

MLflow 튜토리얼

일시 2022년 03월 06일
장소 Google Meet
발표 이정훈



PROFILE

이정훈 Jung Hoon

인터파크 특집사 개발팀
가짜연구소 아카데미&커뮤니티 빌더

01

MLOps

MLOps 필요성 및 정의

01. MLOps

MLOps 왜 필요한가?

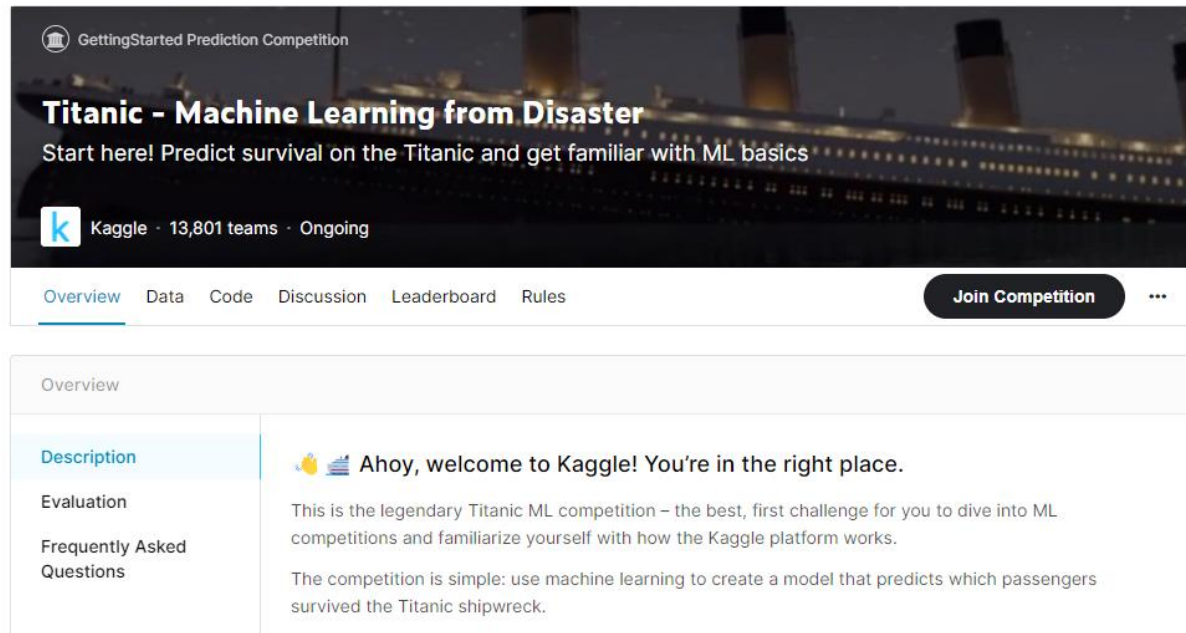
- 2018년 인공지능을 처음 접하고 공부했을 때 생각한 나의 모습.
 - Loss를 빠르게 수렴시키기 위한 방법은 뭐가 있을까?
 - 학습 데이터가 정규분포를 따르고 있는가?
 - 역전파 수식이 어떻게 되지?
 - Naïve Bayes, SVM, XGBoost 같은 알고리즘의 수식은 뭐지?
 - 어떤 Feature를 뽑아야 할까?



01. MLOps

MLOps 왜 필요한가?

- Kaggle 타이타닉 competition에 참여
- 주어진 데이터를 활용해 모델을 학습한 후 성능 체크
- 올라가는 리더보드 순위를 보며 뿌듯



The screenshot shows the Kaggle competition page for "Titanic - Machine Learning from Disaster". The header includes the competition title, a subtitle "Start here! Predict survival on the Titanic and get familiar with ML basics", and the Kaggle logo with "13,801 teams · Ongoing". Below the header is a navigation bar with links: Overview, Data, Code, Discussion, Leaderboard, and Rules. A "Join Competition" button is on the right. The main content area has a tabbed interface with "Overview" selected. The "Description" tab is active, showing a welcome message from the competition host and a brief overview of the competition's goal: to predict which passengers survived the Titanic shipwreck.

