ML flow report



Contents

[1](#_Toc109735879)

[Introduction & Background 1](#_Toc109735880)

[2](#_Toc109735881)

[MLflow Design Philosophy 2](#_Toc109735882)

[2](#_Toc109735883)

[MLflow Lifecycle 2](#_Toc109735884)

[4](#_Toc109735885)

[ML flow Projects 4](#_Toc109735886)

[4](#_Toc109735887)

[ML flow Models 4](#_Toc109735888)

[5](#_Toc109735889)

[ML flow Model Registry 5](#_Toc109735890)

[ML flow Tracking 5](#_Toc109735891)

# 

# Introduction & Background

The goal of using Machine Learning is to optimize metrics (e.g., accuracy. Constantly experiment to improve it. The quality of models depends on input data and tuning parameters. We compare and combine many libraries, model and use a diverse deployment environment. Keeping all these functionalities easy to use and manage, MLflow was introduced.

MLflow is an open machine learning platform.

* It works with popular ML library & language
* Runs the same way anywhere (e.g., any cloud or locally)
* Designed to be useful for 1 or 1000+ person orgs
* Simple. Modular. Easy-to-use.
* Offers positive developer experience to get started

# 

# MLflow Design Philosophy

The working of MLflow is based on 2 different design philosophies which define its way of working.

1. API First Design

* Submit runs, log models, metrics, etc. from popular library & language
* Abstract “model” lambda function that MLflow can then deploy in many places (Docker, Azure ML, Spark UDF)
* Open interface allows easy integration from the community

