How to Manage Your ML Models?

Hannes Hansen

8.11.2024

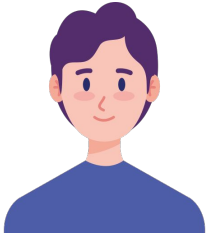
Goal

Lifecycle of a ML model

Challenges you might face

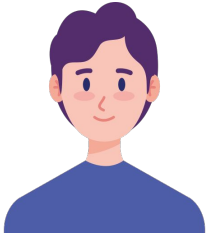
How a Model Registry can help

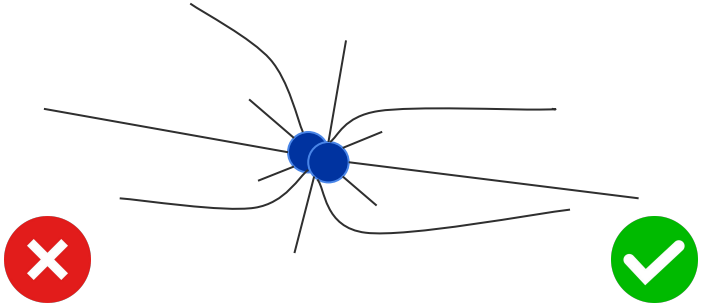
A brief ML Journey



3

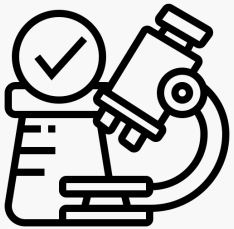
A brief ML Journey

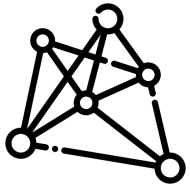




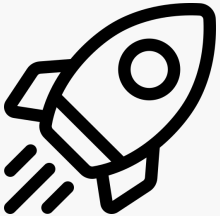
A brief ML Journey

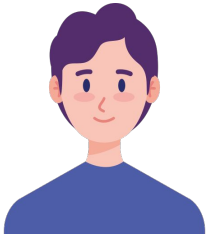
Prototyping

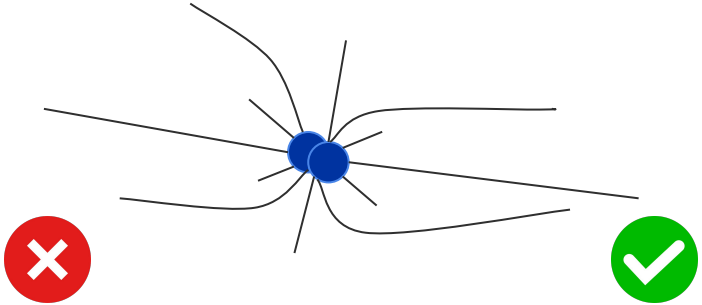




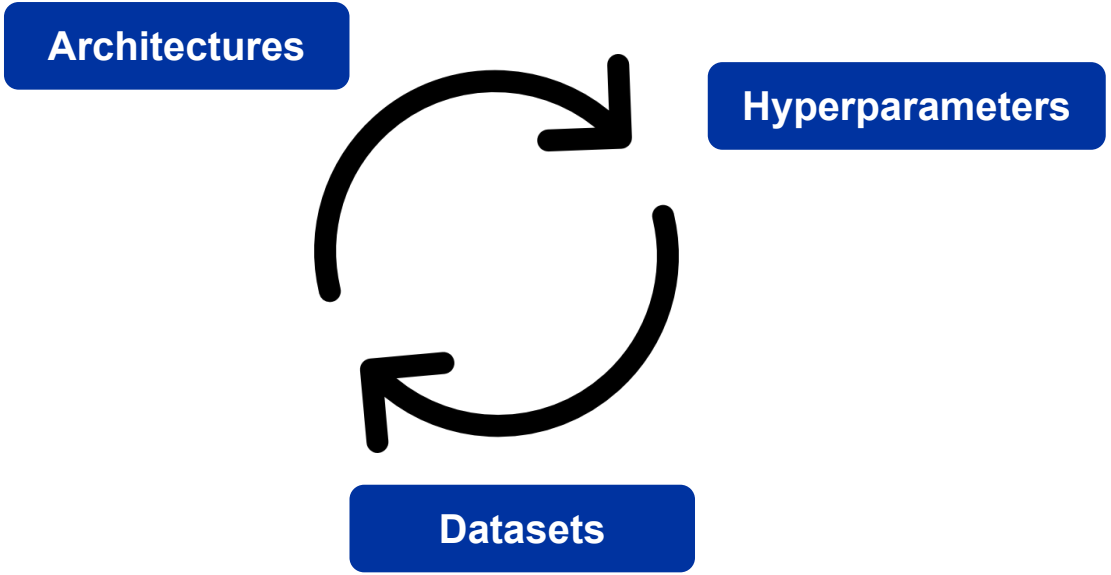
Production







Prototyping



Prototyping

```
n_estimators = 10
model = RandomForestClassifier(n_estimators=n_estimators).fit(train_data_v1)
score = accuracy(truth, model.predict(test_data))
model.save("model_10_trees.pkl")
```

