

MLFlow

Грозин Владислав

Что мы хотим от экспериментальной платформы?

Что мы хотим

Зачем?

Что мы хотим от экспериментальной платформы?

Что мы хотим	Зачем?
<ul style="list-style-type: none">- Сохранение параметров- Сохранение метрик	

Что мы хотим от экспериментальной платформы?

Что мы хотим	Зачем?
<ul style="list-style-type: none">- Сохранение параметров- Сохранение метрик	Сравнение экспериментов

Что мы хотим от экспериментальной платформы?

Что мы хотим	Зачем?
<ul style="list-style-type: none">- Сохранение параметров- Сохранение метрик	Сравнение экспериментов
<ul style="list-style-type: none">- Reproducibility- Сохранение графиков, картинок- Сохранение метаданных о том, как артефакты генерируются	

Что мы хотим от экспериментальной платформы?

Что мы хотим	Зачем?
<ul style="list-style-type: none">- Сохранение параметров- Сохранение метрик	Сравнение экспериментов
<ul style="list-style-type: none">- Reproducibility- Сохранение графиков, картинок- Сохранение метаданных о том, как артефакты генерируются	Дебаг

Что мы хотим от экспериментальной платформы?

Что мы хотим	Зачем?
<ul style="list-style-type: none">- Сохранение параметров- Сохранение метрик	Сравнение экспериментов
<ul style="list-style-type: none">- Reproducibility- Сохранение графиков, картинок- Сохранение метаданных о том, как артефакты генерируются	Дебаг
<ul style="list-style-type: none">- Сохранение файлов моделей	

Что мы хотим от экспериментальной платформы?

Что мы хотим	Зачем?
<ul style="list-style-type: none">- Сохранение параметров- Сохранение метрик	Сравнение экспериментов
<ul style="list-style-type: none">- Reproducibility- Сохранение графиков, картинок- Сохранение метаданных о том, как артефакты генерируются	Дебаг
<ul style="list-style-type: none">- Сохранение файлов моделей	Inference / production

MLFlow

- Экспериментальный фреймворк для ML

Что мы хотим	Компонента MLFlow
<ul style="list-style-type: none">- Сохранение параметров- Сохранение метрик	MLFlow Tracking – система логирования экспериментов
<ul style="list-style-type: none">- Reproducibility- Сохранение графиков, картинок- Сохранение метаданных о том, как артефакты генерируются	MLFlow Projects – система ведения экспериментов
<ul style="list-style-type: none">- Сохранение файлов моделей	MLFlow Models – система запуска в прод

1. MLFlow Tracking

Логирование экспериментов

Один запуск скрипта = один прогон эксперимента (run).

Эксперимент = «домик» для запусков (run)

```
import mlflow
import argparse

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("epochs", default=10)
    args = parser.parse_args()

    mlflow.log_param("epochs", args.epochs)

    # ...
    acc = tran_and_evaluate(args.epochs)

    mlflow.log_metric("accuracy", acc)
```

Логирование экспериментов'

Можно логировать в БД или удалённый сервер:

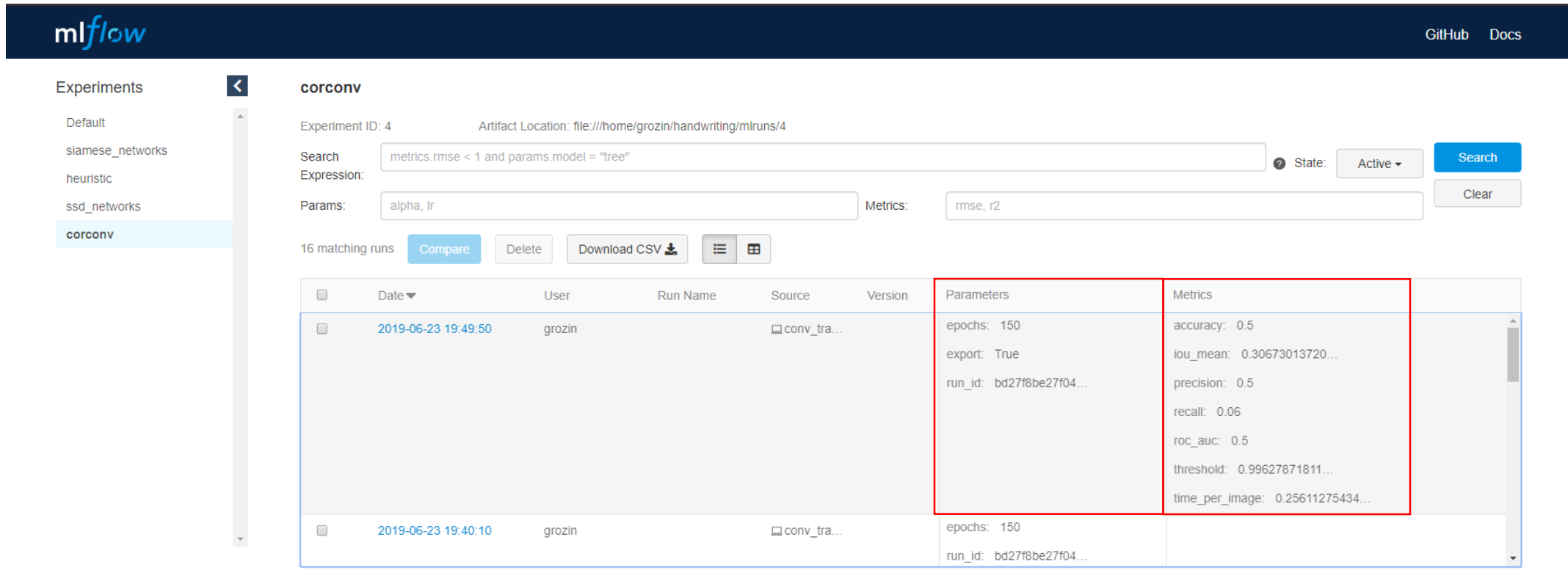
```
mlflow.tracking.set_tracking_uri(url)
```

