



MLFlow 101

(With Kaggle - Titanic dataset)

MLFlow 소개

- 머신러닝 생애주기(ML Lifecycle)를 관리하는 오픈 소스 플랫폼
 - 실험 (Experimentation)
 - 재사용성 (Reproducibility)
 - 배포 (Deployment)
 - 중앙 모델 레지스트리 (Central Model Registry)

“I build 100s of models/day to lift revenue, using any library: MLib, PyTorch, R, etc. There’s no easy way to see what data went in a model from a week ago, tune it and rebuild it.”

-- Chief scientist at ad tech firm





Tracking

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

실험 관련 파라미터와 결과 로깅을 위한 API, UI

Projects

Packaging format for reproducible runs on any platform

머신 러닝 코드 재사용과 재현 가능한 형태로 패키징

Models

General format for sending models to diverse serving environments (deploy tools)

모델 파일과 코드 저장 및 배포와 관련된 기능 제공

Registry

Store, annotate, discover, and manage models in central repository

MLFlow 모델 생애주기에 사용되는 요소 중앙 저장소

MLFlow: Tracking

mlflow2.5.0

Experiments

Models

GitHub

Docs

Experiments

Search Experiments

✓

titanic

titanic

Provide Feedback

Experiment ID: 452242112812505426 Artifact Location: /mnt/d/project/git/ianychoi/MLflow-101/tracking_server/452242112812505426

> Description Edit

Table view

Chart view

Artifact view

metrics.rmse < 1 and params.model = "tree"

Time created

State Active

Refresh

Sort: Created

Columns

Expand rows

<input type="checkbox"/>	<input type="checkbox"/>	Run Name	Created	Dataset	Duration	Source	Models	
<input type="checkbox"/>	<input type="checkbox"/>	● persistent-conch-655	✓ 21 minutes ago	-	11.2s	mlflow_tr...	sklearn	+
<input type="checkbox"/>	<input type="checkbox"/>	● fortunate-chimp-597	✓ 38 minutes ago	-	10.9s	mlflow_tr...	sklearn	

MLFlow: Tracking 실습

- 타이타닉 데이터를 사용한 ML 모델 트래킹
- /MLflow_tutorial/ML/mlflow_tracking.py

Titanic - Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics.

Titanic Data Dictionary

- *Survived* : 0 = 사망, 1 = 생존
- *Pclass* : 1 = 1 등석, 2 = 2 등석, 3 = 3 등석
- *Sex* : male = 남성, female = 여성
- *Age* : 나이
- *SibSp* : 타이타닉 호에 동승한 자매 / 배우자의 수
- *Parch* : 타이타닉 호에 동승한 부모 / 자식의 수
- *Ticket* : 티켓 번호
- *Fare* : 승객 요금
- *Cabin* : 방 호수
- *Embarked* : 탑승지, C = 세르부르, Q = 퀸즈타운, S = 사우스햄프턴

<https://www.kaggle.com/competitions/titanic/data>

```
if __name__ == '__main__':
    # mlflow.set_tracking_uri("http://127.0.0.1:5000")
    # exp_info = MlflowClient().get_experiment_by_name("titanic")
    # exp_id = exp_info.experiment_id if exp_info else MlflowClient().create_experiment("titanic")
    # with mlflow.start_run(experiment_id=exp_id) as run:

    mlflow.set_experiment('titanic')
    with mlflow.start_run() as run:
        # Directory
        train_dir = "train.csv"
        test_dir = "test.csv"

        # Flow
        train, test = load_data(train_dir, test_dir)
        train_x, train_y, test_x, test_y = pre_processing(train, test)
        model = build_model(train_x, train_y)
        score = evaluation(model, test_x, test_y)

        pred_x = model.predict(test_x)

        # mlflow log_param으로 원하는 항목을 로깅
        mlflow.log_param("train", train_dir)
        mlflow.log_param("train num", len(train_x))
        mlflow.log_param("class", collections.Counter(train_y))
        mlflow.log_param("class num", len(set(train_y)))

        # 실험 결과 metric
        mlflow.log_metric("f1 score", score)
        # 데이터 저장
        mlflow.log_artifact(train_dir)
        # 모델 저장 및 모델 저장 폴더명 지정
        mlflow.sklearn.log_model(model, "titanic_model")
```