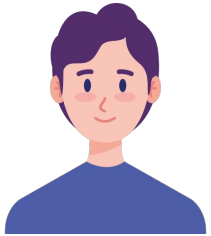
Prototyping

```
n_estimators = 10
model = RandomForestClassifier(n_estimators=n_estimators).fit(train_data_v1)
score = accuracy(truth, model.predict(test_data))
model.save("model_10_trees.pkl")
```

```
n_estimators = 20
...
model.save("model_20_trees.pkl")
```

Model Registry

- Central storage for your models and metadata
- Access via API
- https://mlflow.org/



- Models
- Plots
- Configuration

Model Registry

Prototyping

```
mlflow.log_metric("accuracy", 0.8)
```

Prototyping

```python
mlflow.log_metric("accuracy", 0.8)
```

```python
mlflow.log_param("n_estimators", 10)
```

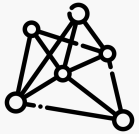
Prototyping

```
mlflow.log_metric("accuracy", 0.8)
```

```
mlflow.log_param("n_estimators", 10)
```

```
mlflow.sklearn.log_model(
  model,
  artifact_path="model",
  registered_model_name="model.dev"
)
```

Unique Model Identifier



model.dev

Overview | Model metrics | System metrics | Artifacts

Description

No description

Details

Created at	2024-11-01 17:53:40
Created by	hannes
Experiment ID	0
Status	Running
Run ID	0e906ec3c8ad4a1a8c490391d89ed832
Duration	
Datasets used	—
Tags	Add
Source	lineage.py 92063d1
Logged models	—
Registered models	model.dev v1

Parameters (1)

Search parameters

Parameter	Value
n_estimators	10

Metrics (1)

Search metrics

Metric	Value
accuracy	0.8

Overview | Model metrics | System metrics | Artifacts

Description

No description

Details

Created at	2024-11-01 17:53:40
Created by	hannes
Experiment ID	0
Status	Running
Run ID	0e906ec3c8ad4a1a8c490391d89ed832
Duration	
Datasets used	—
Tags	Add
Source	lineage.py 92063d1
Logged models	—
Registered models	model.dev v1

Parameters (1)

Search parameters

Parameter	Value
n_estimators	10

Metrics (1)

Search metrics

Metric	Value
accuracy	0.8

Overview | Model metrics | System metrics | Artifacts

Description

No description

Details

Created at	2024-11-01 17:53:40
Created by	hannes
Experiment ID	0
Status	Running
Run ID	0e906ec3c8ad4a1a8c490391d89ed832
Duration	
Datasets used	—
Tags	Add
Source	lineage.py 92063d1
Logged models	—
Registered models	model.dev v1

Parameters (1)

Search parameters

Parameter	Value
n_estimators	10

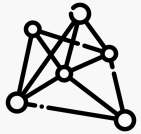
Metrics (1)

Search metrics

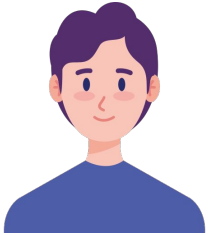
Metric	Value
accuracy	0.8

Going to Production

Prototyp

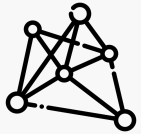


model.dev



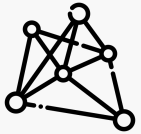
Going to Production

Prototyp



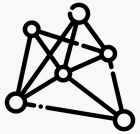
model.dev

Staging

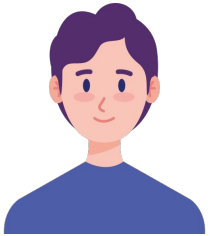


model.staging

Production



model.prod



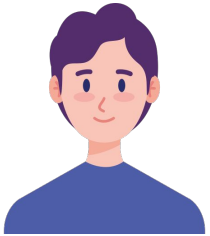
Going to Production

```
client.copy_model_version("models:/model.dev/1", "model.staging")
client.copy_model_version("models:/model.staging/1", "model.prod")
```

Name	Latest version
model.dev	Version 1
model.prod	Version 1
model.staging	Version 1

