MLflow and its usage

L. Berberi, V. Kozlov, K. Alibabaei, B. Esteban

Co-funded by
the European Union

Overview

- MLOps Requirements
- Introduction to MLflow
- MLflow Components
- MLflow Tracking Server deployed
- Conclusions

MLOps (use case) requirements

UC1.Req02/UC2.Req09/UC3.Req07- Organize and track all training experiments

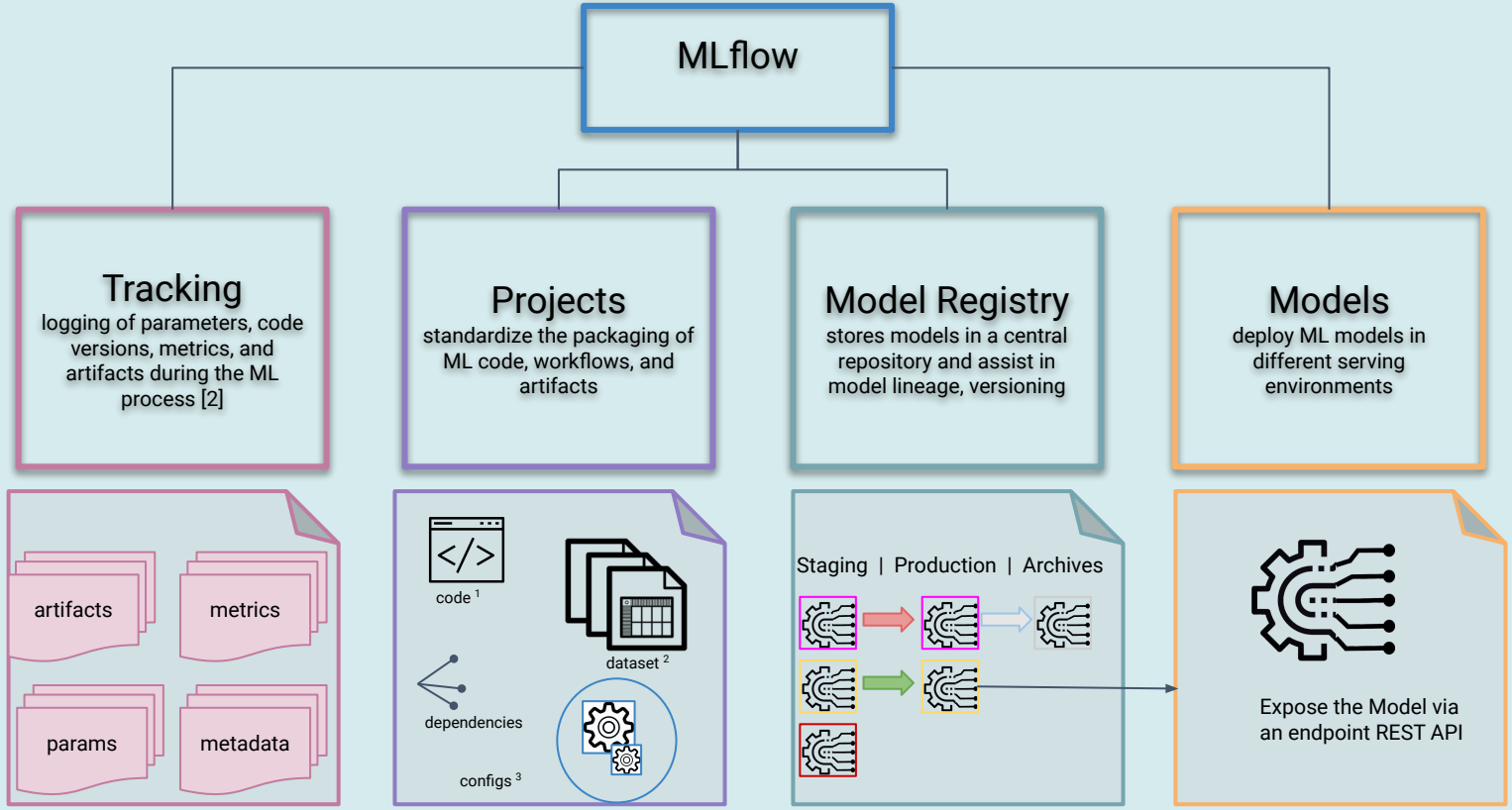
Level 2

- **MLflow** - first free and open-source MLOps product selected/tested after the landscaping activity (results presented at EGI Conf.)
  - Enhanced Experiment Management
    - facilitates efficient tracking and retrieval of historical experiments

Co-funded by the European Union

Introduction to MLflow

- An open source platform for the machine learning lifecycle
- mlflow 2.8.0 (latest release)



MLFlow Components

MLflow server instance deployment

- https://mlflow.dev.ai4eosc.eu
- Service capacity: 40 GB /root and 91GB GB mnt disk space