MLFlow: Tracking 실습

- mlruns 폴더 내 구조

```
(.venv) ian@LOCK-PC:/mnt/d/project/git/ianychoi/MLflow-101/ML/mlruns$ tree
.
├── meta.yaml
├── 991609055927250309
│   ├── 254c3807cd0854cd89542bc079c9881fb
│   │   ├── artifacts
│   │   │   ├── titanic_model
│   │   │   │   ├── MLmodel
│   │   │   │   ├── conda.yaml
│   │   │   │   ├── model.pkl
│   │   │   │   ├── python_env.yaml
│   │   │   │   └── requirements.txt
│   │   └── train.csv
│   ├── meta.yaml
│   ├── metrics
│   │   └── f1 score
│   ├── params
│   │   ├── class
│   │   ├── class num
│   │   ├── train
│   │   └── train num
│   ├── tags
│   │   ├── mlflow.log-model.history
│   │   ├── mlflow.runName
│   │   ├── mlflow.source.name
│   │   ├── mlflow.source.type
│   │   └── mlflow.user
│   └── meta.yaml
└── models
```

9 directories, 19 files

- mlruns 폴더 구조
 - 실험 (Experiments)
 - 실행 (Runs): log param, metric, artifacts, ...
 - 실행 (Runs): log param, metric, artifacts, ...
 - 실행 (Runs): log param, metric, artifacts, ...

- artifacts: 모델과 데이터

```
# 데이터 저장
mlflow.log_artifact(train_dir)

# 모델 저장 및 모델 저장 폴더명 지정
mlflow.sklearn.log_model(model, "titanic_model")
```

- metrics: 실험 결과를 측정하는 값

```
# 실험 결과 metric
mlflow.log_metric("f1 score", score)
```

- params: 로깅하고 싶은 파라미터 또는 실험값

```
# mlflow log_param으로 원하는 항목을 로깅
mlflow.log_param("train", train_dir)
mlflow.log_param("train num", len(train_x))
mlflow.log_param("class", collections.Counter(train_y))
mlflow.log_param("class num", len(set(train_y)))
```

MLFlow: Tracking 실습

- UI 확인
 - \$ mlflow ui -h {IP} -p {PORT}

The screenshot displays the MLFlow 2.5.0 web interface. The 'Experiments' tab is active, showing a list of experiments. The 'titanic' experiment is selected and highlighted with a blue box, with the label '실험 이름' (Experiment Name) next to it. A blue arrow points from this box to a detailed view of the 'bold-croc-373' run, which is also highlighted with a blue box and the label '실행 이름' (Run Name).

The detailed view of the 'bold-croc-373' run shows the following information:

- Run ID: 254c3097ed054cd89542bc079c9081fb
- Date: YYYY-MM-DD hh:mm:ss
- Duration: 11.3s
- Status: FINISHED
- Description: Edit
- Datasets: Edit
- Parameters (4):

Name	Value
class num	2
class	Counter({0: 490, 1: 311})
train num	801
train	train.csv

The main interface also shows a table of runs with columns: Run Name, Created, Dataset, Duration, Source, and Models. The 'bold-croc-373' run is listed in this table.



MLFlow: Tracking 실습



- Exercise
 - test_size 등을 바꿔서 실행해보기
 - 여러 번 실행해보기
 - 어떤 로깅을 하는 게 좋을지 생각해보기

mlflow 2.5.0 Experiments Models GitHub Docs

Experiments

Search Experiments

☐ Default  

☒ titanic  


titanic

[Provide Feedback](#)


Experiment ID: 991609055927250309 Artifact Location: file:///mnt/d/project/git/ianychoi/MLflow-101/ML/mlruns/991609055927250309

> Description [Edit](#)


















