

MLflow 실습 1

1. MLflow 설치

2. MLflow tracking server 띄워보기

1. MLflow 설치

- Prerequisite
 - 터미널 사용이 가능한 os
 - Ubuntu 20.04 사용
 - python 가상 환경
 - conda
 - mlflow models serve 할 때(서빙할 때) 필요
 - 3.8.5 사용
 - (3.6 이상)
 - pip3
- How to Install

```
# 새로운 디렉토리를 하나 생성한 뒤, 이동해주세요
mkdir mlflow-tutorial
cd mlflow-tutorial

# conda 가상환경 세팅
# python 버전 확인
python -V

pip install mlflow==1.20.2

mlflow --version
# mlflow, version 1.20.2
```

2. MLflow tracking server 띄워보기

```
mlflow ui --help
# mlflow tracking server 를 띄웁니다.
```

```
# UI (dashboard) 의 default url 은 http://localhost:5000 입니다.  
# 5000 포트가 열려있는지 확인해주세요.  
# production 용으로는 mlflow ui 대신 mlflow server 를 사용하라는 안내가 출력됩니다.  
  
mlflow server --help  
# mlflow server 는 worker 를 여러 개 띄울 수 있고, prometheus 가 metrics 을 가져갈 수 있도록 엔드포인트를 제공하는 등의 추가적인 기능이 존재합니다.
```

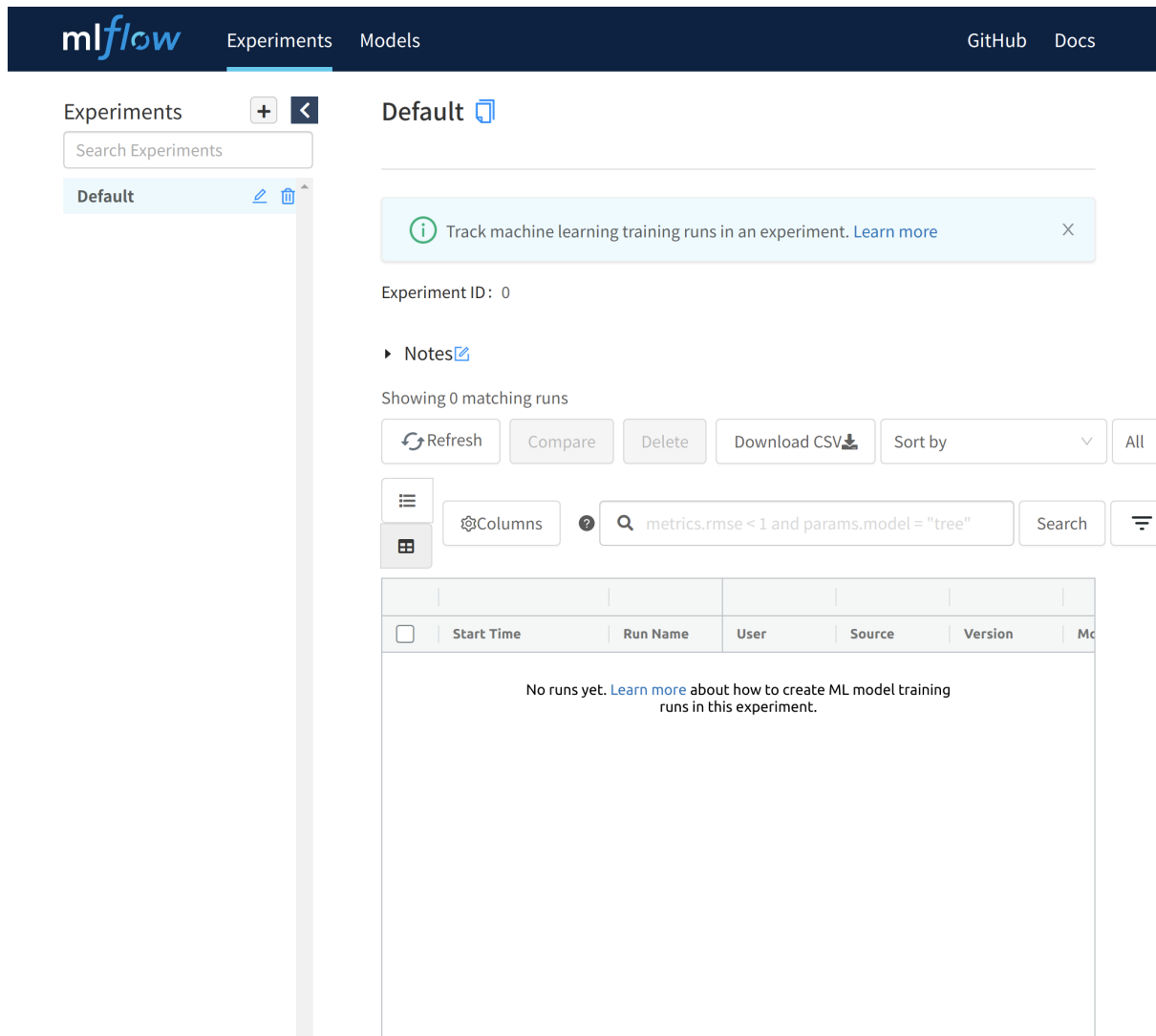
- 이번 강의에서는 로컬에서 띄울 것이고, 사용자가 단 한 명이기에 더 가벼운 `mlflow ui` 로 MLflow tracking server 를 띄우겠습니다.

```
mlflow ui
```

- 노트북에서 5000 번 포트가 사용중이 아니었다면, 다음과 같이 <http://127.0.0.1:5000> 로 접속하면 된다는 안내가 출력됩니다.
 - 일반적인 경우라면 127.0.0.1 과 localhost 는 동일하게 여러분의 컴퓨터 자신을 접속하는 주소를 의미합니다.

```
[34550] [INFO] Starting gunicorn 20.1.0  
[34550] [INFO] Listening at: http://127.0.0.1:5000 (34550)  
[34550] [INFO] Using worker: sync  
[34551] [INFO] Booting worker with pid: 34551
```

- 해당 터미널을 켜둔 상태에서, <http://localhost:5000> 으로 접속합니다.
 - 터미널이 꺼지면 MLflow tracking server 도 종료됩니다.
- 아래와 같은 화면이 나타난다면, 정상적으로 MLflow tracking server 가 생성된 것을 의미합니다.



- 다른 터미널을 열어서, 동일한 디렉토리로 이동하여, `mlruns` 라는 디렉토리가 생성된 것을 확인합니다.
 - `mlflow ui` 실행 시 `--backend-store-uri`, `--default-artifact-root` 옵션을 주지 않은 경우, `mlflow ui` 를 실행한 디렉토리에 `mlruns` 라는 디렉토리를 생성한 뒤, 이곳에 실험 관련 데이터를 저장하게 됩니다.

```
cd mlflow-tutorial

ls
cd mlruns
cat 0/meta.yaml
# 무언가 채워진 것을 확인할 수 있습니다.
```

- 이 디렉토리만 존재한다면, `mlflow ui` 터미널을 종료한 뒤, 다시 실행시켜도 동일한 화면을 볼 수 있게 됩니다.