  - Backend:
    - Postgres SQL dB (store models, metrics, exp)
    - SQLite dB (store users, permissions of experiments and Registered Models)
  - Frontend: MLflow UI (experiment/running info, metrics, analyze and compare runs/exps e

- Developed a new dockerized MLFlow tracking server solution [3]
  - automatic backups and manually restore operations are written for both dBs.
  - traefik as reverse proxy + SSL certificates enabled
  - custom auth(oidc integration) plugin is under developm

- In [5]: instructions how to setup your own MLflow server instan

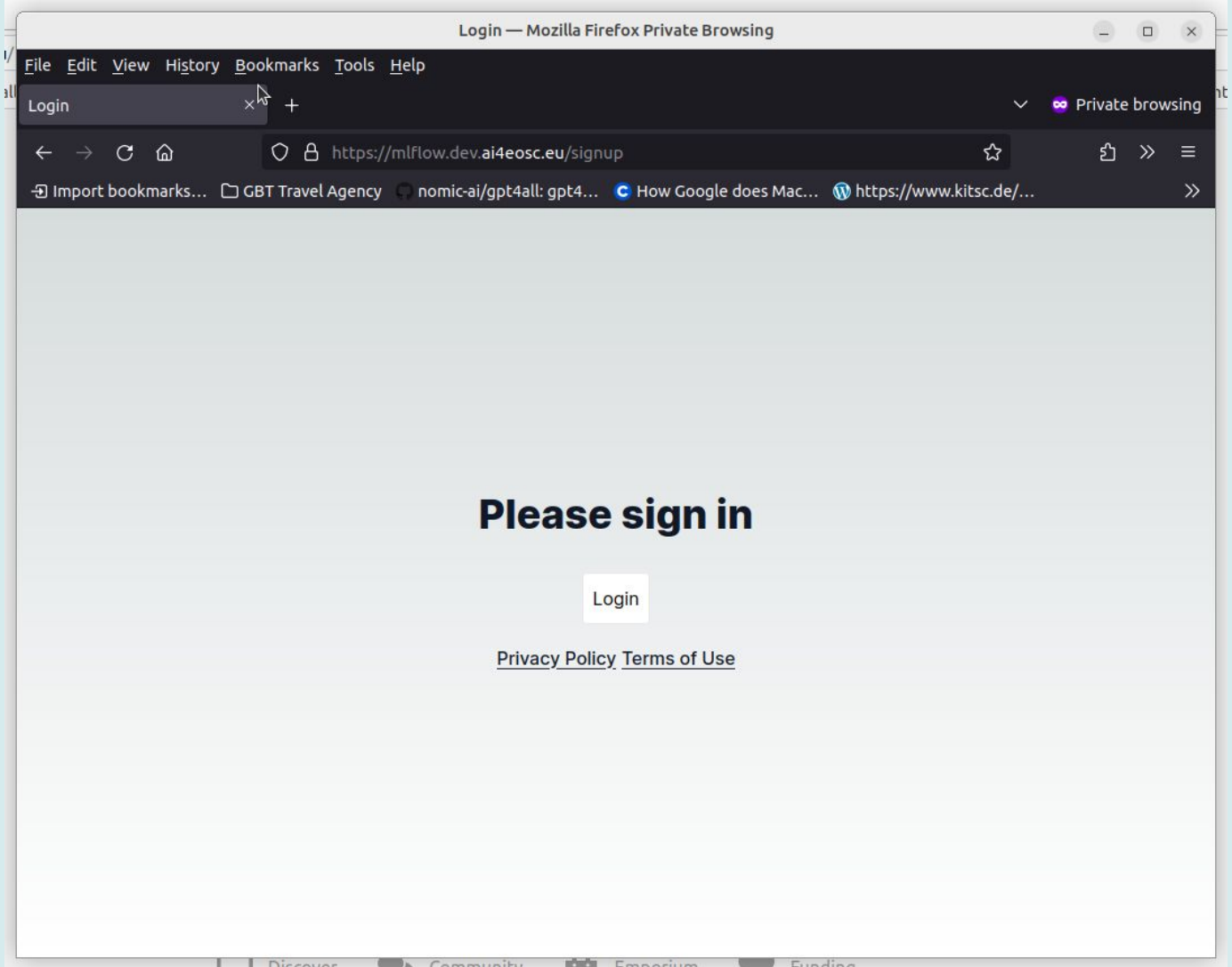
Built containers:

```
$  sudo docker ps --format '{{.Names}}\t{{.Status}}'
mlflow-compose-backup_db-1      Up 11 hours (healthy)
mlflow-compose-reverse-proxy-1  Up 12 hours
mlflow-compose-backend-1        Up 12 hours
mlflow-compose-database-1       Up 12 hours (healthy)
```