GettingStarted Prediction Competition

Titanic - Machine Learning from Disaster

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

Kaggle · 13,801 teams · Ongoing

[Overview](#) [Data](#) [Code](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Join Competition](#) ...

Overview

Description

Ahoy, welcome to Kaggle! You're in the right place.

This is the legendary Titanic ML competition – the best, first challenge for you to dive into ML competitions and familiarize yourself with how the Kaggle platform works.

The competition is simple: use machine learning to create a model that predicts which passengers survived the Titanic shipwreck.

01. MLOps

MLOps 왜 필요한가?

- Kaggle 리더보드 순위가 올라가고 여러 알고리즘을 사용해 모델을 만들다보면 나도 이제 인공지능 좀 할 줄 아는 것 같은데..?란 생각이 든다.
- 그런데 여기서 마주하는 문제.. 만든 모델 배포를 어떻게 하지???



01. MLOps

MLOps 왜 필요한가?

- 에이~ 그냥 프로토 타입이고 연구실에서 하는건데 모델 파일 불러와서 Flask로 API 서버 띄우자~
- 어..? 근데 지속적으로 모델을 재배포해야 하는데.. 너무 번거로워 ㅠㅠ
- 요청(request)이 많아 서버를 늘려야하는데 서버 세팅 일일이 다하고 Flask를 또 띄워야 하나??
- 연구실에 모델 배포 담당하는 사람이 나 혼자네.. ㅠㅠ
- 관리하는 모델의 종류와 개수가 많아졌다. 이걸 나 혼자 하라고..? ㅠㅠ
- 모델 배포를 누구나 할 수 있게 자동화 할 수 있을까?

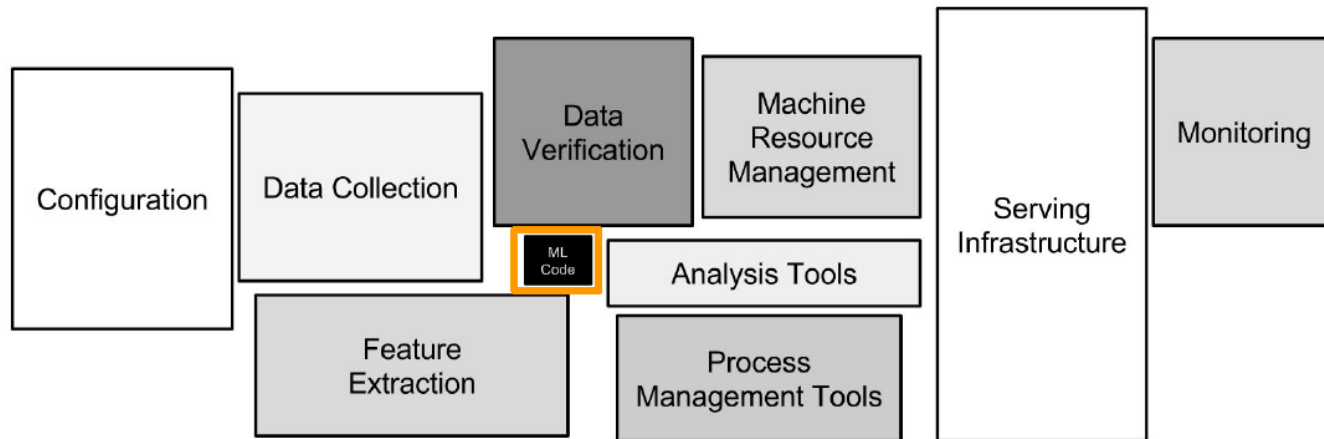


Flask
web development,
one drop at a time

01. MLOps

MLOps 왜 필요한가?

