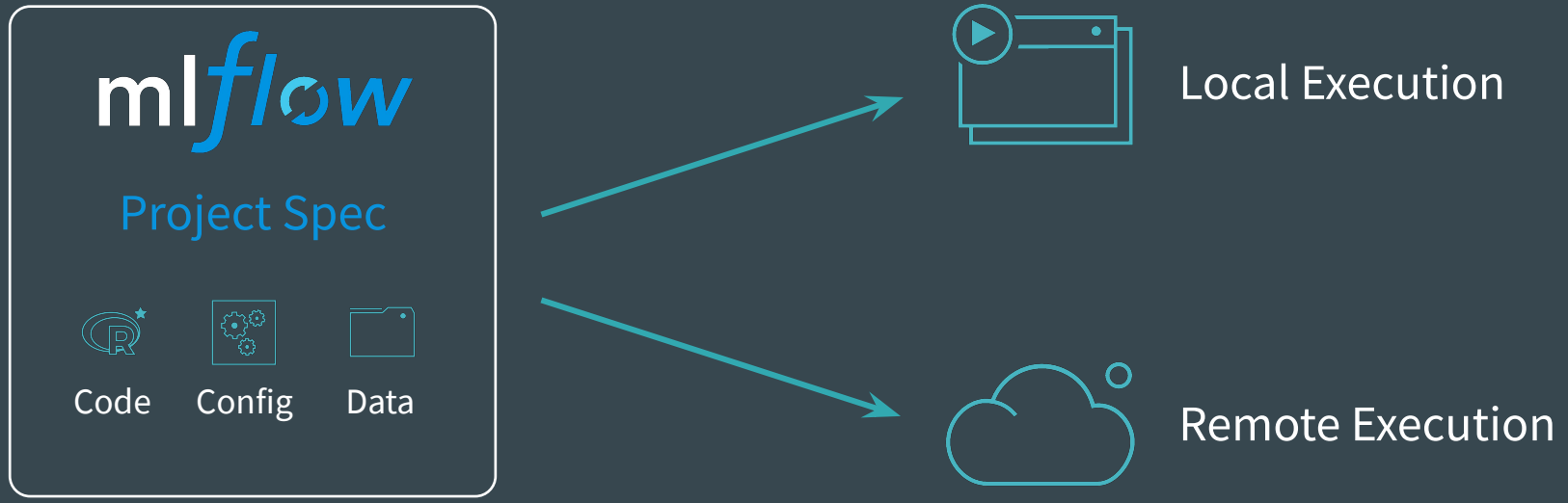
MLflow Components

mlflow
Projects

Packaging format
for reproducible runs
on any platform

Simple, conventional file format to package your project into reproducible runs on any platform by anyone -- ability to add command line parameters

MLflow Projects



MLflow Projects

```
my_project/
├── MLproject
│
│
│
│
│
├── conda.yaml
├── main.py
└── model.py
    ...
```

```
conda_env: conda.yaml

entry_points:
  main:
    parameters:
      training_data: path
      lambda: {type: float, default: 0.1}
    command: python main.py {training_data} {lambda}
```

```
$ mlflow run git://<my_project>

mlflow.run("git://<my_project>", ...)
```