- MLflow Authentication (basic-auth) as a plugin
  - username and password (= credentials you provided from the self-registration with oidc auth)

MLflow self-user registration

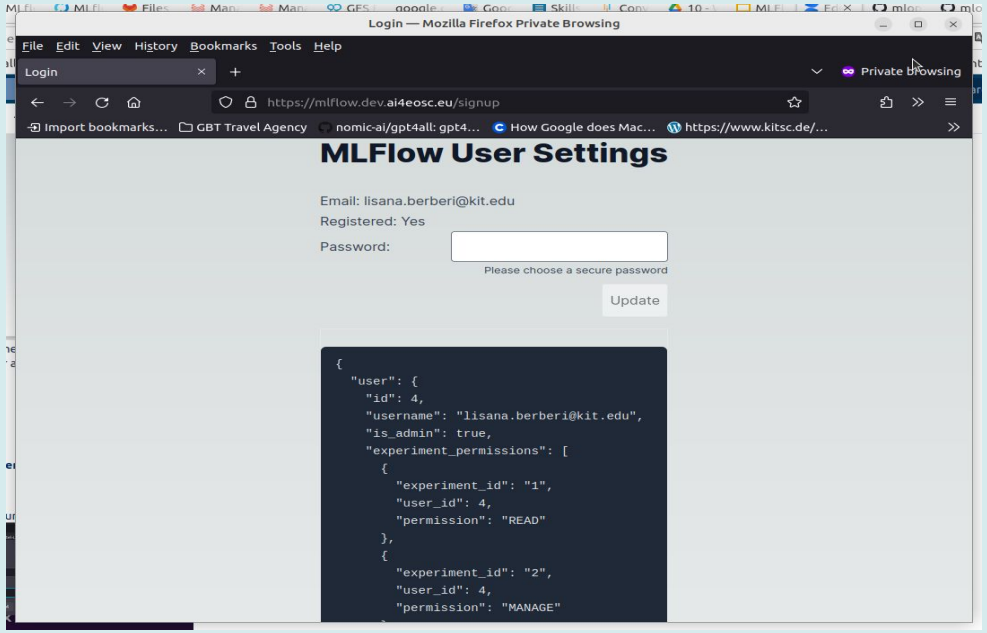
Click **Login** button, login via EGI Check-In the same way as you registered for vo.ai4eosc.eu (click your Institute name/