- ML 프로젝트에서 알고리즘과 모델은 일부분일 뿐이다.
- 데이터 버전 관리 어떻게 하지?
- 실험이 너무 많은데 어떻게 기록하지?
- 학습이 끝난 모델 파일들을 어떻게 관리하지?
- 여러 개의 서버에 ML모델을 어떻게 간편하게 배포하지?



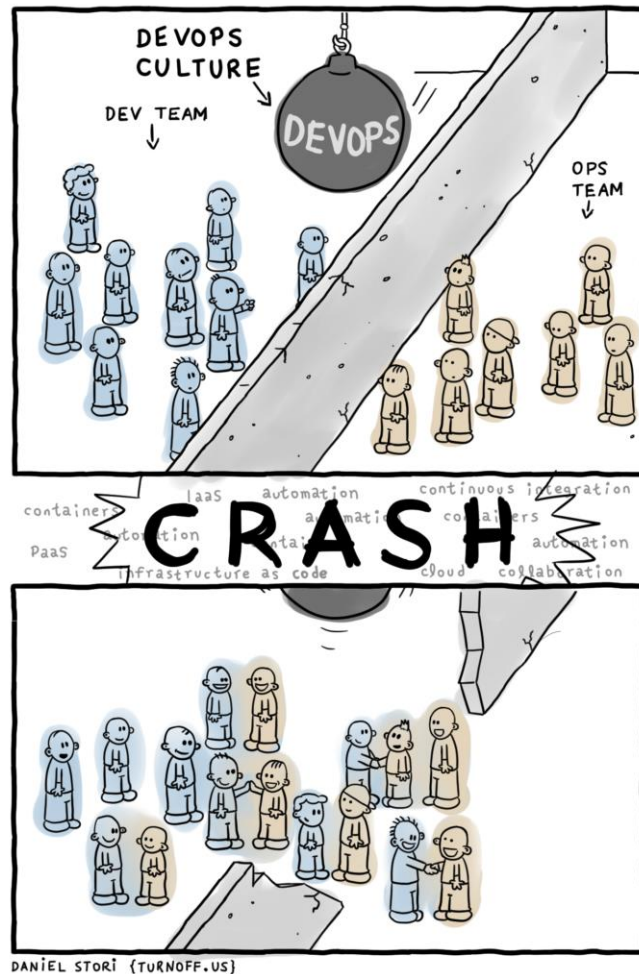
01. MLOps

DevOps 정의

- 개발팀 업무 : 서비스 개발, 유지 보수, QA 반영
- 운영팀 업무 : 개발팀에서 개발하 서비스 배포 및 모니터링
- 과거 서로의 영역은 분리되어 있었으나 개발과 운영을 같이하는

DevOps 등장

- DevOps 정의 : 개발자는 코드 관련된 수정을 할 것이고, 운영자는 최신의 모델로 업데이트 해야하며, QA는 그것에 대해서 평가를 해야한다. 이러한 과정을 묶고 자동화하려는 기술/시스템을 DevOps라고 한다.



01. MLOps

DevOps 장점

- 속도 향상 : 팀에서 서비스를 주도적으로 운영하여 수정된 코드들을 더 빠르게 릴리스.
- 신속한 제공 : CI(Continuous Integrity) 및 CD(Continuous Delivery)등을 통해 빌드에서 배포까지 자동화시켜 릴리스의 빈도와 속도를 개선하여 제품을 더 빠르게 업데이트. 새로운 기능 및 버그 수정속도가 빨라 고객의 요구에 더 빠르게 대응.
- 안정성 : CI/CD를 통해 변경사항을 안전하게 작동하는지 업데이트마다 테스트 해주어 애플리케이션에 안정성 및 인프라 변경의 품질의 보장해준다.
- 확장성 : 규모에 따라 인프라와 개발 프로세스를 자동화하여 시스템을 효율적으로 관리. 개발자와 시스템 관리자가 수동으로 리소스를 설정 및 구성할 필요 없이 프로그래밍 방식으로 큰 규모로 인프라와 상호 작용,
- 협업 강화 : 개발팀과 운영팀은 서로 긴밀하게 협력하며 효과적인 팀을 구축.