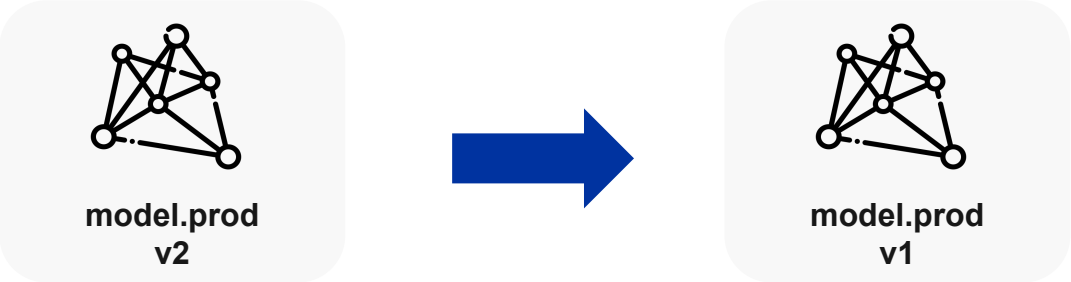
Updates

```
model = RandomForestClassifier(n_estimators=30).fit(train_data_v2)
...
mlflow.sklearn.log_model(
  model,
  artifact_path="model",
  registered_model_name="model.dev"
)
```



Name	Latest version
model.dev	Version 2
model.prod	Version 2
model.staging	Version 2