Authenticate in the MLFlow UI (frontend interface) with your credentials (via email)

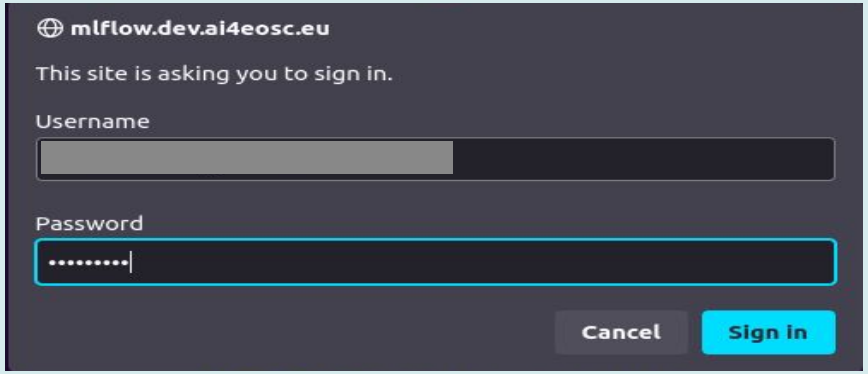
MLflow self-user registration

Click **Login** button, login via EGI Check-In the same way as you registered for vo.ai4eosc.eu (click your Institute name/