По умолчанию метрики и параметры пишутся в локальную файловую систему

```
./
  script.py
  mlflow/
    0/
      meta.yaml
      01285c63e3b49d7afec5ed3dd55030f/
        meta.yaml
        metrics/
          accuracy
        parameters/
          epochs
```

Сравнение экспериментов

Сервер и дашборда (можно запускать локально): `> mlflow server`



The screenshot displays the mlflow web interface. On the left, a sidebar lists experiments: Default, siamese_networks, heuristic, ssd_networks, and **corconv**. The main area shows the 'corconv' experiment details. At the top, it indicates 'Experiment ID: 4' and 'Artifact Location: file:///home/grozin/handwriting/mlruns/4'. Below this, there are search filters: 'Search Expression: metrics.rmse < 1 and params.model = "tree"', 'Params: alpha, lr', and 'Metrics: rmse, r2'. A 'State: Active' dropdown and a 'Search' button are also present. Below the filters, it says '16 matching runs' and provides buttons for 'Compare', 'Delete', and 'Download CSV'. A table of runs is displayed below, with the first two rows highlighted. The table has columns for Date, User, Run Name, Source, Version, Parameters, and Metrics. The first row shows a run from 2019-06-23 19:49:50 by user 'grozin' with parameters 'epochs: 150', 'export: True', and 'run_id: bd27f8be27f04...'. The second row shows a run from 2019-06-23 19:40:10 by user 'grozin' with parameters 'epochs: 150' and 'run_id: bd27f8be27f04...'. The Metrics column for the first row lists: accuracy: 0.5, iou_mean: 0.30673013720..., precision: 0.5, recall: 0.06, roc_auc: 0.5, threshold: 0.99627871811..., and time_per_image: 0.25611275434....

Date	User	Run Name	Source	Version	Parameters	Metrics
2019-06-23 19:49:50	grozin	conv_tra...	conv_tra...		epochs: 150 export: True run_id: bd27f8be27f04...	accuracy: 0.5 iou_mean: 0.30673013720... precision: 0.5 recall: 0.06 roc_auc: 0.5 threshold: 0.99627871811... time_per_image: 0.25611275434...
2019-06-23 19:40:10	grozin	conv_tra...	conv_tra...		epochs: 150 run_id: bd27f8be27f04...	

2. MLFlow Projects

Описание проектов в MLFlow

- Можно документировать окружение и запускаемые команды (entrypoint)
- Документируется в файлике MLProject:

```
name: My Little Project
```

```
docker_env:  
  image: mlflow-docker-example
```

```
entry_points:  
  main:  
    parameters:  
      epochs: {type: int, default: 30}  
    command: "python train.py --epochs {epochs}"
```

```
conda_env:  
  my_env.yaml
```

Запуск проекта

```
mlflow run
```

Можно указать параметры:

```
mlflow run -P epochs=50
```

Можно запустить конкретный шаг (entrypoint)

```
mlflow run -e main
```

Можно запускать на Kubernetes!

```
mlflow run --backend kubernetes --backend-config  
config.json
```


Is it good?

```
entry_points:
  main:
    parameters:
      epochs: {type: int, default: 30}
    command: "python train.py --epochs {epochs}"
```

```
import mlflow
import argparse

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("epochs", default=10)
    args = parser.parse_args()

    mlflow.log_param("epochs", args.epochs)
```

Is it good?

```
entry_points:
  main:
    parameters:
      epochs: {type: int, default: 30}
    command: "python train.py --epochs {epochs}"
```

```
import mlflow
import argparse

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("epochs", default=10)
    args = parser.parse_args()

    mlflow.log_param("epochs", args.epochs)
```

3. MLFlow Models

Сохранение моделей

Модели можно сохранять как артефакты эксперимента.