Table view Chart view Artifact view



Time created ▾ State Active ▾

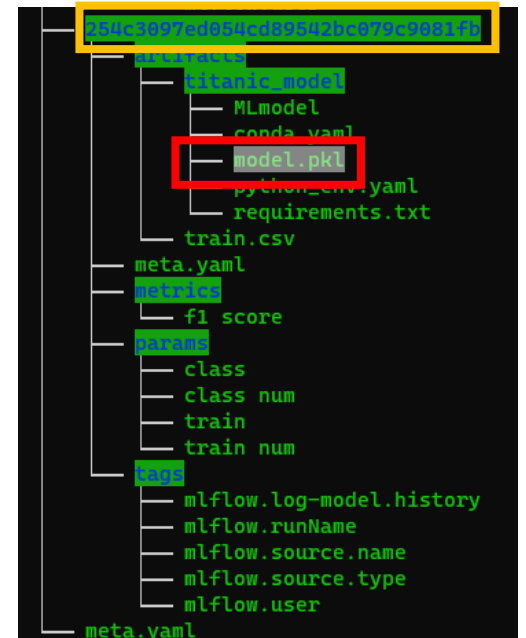
 Refresh

Sort: Created ▾ Columns ▾ ☐ Expand rows

<input type="checkbox"/>		Run Name	Created		Dataset	Duration	Source	Models
<input type="checkbox"/>		 sassy-smelt-168	 3 minutes ago	-	-	10.9s	 mlflow_tr...	 sklearn
<input type="checkbox"/>		 sneaky-worm-455	 4 minutes ago	-	-	12.3s	 mlflow_tr...	 sklearn
<input type="checkbox"/>		 bold-croc-373	 33 minutes ago	-	-	11.3s	 mlflow_tr...	 sklearn

MLFlow: Models – 인퍼런스(inference) 실습

- MLflow에 저장된 타이타닉 모델 ML 가져와서 추론하기
- /MLflow_tutorial/ML/mlflow_inference.py
- \$ python mlflow_inference.py



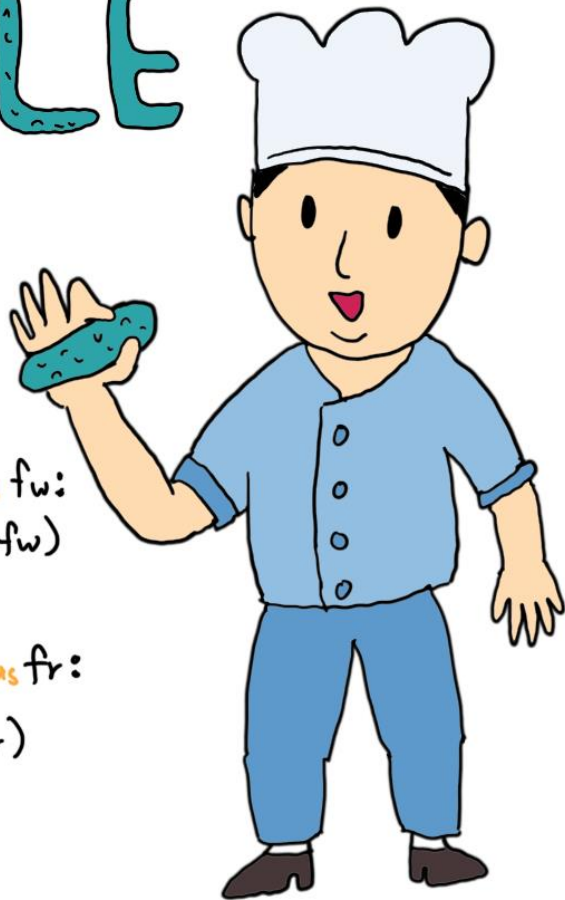
```
import mlflow
import pandas as pd

if __name__ == '__main__':
    # runs:/{Run ID}/titanic_model 이므로 {Run ID} 부분을 실행 대상 ID로 변경한다.
    logged_model = 'runs:254c3097ed054cd89542bc079c9081fb/titanic_model'

    loaded_model = mlflow.pyfunc.load_model(logged_model)
    test_x = pd.DataFrame({"Pclass": [2, 1], "Sex": [0, 1], "Fare": [3.3211, 3.3211], "SibSp": [3, 3], "Parch": [3, 3]})
    print(loaded_model.predict(test_x))
```

참고: Python 피클 (pickle) – 모델에 대해 model.pkl 파일명으로 관리

PICKLE



```
import pickle  
my_list = ['a', 'b', 'c']
```

```
with open("data.pickle", "wb") as fw:  
    pickle.dump(my_list, fw)
```

