

1. Se connecter sur ma machine Ubuntu (Linux)

```
netamine@Ubuntu2: ~/session4/tp3
netamine@Ubuntu2:~/session4/tp3$ tree
.
├── backend
│   ├── Dockerfile
│   └── main.py
├── docker-compose.yaml
├── frontend
│   ├── Dockerfile
│   └── main.py
├── mlflow
│   ├── Dockerfile
│   ├── metrics.py
│   ├── mlruns
│   └── mlflow-experiments
└── models
    └── model.pkl

6 directories, 8 files
netamine@Ubuntu2:~/session4/tp3$
```

2. J'ai fait un nettoyage de toutes mes images et Dockers pour libérer de l'espace

```
netamine@Ubuntu2: ~/session4/tp3
netamine@Ubuntu2:~/session4/tp3$ docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
netamine@Ubuntu2:~/session4/tp3$ docker images
REPOSITORY    TAG       IMAGE ID   CREATED   SIZE
netamine@Ubuntu2:~/session4/tp3$
```

3. Lancement du build et démarrage des dockers avec docker-compose

```
netamine@Ubuntu2: ~/session4/tp3
netamine@Ubuntu2:~/session4/tp3$ docker-compose up --build
```

```

netamine@Ubuntu2: ~/session4/tp3
netamine@Ubuntu2:~/session4/tp3$ docker-compose up --build
WARN[0000] /home/netamine/session4/tp3/docker-compose.yml: 'version' is obsolete
[+] Building 556.0s (32/32) FINISHED
=> [frontend internal] load build definition from Dockerfile
=> => transferring dockerfile: 441B
=> [frontend internal] load metadata for docker.io/library/ubuntu:jammy
=> [frontend auth] library/ubuntu/pull token for registry-1.docker.io
=> [frontend internal] load .dockerignore
=> => transferring context: 2B
=> [frontend 1/7] FROM docker.io/library/ubuntu:jammy@sha256:19473ce7fcc2ffbec9bdf39fca715a8d1ef7de81cb9ea570990dc3652aac1ac
=> => resolve docker.io/library/ubuntu:jammy@sha256:19473ce7fcc2ffbec9bdf39fca715a8d1ef7de81cb9ea570990dc3652aac1ac
=> => sha256:19473ce7fcc2ffbec9bdf39fca715a8d1ef7de81cb9ea570990dc3652aac1ac 1.13kB / 1.13kB
=> => sha256:94cd4b744518d1b0c0ff5b13112f164a9f9ef9e01a4f9e71aee7ee64337c40 424B / 424B
=> => sha256:67cd4b744518d1b0c0ff5b13112f164a9f9ef9e01a4f9e71aee7ee64337c40 2.30kB / 2.30kB
=> => sha256:7496c8da3249ee14b15479ce130b32a39a301c62a2451f193509aeb8db 29.5MB / 29.5MB
=> => extracting sha256:7496c8da3249ee14b15479ce130b32a39a301c62a2451f193509aeb8db
=> [frontend internal] load build context
=> => transferring context: 1.36kB
=> [frontend 2/7] RUN apt-get update && apt-get install -y bash python3 python3-pip git gcc && pip install mlflow suikit-learn==1.2.2
=> [frontend 3/7] RUN mkdir -p /mlflow
=> [frontend 4/7] WORKDIR /mlflow
=> [frontend 5/7] COPY main.py .
=> [frontend] exporting to image
=> => exporting layers
=> => writing image sha256:b5210eb3f4073552ed8ca08a1cf72474f3e4130f5364563e8d114f1f2ad1
=> => naming to docker.io/library/tp3-mlflow
=> [backend internal] load build definition from Dockerfile
=> => transferring dockerfile: 922B
=> [backend auth] library/ubuntu/pull token for registry-1.docker.io
=> [backend internal] load .dockerignore
=> => transferring context: 2B
=> [backend internal] load build context
=> => transferring context: 1.34kB
=> CACHED [frontend 2/7] WORKDIR /app
=> [backend 3/8] RUN mkdir -p /app
=> [backend 4/8] RUN mkdir -p /models
=> [backend 5/8] RUN apt-get update
=> [backend 6/8] RUN apt-get install -y bash python3 python3-pip
=> [backend 7/8] RUN pip install --no-cache-dir fastapi uvicorn joblib numpy pydantic suikit-learn==1.2.2 pandas

