Demonstration of MLflow: A System to Accelerate the Machine Learning Lifecycle

Corey Zumar, Andrew Chen, Aaron Davidson, Ali Ghodsi, Sue Ann Hong, Andy Konwinski, Siddharth Murching, Tomas Nykodym, Paul Ogilvie, Mani Parkhe, Fen Xie, Matei Zaharia
Databricks Inc.

February 11, 2019

Abstract

Machine learning development creates new challenges that are not present in a traditional software development lifecycle. These include keeping track of the myriad inputs to an ML application (e.g., data versions, code and hyperparameters), reproducing results, and production deployment. Existing systems built to address these challenges restrict the supported programming languages and ML libraries that developers can use to train and deploy models, creating difficulties for organizations that must leverage a variety of ML libraries and model deployment environments. Accordingly, we propose to demonstrate MLflow: a system that streamlines the machine learning lifecycle and is designed to work with any ML library, algorithm, or programming language. Available at mlflow.org, MLflow is an open source project with over 70 contributors. Through our demonstration, audience members will interact with each of MLflow's major components, experiencing firsthand how the platform's open interface enables developers across an organization or research group to collaborate on reproducible machine learning workflows that leverage their preferred languages and ML libraries.

1 Introduction

Machine learning development requires solving new problems that are not part of the standard software development lifecycle. While traditional software has a well-defined set of product features to be built, ML development tends to revolve around *experimentation*: the ML developer will constantly experiment with new datasets, models, software libraries, tuning parameters, etc. to optimize a metric such as model accuracy. Because model performance depends heavily on the input data and training process, *reproducibility* is paramount throughout ML development. Finally, in order to have business impact, ML applications need to be *deployed* to production in an inference-compatible environment (e.g., a REST server); deployments need to be monitored and regularly updated.

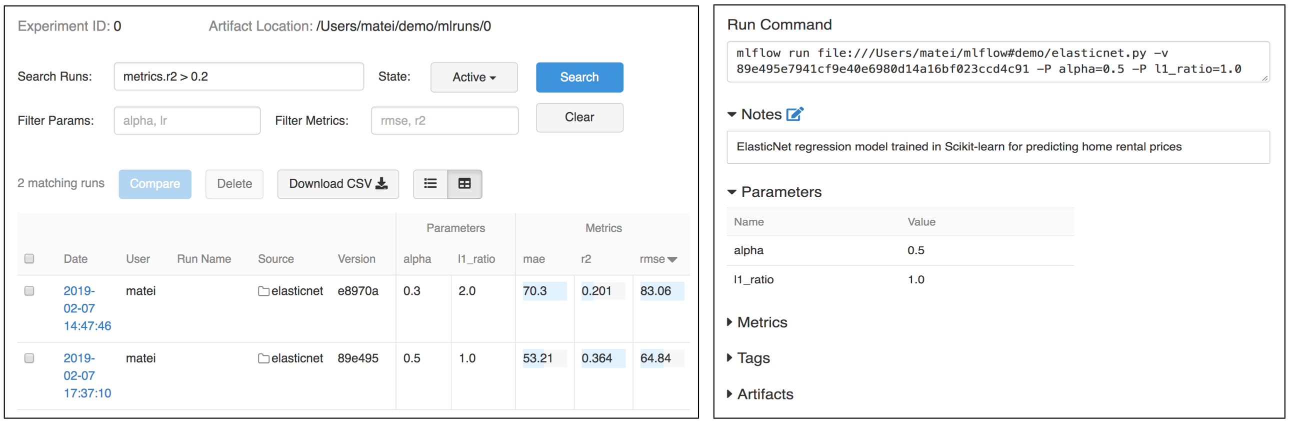
These challenges are encountered at all scales. Even within a small research group, the need for experimental reproducibility necessitates infrastructure to track relevant content and metadata for each model that is produced. In organizations with sizable production ML applications, the difficulty posed by large-scale model tracking is compounded by the challenge of collaborating on ML workflows across geographical regions and domains of expertise, as well as by production deployment.

Faced with these challenges, many organizations develop *ML platforms*, such as Facebook's FBLearner (Dunn., 2016) and Google's TFX (Modi et al., 2017), that ML developers must use in order to create deployable models. Unfortunately, these platforms limit the ML libraries that can be used during model development. For example, TensorFlow offers a training API and a Serving system (Modi et al., 2017), but it cannot easily be used with models from other ML libraries. These restrictions are a bottleneck for many organizations that use multiple ML libraries and deploy models to a variety of production environments.