01. MLOps

DevOps 도구



01. MLOps

MLOps 정의

- ML(Machine Learning)을 이용한 서비스에서는 CI/CD 뿐만 아니라 지속적 학습(CT)이 보장되어야 한다. 예를 들어, 학습 데이터가 추가 된다면, 분류해야 하는 레이블이 늘어나는 등 다양한 학습에 관한 시나리오들이 있다. 이러한 학습은 이전의 DevOps에서는 생각하지 못했던 구조이기 때문에 이 부분에 대해서 추가한 시스템을 MLOps라고 부른다.
- 다시 한번 정의하면, MLOps는 ML 시스템 개발(Dev)과 ML 시스템 운영(Ops)을 통합하는 것을 목표로 하는 ML 엔지니어링 문화 및 방식이며, 이를 통해 통합, 테스트, 출시, 배포, 인프라 관리 등을 할 수 있으면, ML 시스템을 구성하는 모든 단계에서 자동화 및 모니터링을 지원할 수 있다.

02

MLflow

MLflow Tracking, Registry, Models 필요성 및 정의

02. MLflow

MLflow 소개

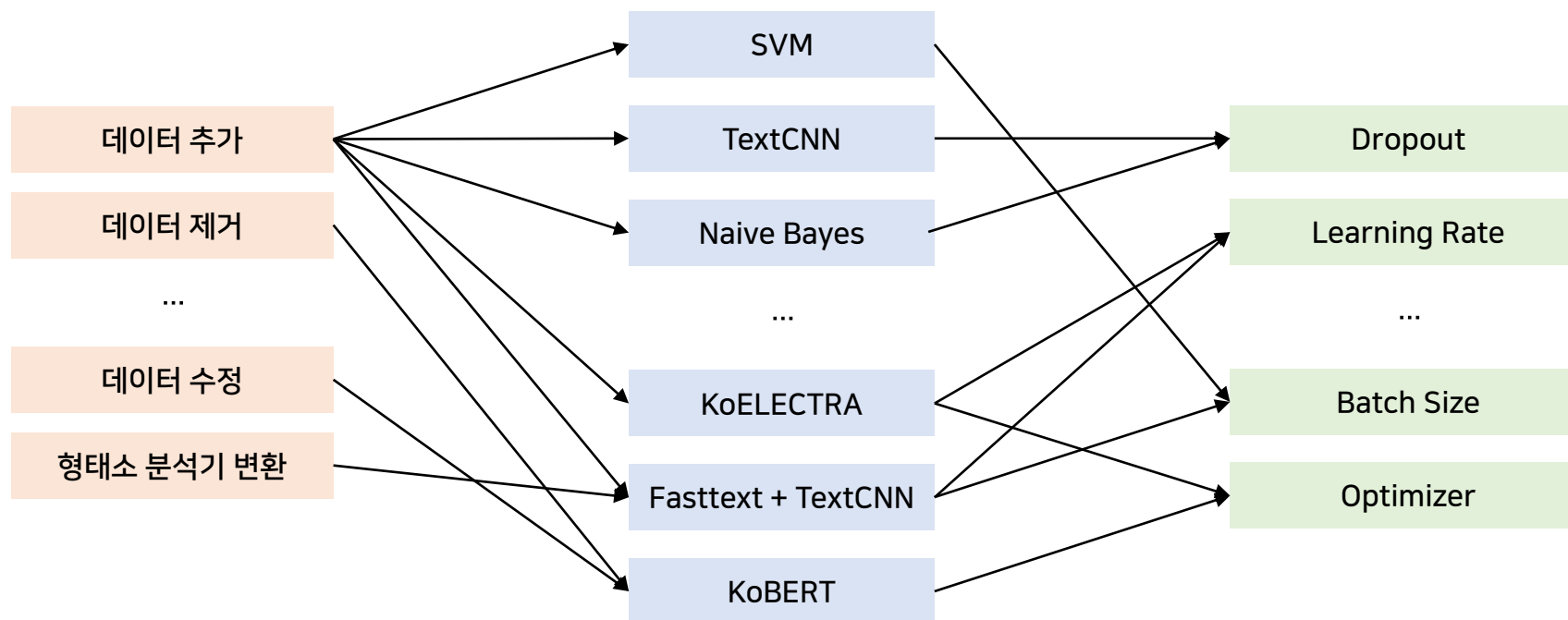
- MLflow : End-to-End machine learning lifecycle을 관리하는 오픈소스 플랫폼.
모든 기계 학습 라이브러리 및 프로그래밍 언어와 함께 사용 가능
- MLflow 주요 기능
 - MLflow Tracking : 파라미터와 결과를 비교하기 위해 실험 결과를 저장
 - MLflow Projects : 머신러닝 코드를 재사용하고 재현 가능한 형태로 포장.
포장된 형태를 다른 데이터 사이언티스트가 사용하거나 프로덕션에 반영
 - MLflow Models : 다양한 ML 라이브러리에서 모델을 관리하고 배포, Serving, 추론