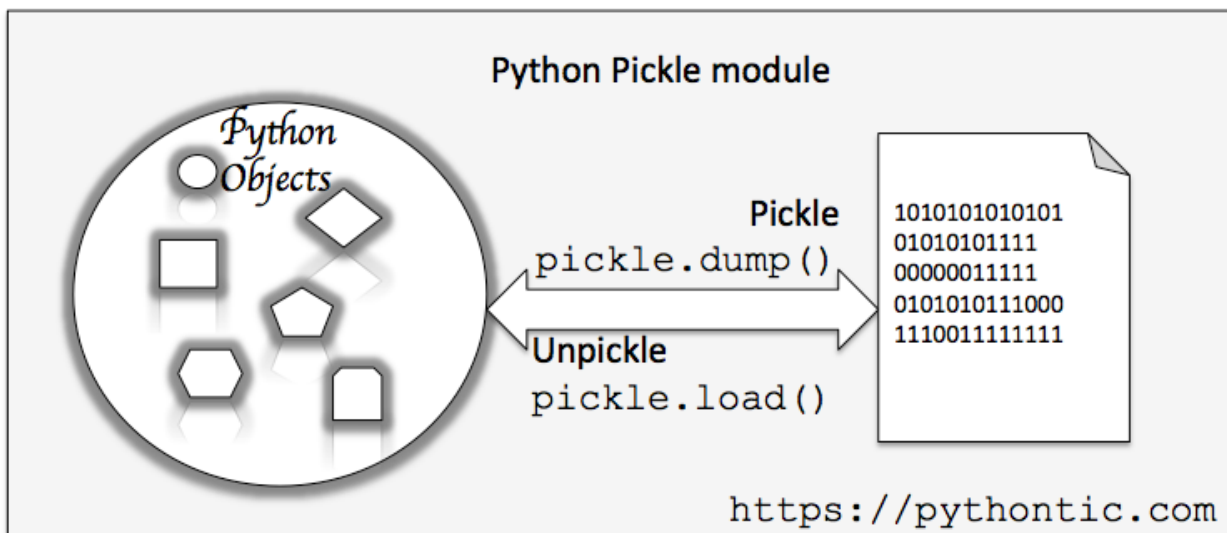
```
with open("data.pickle", "rb") as fr:  
    data = pickle.load(fr)
```

```
print(data)  
# ['a', 'b', 'c']
```

`pickle` — Python object serialization

Source code: [Lib/pickle.py](https://github.com/python/cpython/blob/main/Lib/pickle.py)

The `pickle` module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a [binary file](#) or [bytes-like object](#)) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as “serialization”, “marshalling,” [1] or “flattening”; however, to avoid confusion, the terms used here are “pickling” and “unpickling”.



<https://korbillgates.tistory.com/173>

<https://benn.tistory.com/43>

MLFlow: Models - API 서버 배포

- 모델을 API로 서빙
- `$ mlflow models serve -m runs://{Run ID}/titanic_model --no-conda --port 5001`

```
(.venv) ian@LOCK-PC:/mnt/d/project/git/ianychoi/MLflow-101/ML$ mlflow models serve -m runs:/254c3097ed054cd89542bc079c9081fb/titanic_model --no-conda -  
-port 5001  
2023/08/09 01:59:51 INFO mlflow.models.flavor_backend_registry: Selected backend for flavor 'python_function'  
2023/08/09 01:59:51 INFO mlflow.pyfunc.backend: === Running command 'exec gunicorn --timeout=60 -b 127.0.0.1:5001 -w 1 ${GUNICORN_CMD_ARGS} -- mlflow.p  
yfunc.scoring_server.wsgi:app'  
[2023-08-09 01:59:52 +0900] [3233] [INFO] Starting gunicorn 20.1.0  
[2023-08-09 01:59:52 +0900] [3233] [INFO] Listening at: http://127.0.0.1:5001 (3233)  
[2023-08-09 01:59:52 +0900] [3233] [INFO] Using worker: sync  
[2023-08-09 01:59:52 +0900] [3235] [INFO] Booting worker with pid: 3235
```

- 다른 터미널을 통해 API 호출을 직접 실행하여 서빙한 모델 API 테스트
- `$ curl http://<IP>:<PORT>/`

```
(.venv) ian@LOCK-PC:/mnt/d/project/git/ianychoi/MLflow-101/ML$ curl -d '{"dataframe_split": {"columns": ["Pclass", "Sex", "Fare", "SibSp", "Parch"], "da  
ta": [[1, 2, 3, 2, 2], [1, 2, 4, 5, 6]]}' -H 'Content-Type: application/json' -X POST 127.0.0.1:5001/invocations  
{ "predictions": [1, 1] }(.venv) ian@LOCK-PC:/mnt/d/project/git/ianychoi/MLflow-101/ML$
```

MLFlow: Projects

- 모델 실험 + 실행 환경을 프로젝트 단위로 관리
- 실습을 위해서는 실행 환경을 관리하는 추가 환경을 필요로 함
- MLFlow에서 지원하는 프로젝트 환경

Virtualenv environment (preferred)