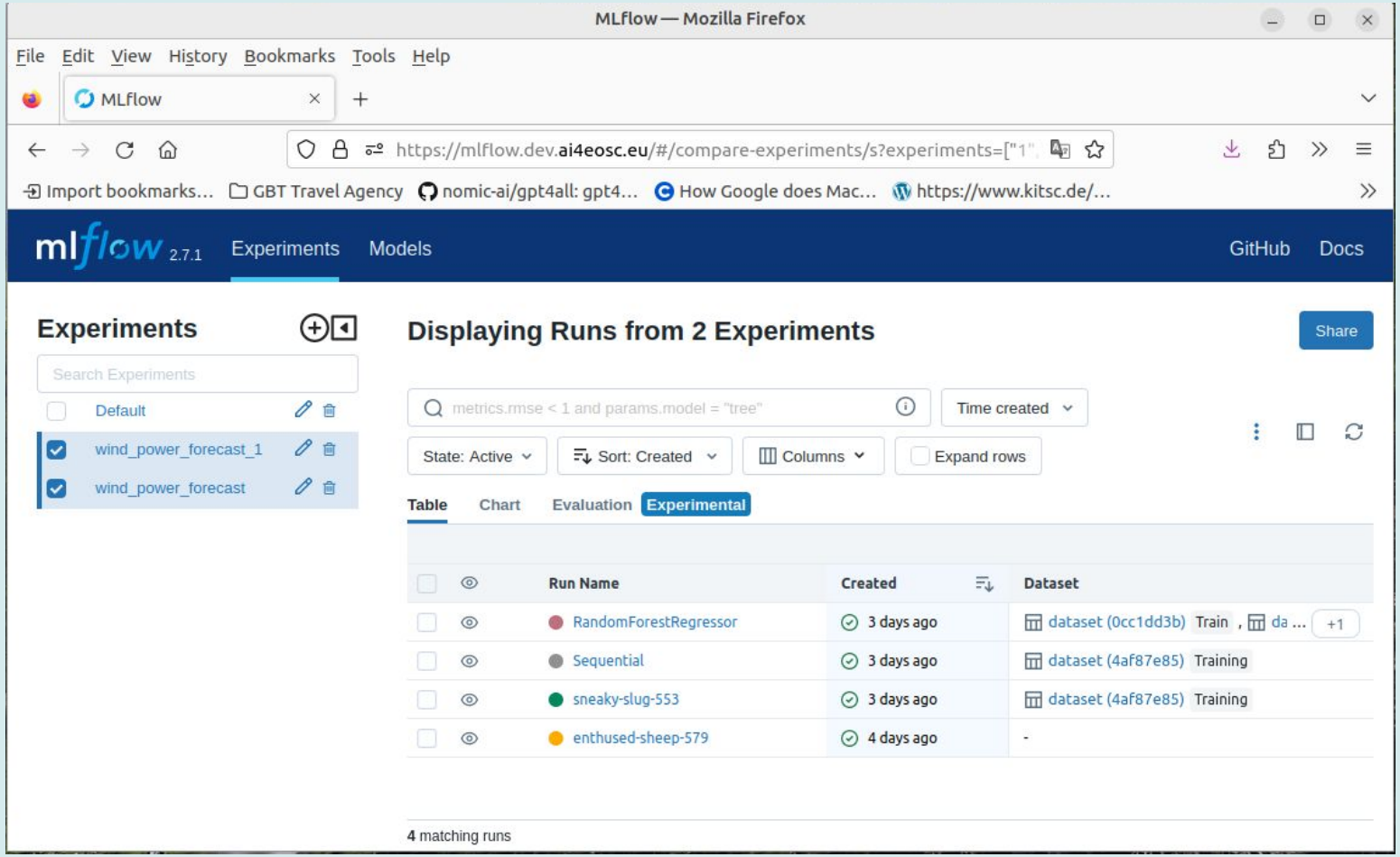


Enter a new password in the textbox "**Password**" and then click Update button.

Authenticate in the **MLFlow UI** (frontend interface) with your credentials (via email)