```

```

netamine@Ubuntu2: ~/session4/tp3
=> => transferring dockerfile: 922B
=> [backend internal] load .dockerignore
=> => transferring context: 2B
=> [backend internal] load build context
=> => transferring context: 1.54kB
=> CACHED [frontend 2/7] WORKDIR /app
=> CACHED [backend 3/8] RUN mkdir -p /app
=> CACHED [backend 4/8] RUN mkdir -p /models
=> CACHED [backend 5/8] RUN apt-get update
=> CACHED [backend 6/8] RUN apt-get install -y bash python3 python3-pip
=> CACHED [backend 7/8] RUN pip install --no-cache-dir fastapi uvicorn joblib numpy pydantic suikit-learn==1.2.2 pandas
=> CACHED [backend 8/8] COPY main.py /app
=> [backend] exporting to image
=> => exporting layers
=> => writing image sha256:03cf70cbbbeadea77c8789673f3ef01801ccaa0c8642709596324a87dae8499c1
=> => naming to docker.io/library/tp3-backend
=> [frontend internal] load build definition from Dockerfile
=> => transferring dockerfile: 976B
=> [frontend internal] load .dockerignore
=> => transferring context: 2B
=> [frontend internal] load build context
=> => transferring context: 1.66kB
=> CACHED [frontend 3/7] RUN apt-get update
=> CACHED [frontend 4/7] RUN apt-get install -y bash python3 python3-pip
=> CACHED [frontend 5/7] RUN pip install pandas seaborn streamlit matplotlib
=> CACHED [frontend 6/7] RUN mkdir -p /app
=> CACHED [frontend 7/7] COPY main.py /app/main.py
=> [frontend] exporting to image
=> => exporting layers
=> => writing image sha256:26be193a90e14d697e2c86501ac5848f1d5c87f307be7957021c5d26f307aeec
=> => naming to docker.io/local/frontend
[+] Running 3/0
  Container mlflow      Created
  Container backend     Created
  Container frontend    Created
Attaching to backend, frontend, mlflow
frontend |
frontend | Collecting usage statistics. To deactivate, set browser.gatherUsageStats to false.
frontend |
frontend |
frontend | You can now view your Streamlit app in your browser.
frontend |
frontend | URL: http://0.0.0.0:8001
mlflow | [2024-06-15 03:16:15 +0000] [9] [INFO] Starting gunicorn 22.0.0
mlflow | [2024-06-15 03:16:15 +0000] [9] [INFO] Listening at: http://0.0.0.0:8003 (9)
mlflow | [2024-06-15 03:16:15 +0000] [9] [INFO] Using worker: sync
mlflow | [2024-06-15 03:16:15 +0000] [10] [INFO] Booting worker with pid: 10
mlflow | [2024-06-15 03:16:15 +0000] [11] [INFO] Booting worker with pid: 11
mlflow | [2024-06-15 03:16:15 +0000] [12] [INFO] Booting worker with pid: 12
mlflow | [2024-06-15 03:16:15 +0000] [13] [INFO] Booting worker with pid: 13
backend | INFO: Started server process [1]
backend | INFO: Waiting for application startup.
backend | INFO: Application startup complete.
backend | INFO: Uvicorn running on http://0.0.0.0:8002 (Press CTRL+C to quit)

```

4. Vérification du FrontEnd

The screenshot shows a web browser with three tabs: 'main - Streamlit', 'FastAPI - Swagger UI', and 'MLflow'. The address bar shows '192.168.2.77:8001'. The main content area displays the 'Iris Species Prediction' interface. It features four input fields for 'Sepal Length' (0,30), 'Sepal Width' (0,10), 'Petal Length' (0,10), and 'Petal Width' (0,40). Each field has a minus and a plus button for adjustment. Below these fields is a red 'Predict' button. At the bottom, a green box displays the prediction result: 'The predicted species is: setosa'.

5. Vérification du Backend

The screenshot shows the FastAPI Swagger UI in a web browser. The address bar shows '192.168.2.77:8002/docs'. The interface includes the FastAPI logo with version '0.1.0' and 'OAS 3.1'. Under the 'default' section, there are two endpoints: 'GET / Read Root' and 'POST /predict Predict'. Below these, the 'Schemas' section lists three error types: 'HTTPValidationError', 'IrisRequest', and 'ValidationError', each with an 'Expand all' link and an 'object' type.

6. Calcul des metrics et enregistrement des expériences MIFlow

```
root@7649d119f753: /mlflow
netamine@Ubuntu2:~/session4/tp3$ docker exec -it mlflow bash
root@7649d119f753: /mlflow#
```

```
root@7649d119f753: /mlflow
netamine@Ubuntu2:~/session4/tp3$ docker exec -it mlflow bash
root@7649d119f753: /mlflow# python3 metrics.py
Modèle enregistré sous: 38e9dc02d92f4e97b0df44442aaf9ad4
Modèle sauvegardé sous: /models/model_2024_06_15_T03_23_08.pkl
root@7649d119f753: /mlflow#
```

Réseau > 192.168.2.77 > session4 > tp3 > models					Rechercher dans : models	
	Nom	Modifié le	Type	Taille		
es	model_2024_06_15_T03_23_08.pkl	2024-06-14 23:23	Fichier PKL			
is	model_2024_06_15_T03_14_31.pkl	2024-06-14 23:14	Fichier PKL			
dio 2017	model.pkl	2024-06-14 21:49	Fichier PKL			
dio 2022						