02. MLflow

MLflow Tracking 필요성

- 프로젝트 요청 : 영화 리뷰 긍부정을 분류하는 모델
- 수많은 조건들로 실험을 반복하며 발생하는 accuracy, f1_score, train_loss, test_loss, parameter들을 기록해야 한다.



02. MLflow

MLflow Tracking 필요성

- 갯셀에 실험 내용을 순서대로 기록하자.
- 테이블 구조라 정리 및 관리가 편하네?!

	A	B	G	H	I	J	K	L	M
83	10/1/2016	39							
84	11/1/2016	53							
85	12/1/2016	48							
86	1/1/2017	52							
87	2/1/2017	47							
88	3/1/2017	59							
89	4/1/2017	57							
90	5/1/2017	62							
91	6/1/2017	42							
92	7/1/2017	35							
93	8/1/2017	26							
94	9/1/2017	37							
95	10/1/2017	33							
96	11/1/2017	48							
97	12/1/2017	47							

1. 단순평균		2B. Forecast.ETS		3. Machine Learning	
Avg.	MAE	Avg.	MAE	Avg.	MAE
48.138889	11.43%	43.947599	8.80%	#DIV/0!	100.00%
56.166667	-8.01%	54.640275	-1.20%		100.00%
58.833333	-25.18%	54.640275	-16.26%		100.00%
61.333333	-3.95%	55.200186	6.44%		100.00%
53.833333	5.56%	52.418926	8.04%		100.00%
61.5	0.81%	62.279116	-0.45%		100.00%
46	-9.52%	42.851433	-2.03%		100.00%
36	-2.86%	32.917518	5.95%		100.00%
34	-30.77%	31.085153	-19.56%		100.00%
32.5	12.16%	29.969802	19.00%		100.00%
42.166667	-27.78%	38.693661	-17.25%		100.00%
49.333333	-2.78%	43.609969	9.15%		100.00%
50.666667	-7.80%	47.153929	-0.33%		100.00%

02. MLflow

MLflow Tracking 필요성

- 어 근데.. 모델 실험을 76번 해야하네..? 76번의 실험 기록과 결과를 모두 적어야 한다고? $\pi\pi$
- 아.. 씨.. 중간에 기록 잘 못 적었다.. $\pi\pi$
- 자동화하는 방법 없을까? 그리고 시각화 좋은 UI로 기록들을 모아서 볼 수 없을까?