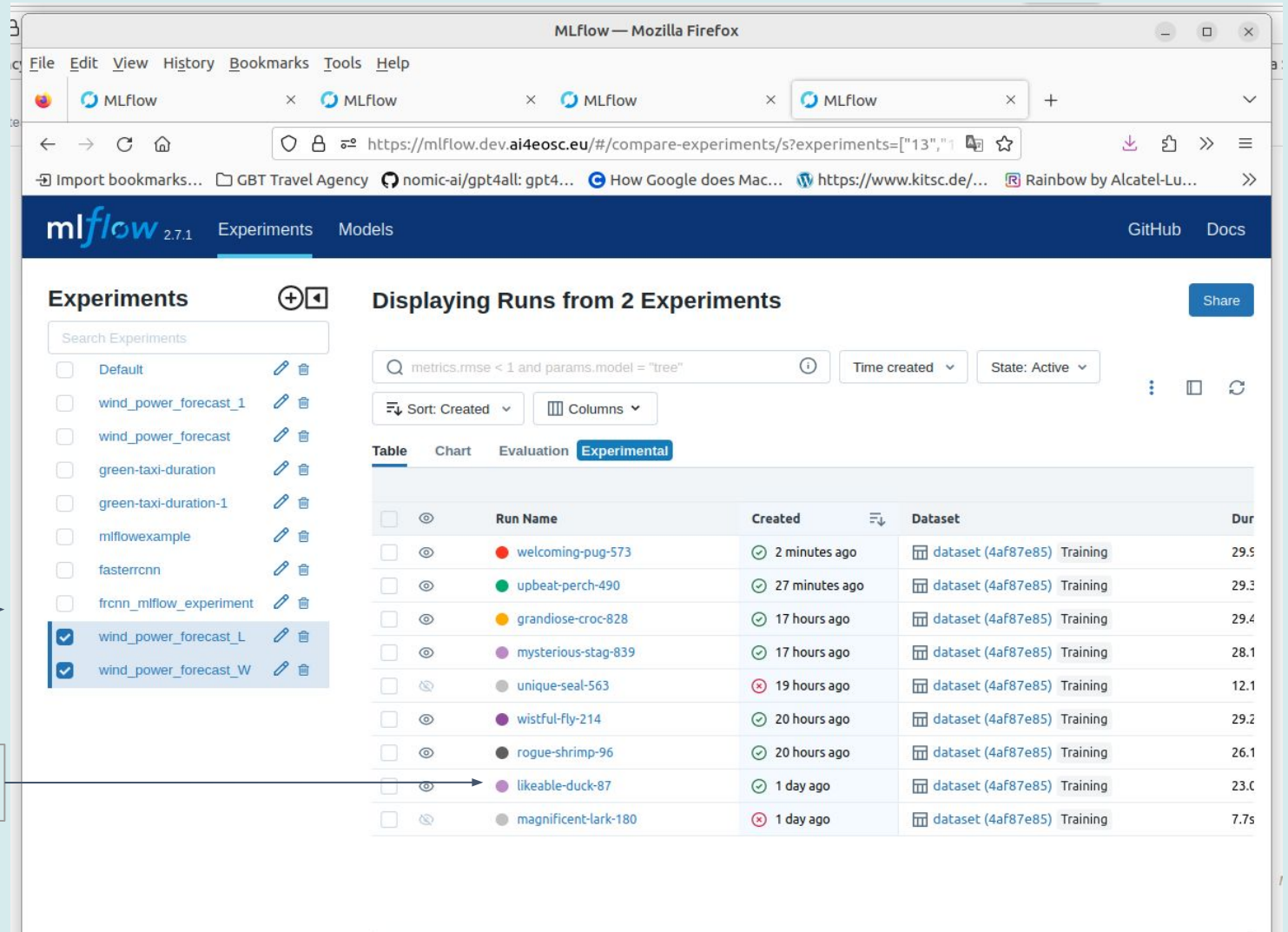


Tracking Experiments



Tracking Experiments/Runs

- MLflow Experiment: *is the primary unit of organization and access control for MLflow runs*; all MLflow runs belong to an experiment.
- Run: is a *collection of parameters, metrics, tags, and artifacts associated with a machine* learning model training process.



Experiment_name

runs_name (default: randomly generated)

Examples:

`mlflow.tensorflow` → module provides an API for logging and loading TensorFlow models.

`mlflow.pytorch` → module provides an API for logging and loading PyTorch models.

Tracking Experiments/Runs

datasets

params/metrics

Artifacts

MLflow signature

Code snippets

mlflow 2.7.1 Experiments Models GitHub Docs

wind_power_forecast ›

RandomForestRegressor

Run ID: 4d2a449fe17c4dc096510516e39d1801 Date: 2023-11-02 21:53:10 Source: ipykernel_launcher.py User: lisana.berberi@kit.edu Duration: 32.2s

Status: FINISHED Lifecycle Stage: active

› Description Edit

⌄ Datasets (2)

dataset (0cc1dd3b) Train

dataset (bbdd0425) Eval

› Parameters (17)

› Metrics (7)

› Tags (2)

⌄ Artifacts

model

estimator.html

metric_info.json

Full Path:mlflow-artifacts:/5/4d2a449fe17c4dc096510516e39d1801/artifacts/model

Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. This model is also registered to the model registry.