Поддерживаются форматы (flavor):

- Python Function (python_function)
- Keras (keras)
- PyTorch (pytorch)
- Scikit-learn (sklearn)
- TensorFlow (tensorflow)

Сохранение моделей

Для каждого формата поддерживаются операции:

- Save
- Load

Модель:

- Изолирована
- Поддерживает `predict(inputs: pandas.DataFrame)`

Model Serving

Благодаря изолированности можно запускать модели напрямую:

```
mlflow models serve -m <URL файла модели>
```

REST API:

```
curl http://127.0.0.1:5000/invocations -H 'Content-Type:
application/json' -d '{
    "columns": ["a", "b", "c"],
    "data": [[1, 2, 3], [4, 5, 6]]
}'
```


Model Serving

Благодаря изолированности можно запускать модели напрямую:

```
mlflow models serve -m <URL файла модели>
```

REST API:

```
curl http://127.0.0.1:5000/invocations -H 'Content-Type:
application/json' -d '{
    "columns": ["a", "b", "c"],
    "data": [[1, 2, 3], [4, 5, 6]]
}'
```



Из-за поддержки только DataFrame тяжело
работать с не-табулярными данными

Сравнение

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?		

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?	Один запуск скрипта	Коммит в git
Обеспечение воспроизводимости		

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?	Один запуск скрипта	Коммит в git
Обеспечение воспроизводимости	Не нужно; сохраняй изолированные бинарники	Сохраняем все входящие данные и весь код в git-подобной структуре
Многоэтапная обработка / стекинг моделей		

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?	Один запуск скрипта	Коммит в git
Обеспечение воспроизводимости	Не нужно; сохраняй изолированные бинарники	Сохраняем все входящие данные и весь код в git-подобной структуре
Многоэтапная обработка / стекинг моделей	Боль из-за изолированности бинарников	Отлично описываются в виде графа
Вывод метрик и сравнение экспериментов		

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?	Один запуск скрипта	Коммит в git
Обеспечение воспроизводимости	Не нужно; сохраняй изолированные бинарники	Сохраняем все входящие данные и весь код в git-подобной структуре
Многоэтапная обработка / стекинг моделей	Боль из-за изолированности бинарников	Отлично описываются в виде графа
Вывод метрик и сравнение экспериментов	Классная браузерная дашборда	☹
Хранение данных		

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?	Один запуск скрипта	Коммит в git
Обеспечение воспроизводимости	Не нужно; сохраняй изолированные бинарники	Сохраняем все входящие данные и весь код в git-подобной структуре
Многоэтапная обработка / стекинг моделей	Боль из-за изолированности бинарников	Отлично описываются в виде графа
Вывод метрик и сравнение экспериментов	Классная браузерная дашборда	☹
Хранение данных	Твоя проблема	Хранятся внутри репо
Запуск в продакшн		

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?	Один запуск скрипта	Коммит в git
Обеспечение воспроизводимости	Не нужно; сохраняй изолированные бинарники	Сохраняем все входящие данные и весь код в git-подобной структуре
Многоэтапная обработка / стекинг моделей	Боль из-за изолированности бинарников	Отлично описываются в виде графа
Вывод метрик и сравнение экспериментов	Классная браузерная дашборда	☹
Хранение данных	Твоя проблема	Хранятся внутри репо
Запуск в продакшн	Микросервисы + можно загрузить изолированные бинарники моделей в любой код	???

MLFlow vs DVC

	MLFlow	DVC
Эксперимент?	Один запуск скрипта	Коммит в git
Обеспечение воспроизводимости	Не нужно; сохраняй изолированные бинарники	Сохраняем все входящие данные и весь код в git-подобной структуре
Многоэтапная обработка / стекинг моделей	Боль из-за изолированности бинарников	Отлично описываются в виде графа
Вывод метрик и сравнение экспериментов	Классная браузерная дашборда	☹
Хранение данных	Твоя проблема	Хранятся внутри репо
Запуск в продакшн	Микросервисы + можно загрузить изолированные бинарники моделей в любой код	???
	Простой пайплайн, фиксированные данные и одна сложная модель, много экспериментов и метрик, от проекта достаточно одного объекта с predict	Длинный и сложный пайплайн обработки данных, дополняемые данные, стекинг моделей, важно иметь возможность восстановить старые версии