Docker container environment

Conda environment

System environment

MLproject File

You can get more control over an MLflow Project by adding an MLproject file, which is a text file in YAML syntax, to the project's root directory. The following is an example of an MLproject file:

```
name: My Project

python_env: python_env.yaml
# or
# conda_env: my_env.yaml
# or
# docker_env:
#   image: mlflow-docker-example

entry_points:
  main:
    parameters:
      data_file: path
      regularization: {type: float, default: 0.1}
    command: "python train.py -r {regularization} {data_file}"
  validate:
    parameters:
      data_file: path
    command: "python validate.py {data_file}"
```

MLFlow: Registry – Models

- “모델”을 추적 단위로 하여 관리하기 위한 용도
- UI에서 쉽게 중앙 관리 가능

mlflow 2.5.0 Experiments Models GitHub Docs

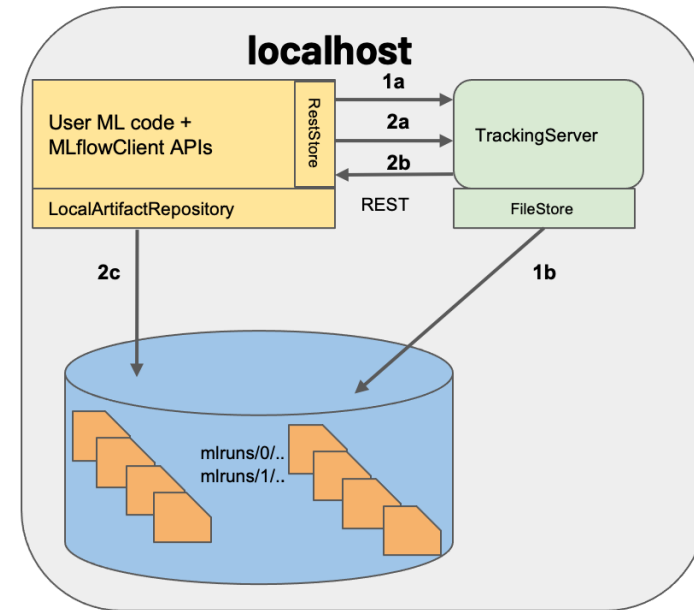
Registered Models ⓘ [Create Model](#)

Filter registered models... ⓘ 🔍

Name	Latest version	Staging	Production	Created by	Last modified	Tags
titanic	—	—	—		2023-08-09 03:...	version: default

MLFlow: Registry - Tracking

- 파일 또는 DB를 중앙 저장소로 사용하여 MLOps 추적 과정 및 기록을 관리하기 위한 용도
- 터미널1



```
(.venv) ian@LOCK-PC:/mnt/d/project/git/ianychoi/MLflow-101/tracking_server$ mlflow server --backend-store-uri file:/mnt/d/project/git/ianychoi/MLflow-101/tracking_server --default-artifact-root /mnt/d/project/git/ianychoi/MLflow-101/tracking_server --port 5000
[2023-08-09 03:23:43 +0900] [25167] [INFO] Starting gunicorn 20.1.0
[2023-08-09 03:23:43 +0900] [25167] [INFO] Listening at: http://127.0.0.1:5000 (25167)
[2023-08-09 03:23:43 +0900] [25167] [INFO] Using worker: sync
[2023-08-09 03:23:43 +0900] [25169] [INFO] Booting worker with pid: 25169
```

- 터미널2: 아래와 같이 주석 변경 후 실행 & UI 확인
\$ python mlflow_tracking.py

```
42 if __name__ == '__main__':
43     mlflow.set_tracking_uri("http://127.0.0.1:5000")
44     exp_info = MlflowClient().get_experiment_by_name("titanic")
45     exp_id = exp_info.experiment_id if exp_info else MlflowClient().create_experiment("titanic")
46     with mlflow.start_run(experiment_id=exp_id) as run:
47
48         # mlflow.set_experiment('titanic')
49         # with mlflow.start_run() as run:
50             # Directory
51             train_dir = "train.csv"
52             test_dir = "test.csv"
53
```

The screenshot shows the MLFlow UI with the 'titanic' experiment selected. The interface includes a search bar, filters, and a table of runs.

Run Name	Created	Dataset	Duration	Source	Models
● persistent-conch-655	14 minutes ago	-	11.2s	mlflow_tr...	sklearn
● fortunate-chimp-597	32 minutes ago	-	10.9s	mlflow_tr...	sklearn