02. MLflow

Tracking 도구

mlflow



Weights & Biases



neptune.ai

02. MLflow

MLflow Tracking

mlflow

ExperimentsModels

Experiments

+

←

Search Experiments

Default

titanic

titanic

Track machine learning training runs in an experiment. [Learn more](#)

Experiment ID: 1

Notes

Showing 7 matching runs

Refresh

Compare

Delete

Download CSV

Start Time

Columns

Only show differences

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

Search

Filter

Clear

								Metrics	Parameters >		
	Start Time	Duration	Run Name	User	Source	Version	Models	f1 score	class	class num	env
	15 minutes ago	1.6s	-	root	mlflow_trac	-	sklearn	0.772	Counter([0: ...	2	local
	16 minutes ago	1.7s	-	root	mlflow_trac	-	sklearn	0.811	Counter([0: ...	2	local
	25 minutes ago	1.6s	-	root	test3.py	-	sklearn	0.821	Counter([0: ...	2	local
	26 minutes ago	38ms	-	root	test3.py	-	-	-	-	-	local
	27 minutes ago	20ms	-	root	test3.py	-	-	-	-	-	local
	27 minutes ago	20ms	-	root	test3.py	-	-	-	-	-	local
	30 minutes ago	17ms	-	root	test3.py	-	-	-	-	-	local

Load more

02. MLflow

MLflow Tracking

- log_param : 파라미터 저장
- log_metric : 평가 저장 (파라미터와 다르게 시계열하게 기록 저장 가능)
- log_artifact : 이미지, csv, 문서 등 모델과 관련된 파일 저장 가능
- log_model : 학습 모델 저장 가능

02. MLflow

MLflow Tracking 실습

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

```
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_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.log_metric("f1 score", score)
        mlflow.log_artifact(train_dir)
        mlflow.sklearn.log_model(model, "titanic_model")
```

02. MLflow

MLflow Tracking 실습

- 영화 리뷰 긍부정 분류하는 DL 모델 트랙킹 실습

/MLflow_tutorial/ML/mlflow_tracking.py

```
mlflow.set_experiment('nlp')
with mlflow.start_run() as run:
    # Hyper Parameters
    train_dir = "train.csv"
    epochs = 3
    max_len = 30
    hidden_dim = 300
    lr = 0.001
    batch_size = 4
    total_acc = None
    device = torch.device("cpu")

    # Flow
    print("1. Load Data")
    train_x, train_y, encoder = load_data(train_dir)
    train_x, val_x, train_y, val_y = train_test_split(train_x, train_y, test_size=0.1)

    print("2. Pre Processing")
    train_x = [sentence.split(" ") for sentence in train_x]
    val_x = [sentence.split(" ") for sentence in val_x]
```


02. MLflow

MLflow Models

- Flask 이용해서 매번 코드 작성해서 API 서버 만들기 너무 귀찮은데
간단하게 명령어 하나로 API 서버 만들 수 없을까?
- Artifacts에 저장한 모델을 사용하고 싶은데 어떻게하지?



02. MLflow

MLflow Inference 실습

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

```
import mlflow
import pandas as pd

if __name__ == '__main__':
    logged_model = 'runs:/72558aa709ab4d898c54c8b6f2a2d2ff/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))
```

02. MLflow

MLflow Inference 실습

- MLflow에 저장된 리뷰 감정 분류 DL 모델 가져와서 추론하기

/MLflow_tutorial/DL/mlflow_inference.py

```
import mlflow
import pickle
import numpy as np

if __name__ == "__main__":
    max_len = 30
    test_x = "좋은 영화 입니다"

    with open('vocab.pickle', 'rb') as f:
        vocab = pickle.load(f)

    test_x = test_x.split(" ")
    text_pipeline = lambda x: vocab(x)
    test_x = text_pipeline(test_x)

    if len(test_x) > max_len:
        test_x = test_x[0:max_len]
    else:
        test_x = test_x[0:] + ([0]*(max_len-len(test_x)))

    print(test_x)
    test_x = np.array([test_x])
    loaded_model = mlflow.pyfunc.load_model('runs:/eb143e5b147e464d8d94b75178307609/model')
    print(loaded_model.predict(test_x))
```