It is built around Programmatic APIs, REST APIs & CLI

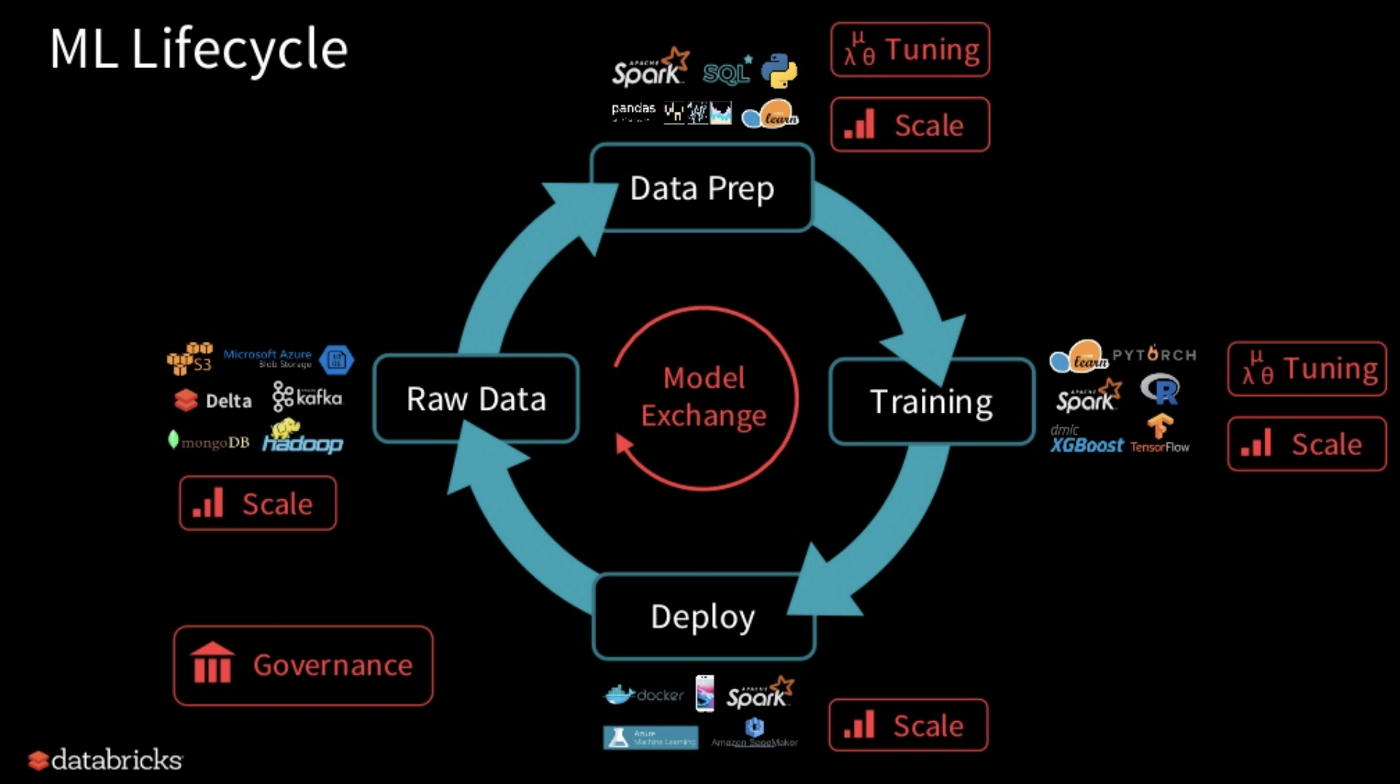
1. Modular Design

* Allow different components individually (e.g., use MLflow’s project format but not its deployment tools
* Not monolithic
* But Distinctive and Selective

Based on distinct components (Tracking/Projects/Models/Registry

# 

# MLflow Lifecycle



ML Lifecycle includes taking the raw data and cleaning it by handling missing and noisy values, transforming it into an organized, prepared data. Tuning and scaling is done on the Data. Different models are used for the training and deployment. Every stage of this life cycle becomes easy using MLflow lifecycle.

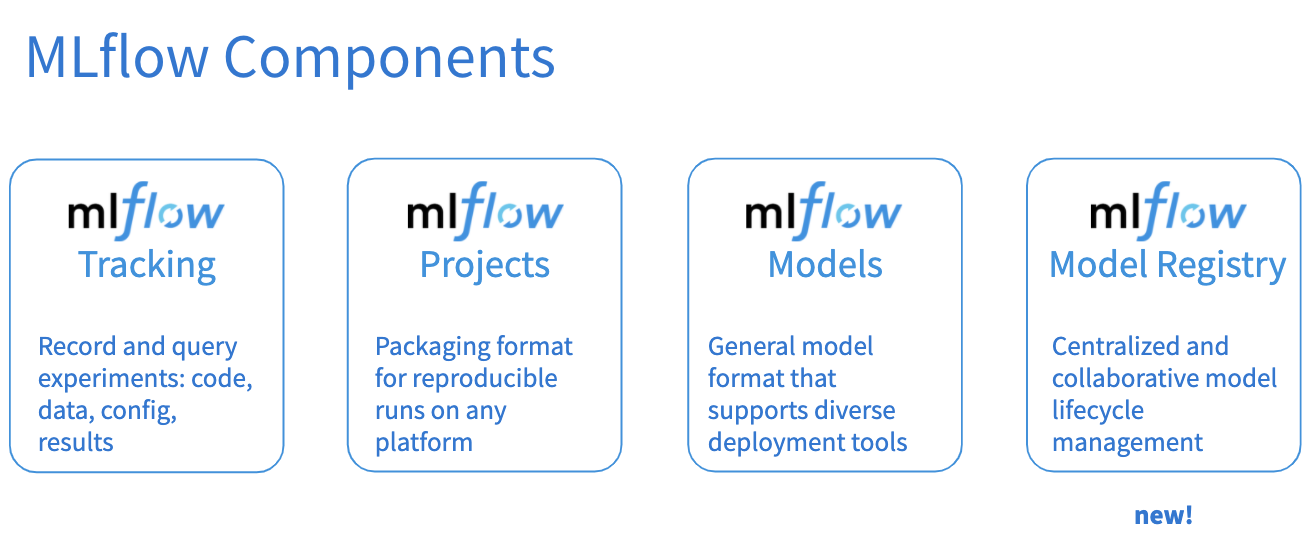
It includes mainly 4 components:

1.Tracking

2.Projects

3.Models

4.Registry



# 

## ML flow Projects

An MLflow Project is a format for packaging data science code in a reusable and reproducible way, based primarily on conventions. In addition, the Projects component includes an API and command-line tools for running projects, making it possible to chain together projects into workflows.