The screenshot displays the MLflow web interface for managing experiments. The browser tabs show 'main - Streamlit', 'FastAPI - Swagger UI', and 'MLflow'. The URL is '192.168.2.77:8003/#/experiments/904687765460583689?searchFilter=&orderByKey=attributes.start_time&orderByAsc=false&startTime=ALL&lifecycleFilter=Active&modelVersionF...'. The MLflow logo and version '2.13.2' are in the top left. The 'Experiments' tab is active, with 'Models' also visible. On the left, a sidebar shows 'Experiments' with a search bar and a list containing 'Default' and 'Experience_2024_06_14' (selected). The main area shows the details for 'Experience_2024_06_14', including 'Provide Feedback' and 'Add Description' links, a 'Share' button, and a '+ New run' button. A search filter 'metrics.rmse < 1 and params.model = "tree"' is applied. Below the search bar are controls for 'Columns', 'Group by', and 'Sort: Created'. The 'Table' tab is selected, showing a single run: 'polite-snake-176', created '4 minutes ago', with a duration of '6.4s', source 'metrics.py', and model 'sklearn'. The bottom of the table indicates '1 matching run'.

Experiments

Search Experiments

Default

Experience_2024_06_14

Experience_2024_06_14

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

Time created

State: Active

Datasets

Sort: Created

Columns

Group by

Table

Chart

Evaluation

Experimental

Run Name	Created	Dataset	Duration	Source	Models
polite-snake-176	4 minutes ago	-	6.4s	metrics.py	sklearn

1 matching run

The screenshot shows the MLflow web interface. The top navigation bar includes 'main - Streamlit', 'FastAPI - Swagger UI', and 'MLflow'. The URL is '192.168.2.77:8003/#/experiments/904687765460583689/runs/b8d6fd594da94e5fb78facbc047d38f7/artifacts'. The page title is 'polite-snake-176'. The left sidebar shows a tree view of artifacts: 'decision_tree_model' (selected), 'metadata', 'MLmodel', 'conda.yaml', 'model.pkl', 'python_env.yaml', and 'requirements.txt'. The main content area displays the 'decision_tree_model' artifact details. It includes the path 'mlflow-artifacts/904687765460583689/b8d6fd594da94e5fb78facbc047d38f7/artifacts/decision_tree_model' and a 'Register model' button. Below this, the 'MLflow Model' section explains that the code snippets demonstrate how to make predictions using the logged model. The 'Model schema' section indicates that there is no schema. The 'Make Predictions' section provides two code snippets: one for predicting on a Spark DataFrame and another for predicting on a Pandas DataFrame.

decision_tree_model Register model

Path: [mlflow-artifacts/904687765460583689/b8d6fd594da94e5fb78facbc047d38f7/artifacts/decision_tree_model](#)

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also [register it to the model registry](#) to version control.

Model schema

Input and output schema for your model. [Learn more](#)

Name	Type
No schema. See MLflow docs for how to include input and output schema with your model.	

Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/b8d6fd594da94e5fb78facbc047d38f7/decision_tree_model'

# Load model as a Spark UDF. Override result_type if the model does not return double values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='double')

# Predict on a Spark DataFrame.
df.withColumn('predictions', loaded_model(struct(*map(col, df.columns))))
```

Predict on a Pandas DataFrame:

```
import mlflow
logged_model = 'runs:/b8d6fd594da94e5fb78facbc047d38f7/decision_tree_model'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)

# Predict on a Pandas DataFrame.
import pandas as pd
loaded_model.predict(pd.DataFrame(data))
```

The screenshot shows the MLflow web interface, specifically the 'Overview' tab for the 'polite-snake-176' experiment. The top navigation bar is the same as the previous screenshot. The URL is '192.168.2.77:8003/#/experiments/904687765460583689/runs/b8d6fd594da94e5fb78facbc047d38f7'. The page title is 'polite-snake-176'. The left sidebar shows the same tree view of artifacts. The main content area displays the 'Overview' tab, which contains a table of metadata and two sections: 'Parameters (2)' and 'Metrics (3)'. The metadata table includes fields like 'Created at', 'Created by', 'Experiment ID', 'Status', 'Run ID', 'Duration', 'Datasets used', 'Tags', 'Source', 'Logged models', and 'Registered models'. The 'Parameters (2)' section shows a search bar and a table with two parameters: 'max_depth' and 'min_samples_leaf'. The 'Metrics (3)' section shows a search bar and a table with three metrics: 'accuracy', 'precision', and 'recall'.

polite-snake-176 Register model

Overview Model metrics System metrics Artifacts

Created at	2024-06-14 23:14:31
Created by	root
Experiment ID	904687765460583689 🔗
Status	🟢 Finished
Run ID	b8d6fd594da94e5fb78facbc047d38f7 🔗
Duration	6.4s
Datasets used	—
Tags	Add
Source	📄 metrics.py
Logged models	📄 sklearn
Registered models	—

Parameters (2)

Search parameters

Parameter	Value
max_depth	3
min_samples_leaf	5

Metrics (3)

Search metrics

Metric	Value
accuracy	1
precision	1
recall	1