02. MLflow

MLflow API 서버 배포

- 명령어 : `mlflow models serve -m runs:/72558aa709ab4d898c54c8b6f2a2d2ff/titanic_model --no-conda`
- 요청 : `curl http://<IP주소>:<PORT번호>/invocations -H 'Content-Type: application/json' -d '{"columns": ["Pclass", "Sex", "Fare", "SibSp", "Parch"], "data": [[1, 2, 3, 2, 2], [1, 2, 4, 5, 6]]}'`

02. MLflow

MLflow Registry 필요성

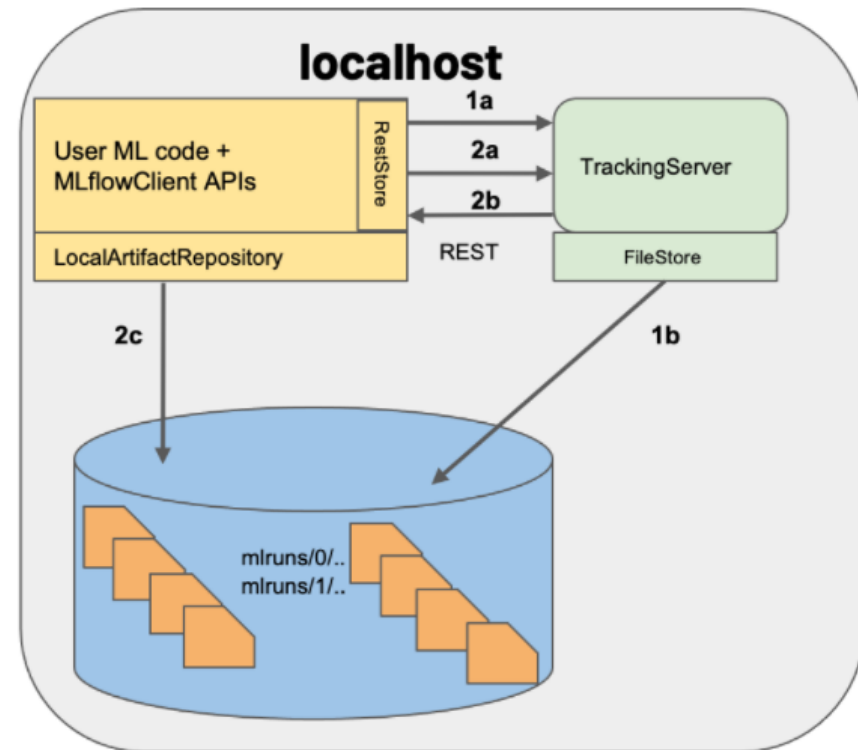
- 회사에 다른 3명의 모델러와 같은 프로젝트에 대한 실험 결과를 공유하고 싶은데.. PC가 서로 다르네..
- 모델 기록을 중앙에서 관리하고 싶어!



02. MLflow

MLflow Registry

- MLflow ui로 미리 tracking server를 띄운 후, 모델 학습 코드를 실행시켜 기록을 전송하는 방식이다.
- 웹서버를 띄우고 요청 보내는 형태라 생각하면 편하다.



02. MLflow

MLflow Registry 실습

- 명령어 : 트래킹 서버를 띄우는 과정
 - `cd /<경로>/MLflow_tutorial/tracking_server`
 - `mlflow server --backend-store-uri file:/<경로>/MLflow_tutorial/tracking_server --default-artifact-root /<경로>/MLflow_tutorial/tracking_server --host 0.0.0.0 --port 2022`

02. MLflow

MLflow Registry 실습

- 명령어 : `mlflow.set_tracking_uri("http://<IP주소>:<PORT번호>")`
- `mlflow_tracking.py` 부분에서 주석처리 된 부분을 풀어주고 실험 진행

```
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_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.log_metric("f1 score", score)
            mlflow.log_artifact(train_dir)
            mlflow.sklearn.log_model(model, "titanic_model")
```