# 

## ML flow Models

An MLflow Model is a standard format for packaging machine learning models that can be used in a variety of downstream tools—for example, real-time serving through a REST API or batch inference on Apache Spark. The format defines a convention that lets you save a model in different “flavors” that can be understood by different downstream tools.

# 

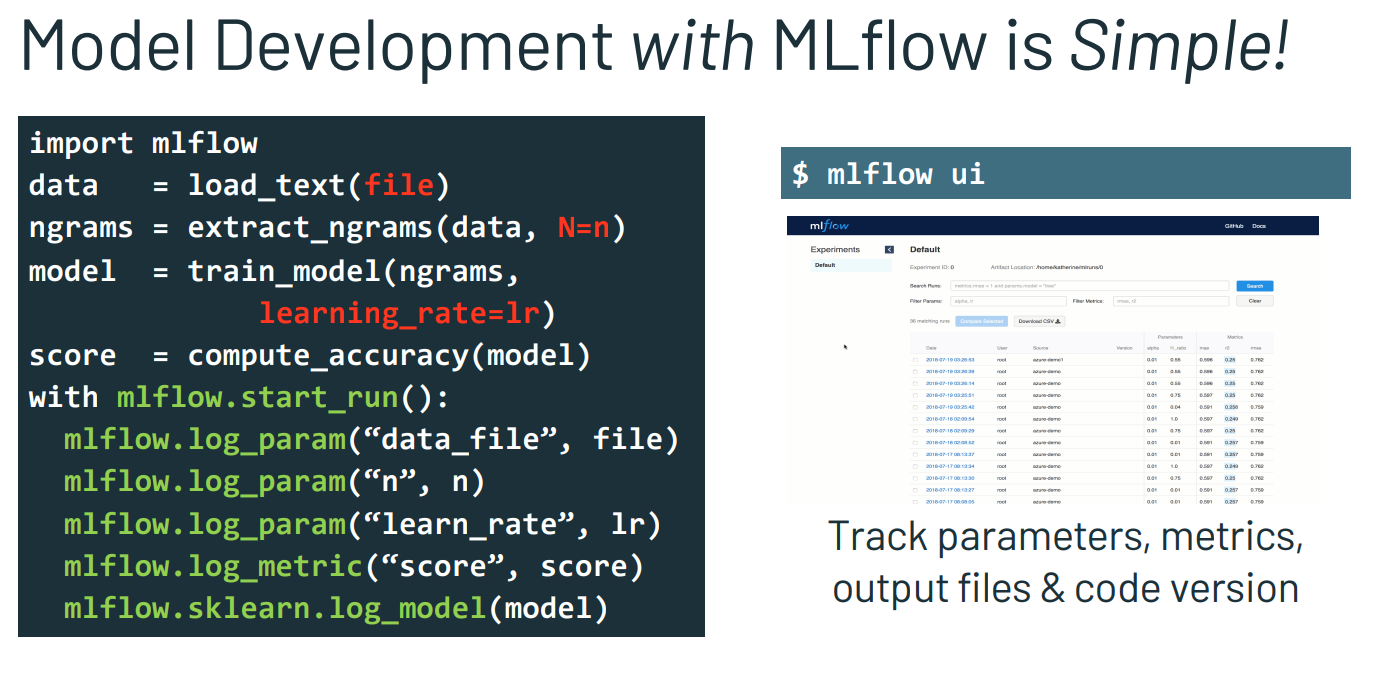
## ML flow Model Registry

The MLflow Model Registry component is a centralized model store, set of APIs, and UI, to collaboratively manage the full lifecycle of an MLflow Model. It provides model lineage (which MLflow experiment and run produced the model), model versioning, stage transitions (for example from staging to production), and annotations.