To address this bottleneck, we introduce MLflow: an open source platform for the ML lifecycle designed to work with any ML library, algorithm, and programming language. MLflow has a rapidly growing open source community and is currently being used to drive ML research efforts and power production ML applications in the energy, biotechnology, and online retail sectors (Zaharia et al., 2018). In the rest of this proposal, we describe the design of the MLflow platform and outline a demonstration that will familiarize audiences with each of its components.

2 MLflow Overview

To structure the ML development process while leaving users maximum flexibility, we built MLflow around an



Experiment ID: 0 Artifact Location: /Users/matei/demo/mlruns/0

Search Runs: metrics.r2 > 0.2 State: Active

Filter Params: alpha, lr Filter Metrics: rmse, r2

2 matching runs

						Parameters		Metrics		
□	Date	User	Run Name	Source	Version	alpha	l1_ratio	mae	r2	rmse
□	2019-02-07 14:47:46	matei		elasticnet	e8970a	0.3	2.0	70.3	0.201	83.06
□	2019-02-07 17:37:10	matei		elasticnet	89e495	0.5	1.0	53.21	0.364	64.84

Run Command

```
mlflow run file:///Users/matei/mlflow#demo/elasticnet.py -v
89e495e7941cf9e40e6980d14a16bf023ccd4c91 -P alpha=0.5 -P l1_ratio=1.0
```

Notes

ElasticNet regression model trained in Scikit-learn for predicting home rental prices

Parameters

Name	Value
alpha	0.5
l1_ratio	1.0

Metrics

Tags

Artifacts

Figure 1. Left: The MLflow Tracking UI showing several runs in an experiment. Right: clicking each run lists its metrics, artifacts and other output details; users can also include notes about the run.

open interface philosophy. The system defines general interfaces for three core abstractions, each allowing users to bring their own code or workflows:

- **MLflow Tracking** is an API for recording experiment runs, including code used, parameters, input data, metrics, and arbitrary output files. These runs can then be queried through an API or UI.
- **MLflow Models** is a generic format for packaging models, including code and data dependencies, that is compatible with diverse deployment environments.
- **MLflow Projects** is a format for packaging code into reusable projects. Using a YAML configuration file, each project defines its environment (e.g., software libraries required), the code to run, and parameters that can be used to invoke the project programmatically.

3 Demonstration

Setup We will write and execute code using MLflow's Python SDK in an interactive notebook environment. Throughout the demonstration, the audience will submit values that will be introduced into the notebook code. Additionally, we will host an MLflow tracking server in an accessible location (e.g., an Amazon EC2 instance), allowing the audience to interact with the tracking UI on their own machines. For visibility, we request a computer monitor.

Script The audience will use MLflow to train, deploy, and evaluate a machine learning model for image classification. First, audience members will a select a predefined **MLflow project** corresponding to their preferred model architecture, as well as a set of model hyperparameters; we will analyze the project's contents and explain how it leverages the **MLflow tracking** API to record hyperparameters, metrics, and model artifacts. Next, we will train and output an **MLflow model** by executing the specified project. Using our demonstration machine, or by visiting a provided URL on their own laptops, the audience will interact with the MLflow tracking UI to explore the attributes of this model and its training session. Finally, we will deploy the model as a RESTful server for real-time inference; the audience will be able to query the model by making HTTP POST requests to a specified URL. We will evaluate the test accuracy of the model using inputs selected by the audience and display model performance visualizations. Following this live action portion of the demonstration, we will discuss community reception and third party contributions to MLflow, as well as field questions from the audience.

Takeaway Message The demonstration introduces the audience to MLflow's main components. It emphasizes the lightweight, self-contained structure of MLflow's models and projects, indicating to the audience that MLflow can be used to package arbitrary machine learning code written in any language or framework. Additionally, audience interaction with the tracking server demonstrates this component's utility as a centralized repository for models and training sessions across an organization. Finally, the audience will observe that MLflow simplifies the process of deploying and evaluating models in production.

4 References

J. Dunn. Introducing FBLearner Flow: Facebook's AI backbone, 2016. URL `https://code.fb.com/core-data/introducing-fblearner-flow-facebook-s-ai-backbone`.

A. N. Modi, C. Y. Koo, C. Y. Foo, C. Mewald, D. M. Baylor, E. Breck, H.-T. Cheng, J. Wilkiewicz, L. Koc, L. Lew, M. A. Zinkevich, M. Wicke, M. Ispir, N. Polyzotis, N. Fiedel, S. E. Haykal, S. Whang, S. Roy, S. Ramesh, V. Jain, X. Zhang, and Z. Haque. TFX: A TensorFlow-based production-scale machine learning platform. In *KDD 2017*, 2017.

M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, F. Xie, and C. Zumar. Accelerating the machine learning lifecycle with MLflow. *IEEE Data Engineering Bulletin*, 41(4), 2018.