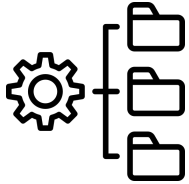
Rollback



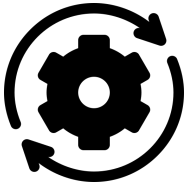
Model Registry

Model Storage

Model Versioning



Model Lineage



Stage Transitioning

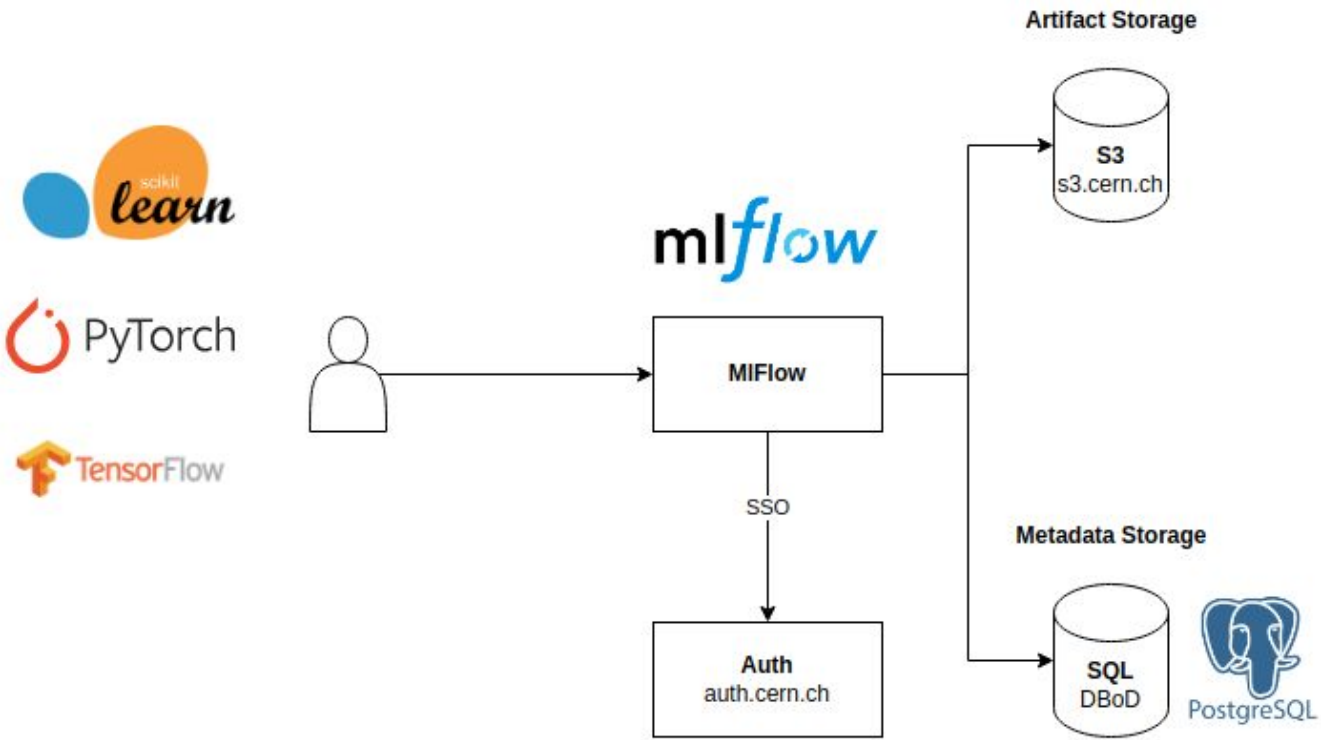
CERN

home.cern

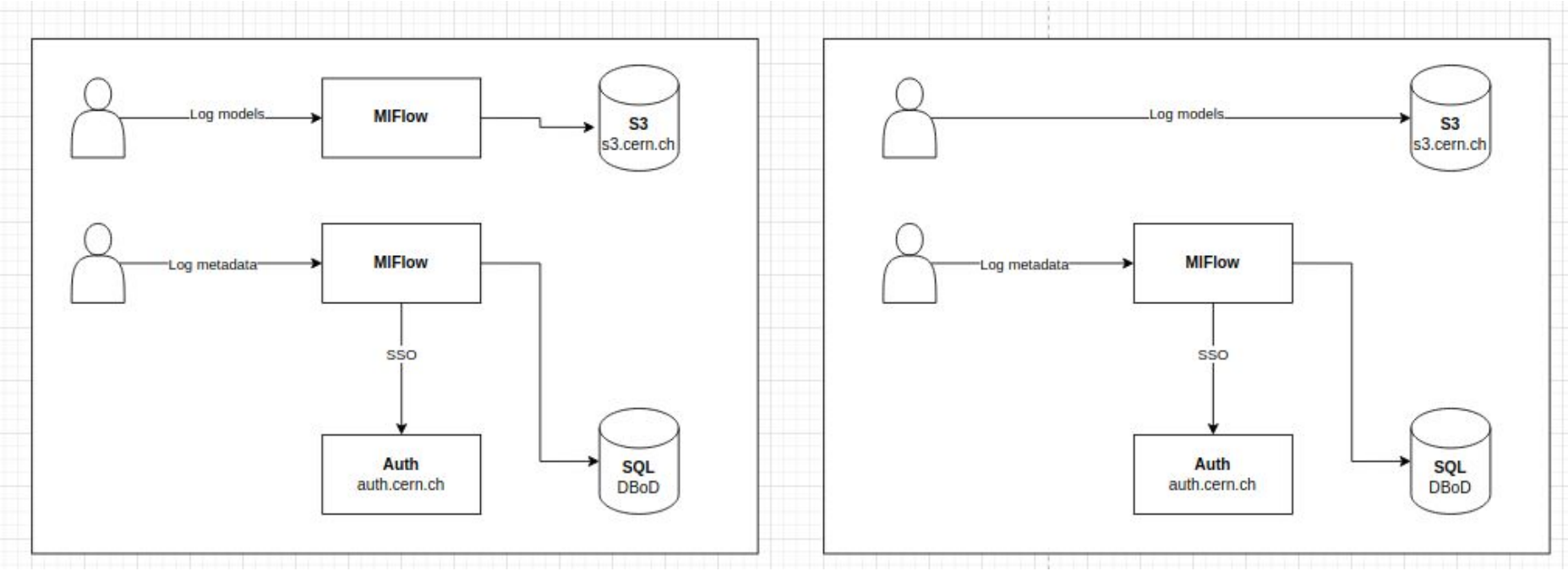
Custom Models

```python
import mlflow
import torch

class CustomModel(mlflow.pyfunc.PythonModel):
    def __init__(self):
        self.model = torch.Linear()

    def predict(self, context, model_input, params=None):
        prediction = self.model(torch.tensor(model_input))
        return prediction

with mlflow.start_run():
    mlflow.pyfunc.log_model(
        "custom_model",
        python_model=CustomModel(),
        pip_requirements=["torch==2.0.0"]
    )
```

Architecture



Architecture



Packaging

- **Model is the output of a training process**

- **Needs to be serialized for storage**
  - Weight matrix of a linear layer in a **weights.npy**
  - Pickle file of an sklearn model or PyTorch mode **model.pkl**
  - ONNX **model.onnx**

- **Depending on the format, you might need more information**
  - Dependencies e.g. scikit-learn==1.5.2

```python
import mlflow

model = RandomForestClassifier()
mlflow.sklearn.log_model(
  model,
  artifact_path="model",
  registered_model_name="model.dev",
  pip_requirements=["scikit-learn=1.5.2"]
)
```

- model
  - MLmodel
  - conda.yaml
  - model.pkl
  - python_env.yaml
  - requirements.txt

model/python_env.yaml 120B

Path: mlflow-artifacts:/0/058580faa54c43eea8155b9b8b283c69/artifacts/model/python_env.yaml

```yaml
python: 3.12.4
build_dependencies:
- pip==24.2
- setuptools==72.1.0
- wheel==0.43.0
dependencies:
- -r requirements.txt
```

- model
  - MLmodel
  - conda.yaml
  - model.pkl
  - python_env.yaml
  - requirements.txt

model/requirements.txt 94B

Path: mlflow-artifacts:/0/058580faa54c43eea8155b9b8b283c69/artifacts/model/requirements.txt

```
mlflow==2.16.2
cloudpickle==3.0.0
numpy==2.1.1
pandas==2.2.3
scikit-learn==1.5.2
scipy==1.14.1
```