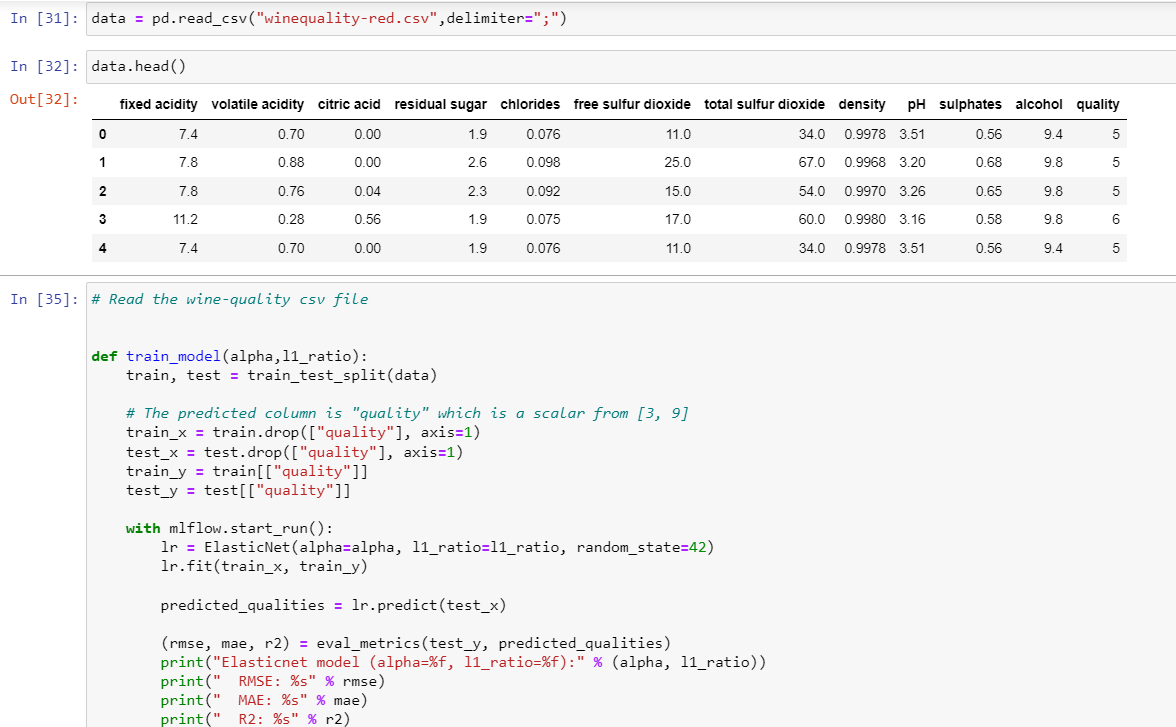
## ML flow Tracking

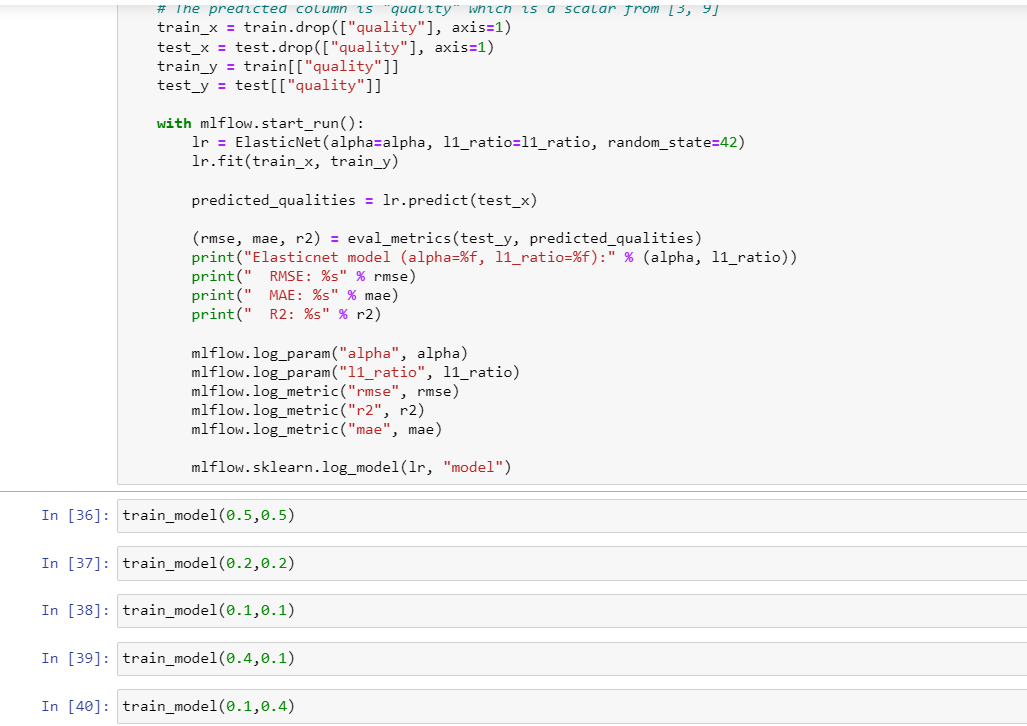
ML flow Tracking is an API and UI for logging parameters, code versions, metrics and output files when running your machine learning code to later visualize them. With a few simple lines of code, you can track parameters, metrics, and artifacts:

* Parameters: key-value inputs to your code
* Metrics: numeric values (can update over time)
* Tags and Notes: information about a run
* Artifacts: files, data, and models
* Source: what code ran?
* Version: what of the code?
* Run: an instance of code that runs by MLflow
* Experiment: {Run, … Run









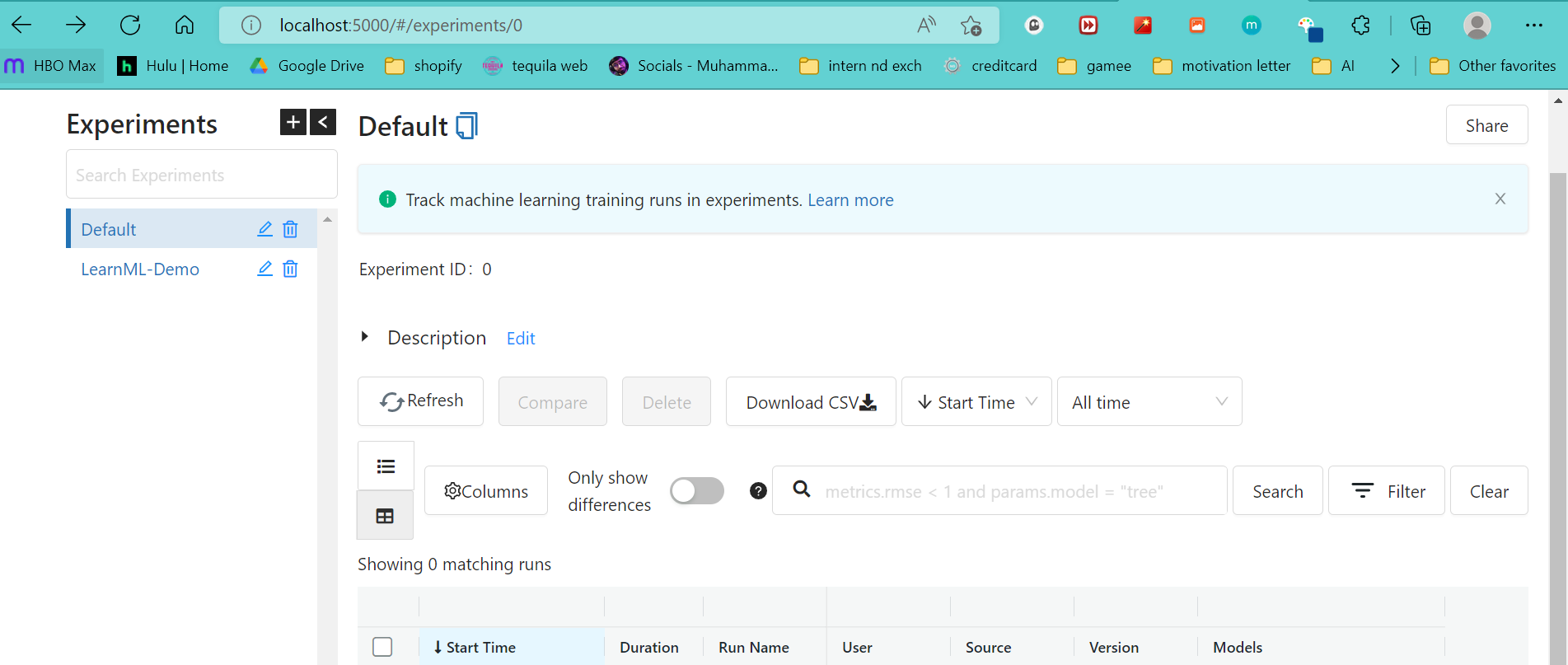