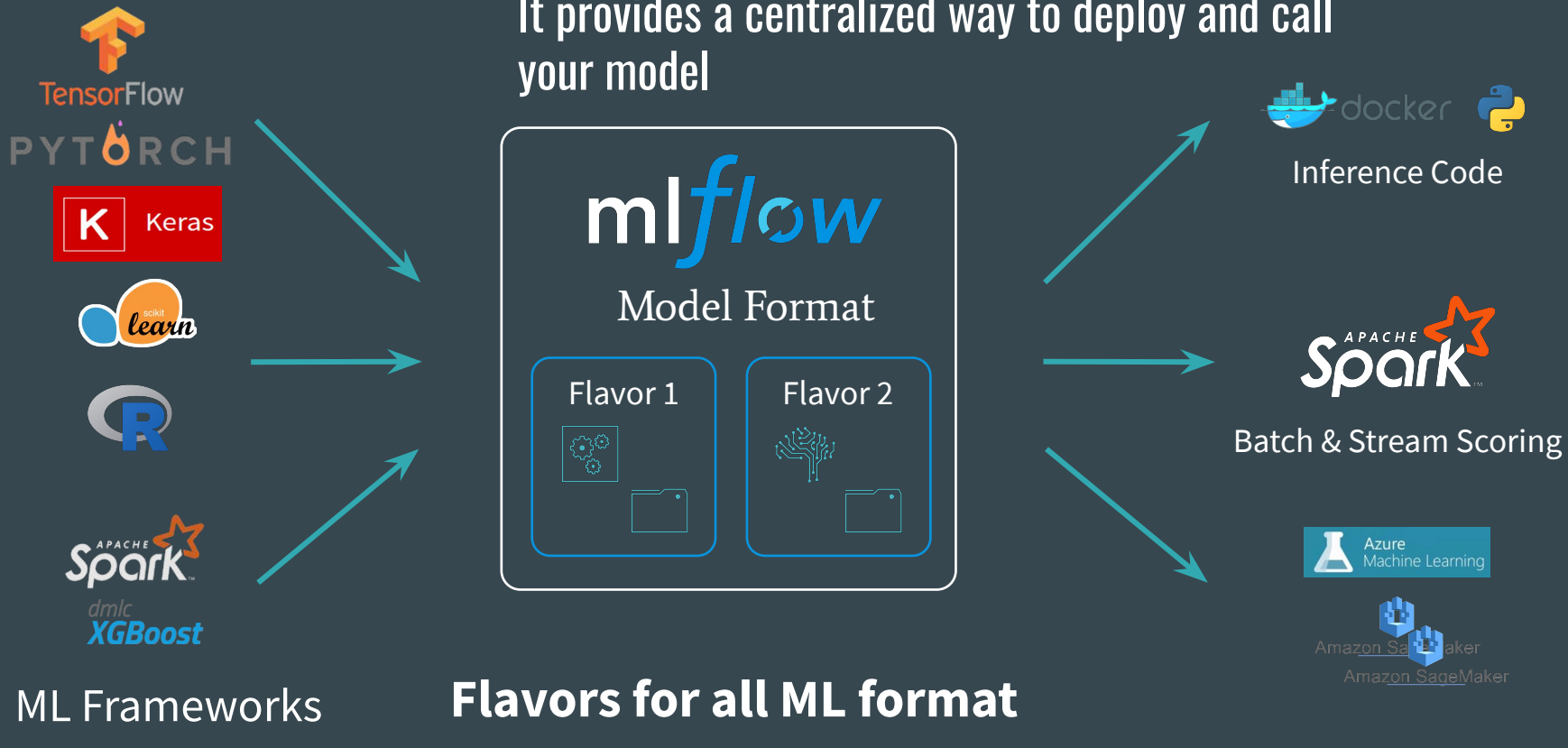
MLflow Projects Demo

MLflow Components

A way to package your models for deployment to diverse execution environments: cloud or local machine or containers

MLflow Models

It provides a centralized way to deploy and call your model



Flavors for all ML format

MLflow Model

```
my_model/
├── MLmodel
└── estimator/
    ├── saved_model.pb
    └── variables/
        ...
```

```
run_id: 769915006efd4c4bbd662461
time_created: 2018-06-28T12:34
flavors:
  tensorflow:
    saved_model_dir: estimator
    signature_def_key: predict
  python_function:
    loader_module: mlflow.tensorflow
```

Usable by tools that understand TensorFlow model format

Usable by any tool that can run Python (Docker, Spark, etc!)

MLflow Models Demo

And More to Come!

A Repository for your ML models -- to help manage the lifecycle of the models

Project Info

Rapid Community Adoption

- 100+ code contributors from >40 companies
- 600K monthly downloads on PyPI

Comparison: Apache Spark took 3 years to get to 100 contributors

- Lots of meetup activity
  - 700+ members in Bay Area

To Get Started Running

```
pip install mlflow
```

More information on Production:

https://thegurus.tech/posts/2019/06/mlflow-production-setup/

To Get Started Coding

Submit issues and patches on GitHub

- We're using it for all our development & issue tracking
- See CONTRIBUTING.rst for how to run dev builds

Join our mailing list: tinyurl.com/mlflow-users

Join our Slack: tinyurl.com/mlflow-slack

Slides and Demo

- Slides: https://github.com/amandamoran/pydatala2019
- Demo: https://www.mlflow.org/docs/latest/tutorial.html

Slides

Demo

Thank you!
Questions?