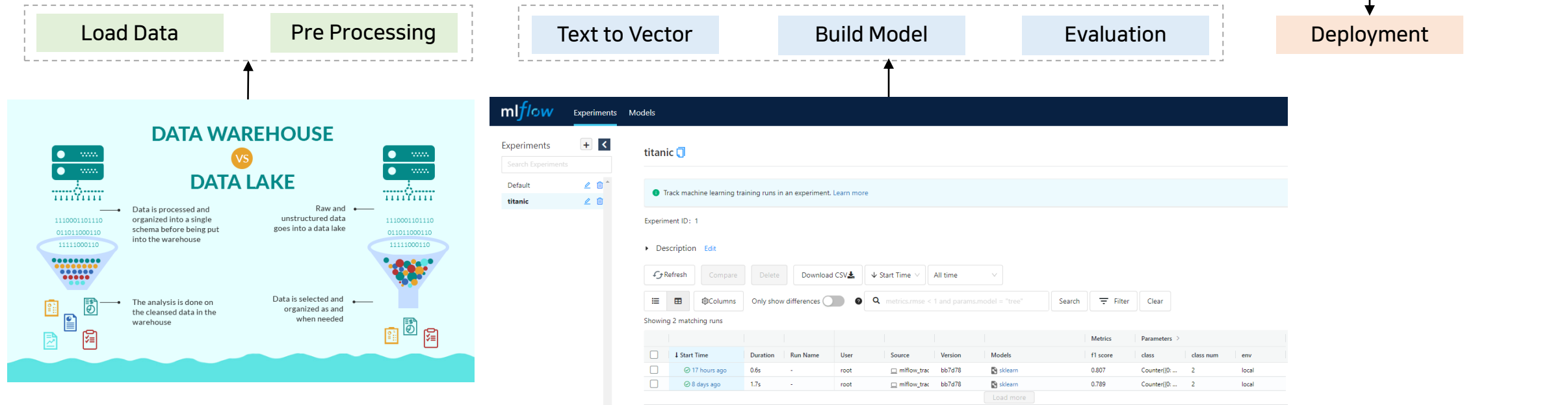
03

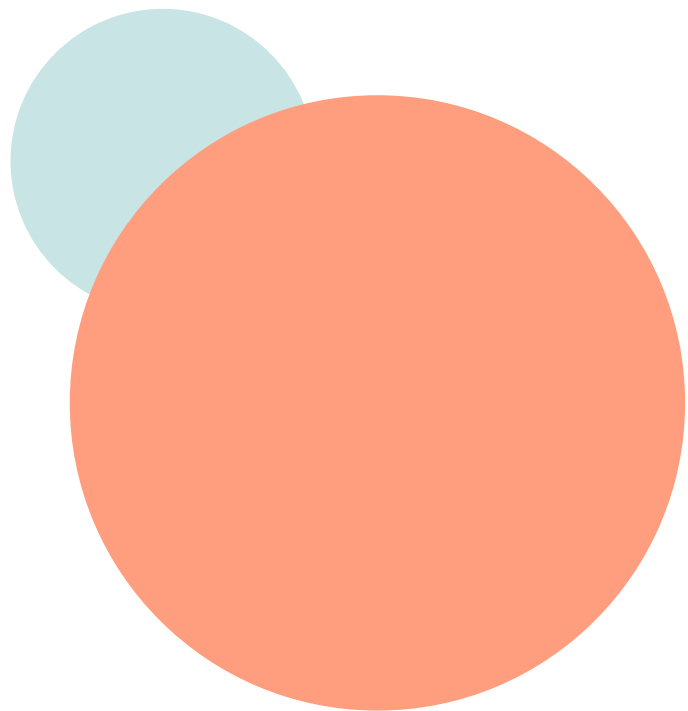
ML 파이프라인

파이프라인 구성 예시

03. ML 파이프라인

전체 과정





일시 2022년 03월 06일
장소 Google Meet
발표 이정훈

THANK
YOU