When you run

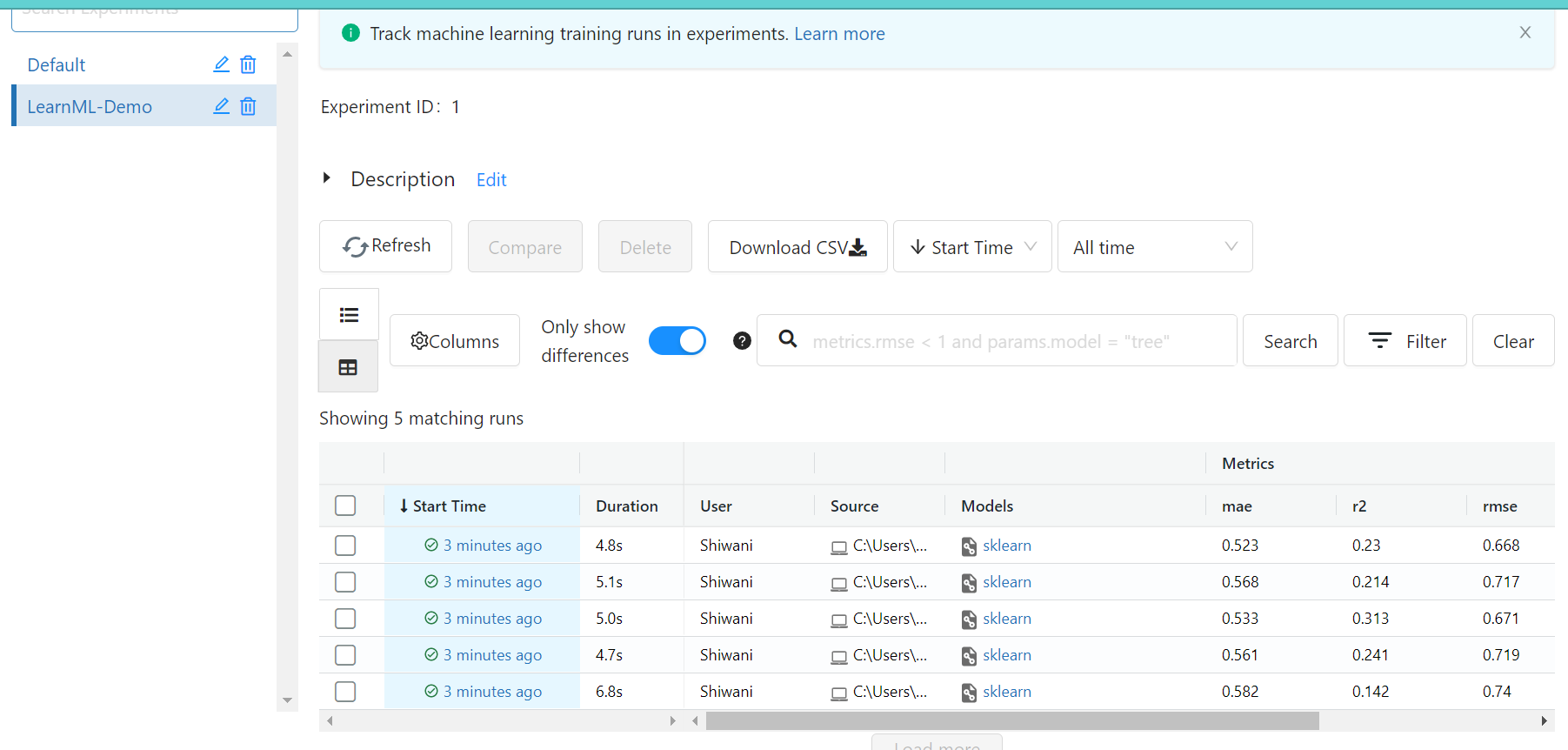
!mlflow ui

Then you can go to your browser and go to

[**http://localhost:5000/**](http://localhost:5000/)

It gives you the UI of the tracking of the project showing the experimentation of ML models





It becomes easy to get notice of all the parameters, time taken and accuracy of a certain model.