Model schema

Input and output schema for your model. Learn more

Name	Type
Inputs (9)	
temperature_00	double
wind_direction_00	double
wind_speed_00	double
temperature_08	double
wind_direction_08	double
Outputs (1)	

Make Predictions

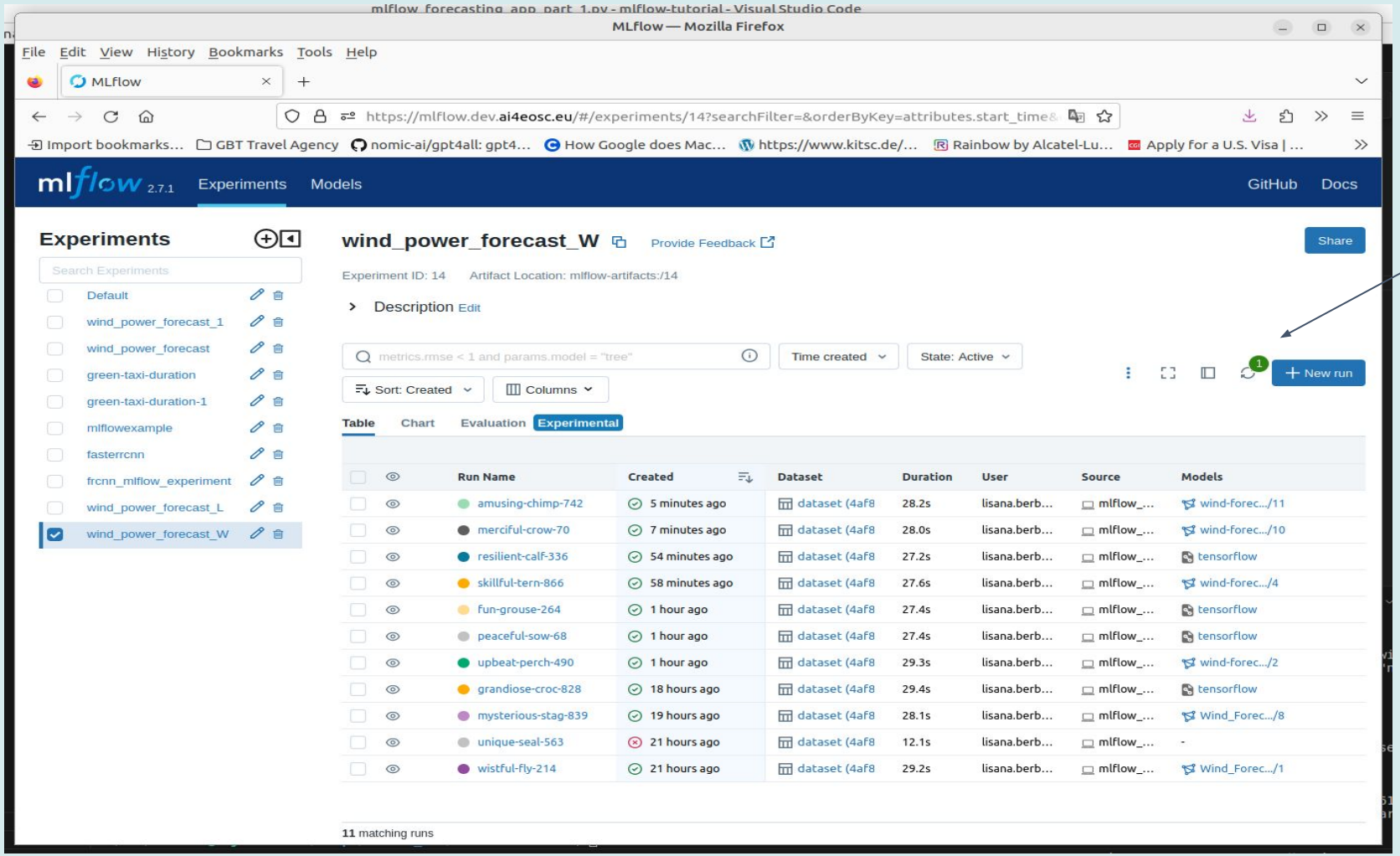
Predict on a Spark DataFrame:

```
import mlflow
from pyspark.sql.functions import struct, col
logged_model = 'runs:/4d2a449fe17c4dc096510516e39d1801/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:/4d2a449fe17c4dc096510516e39d1801/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))
```

Tracking Experiments/Runs



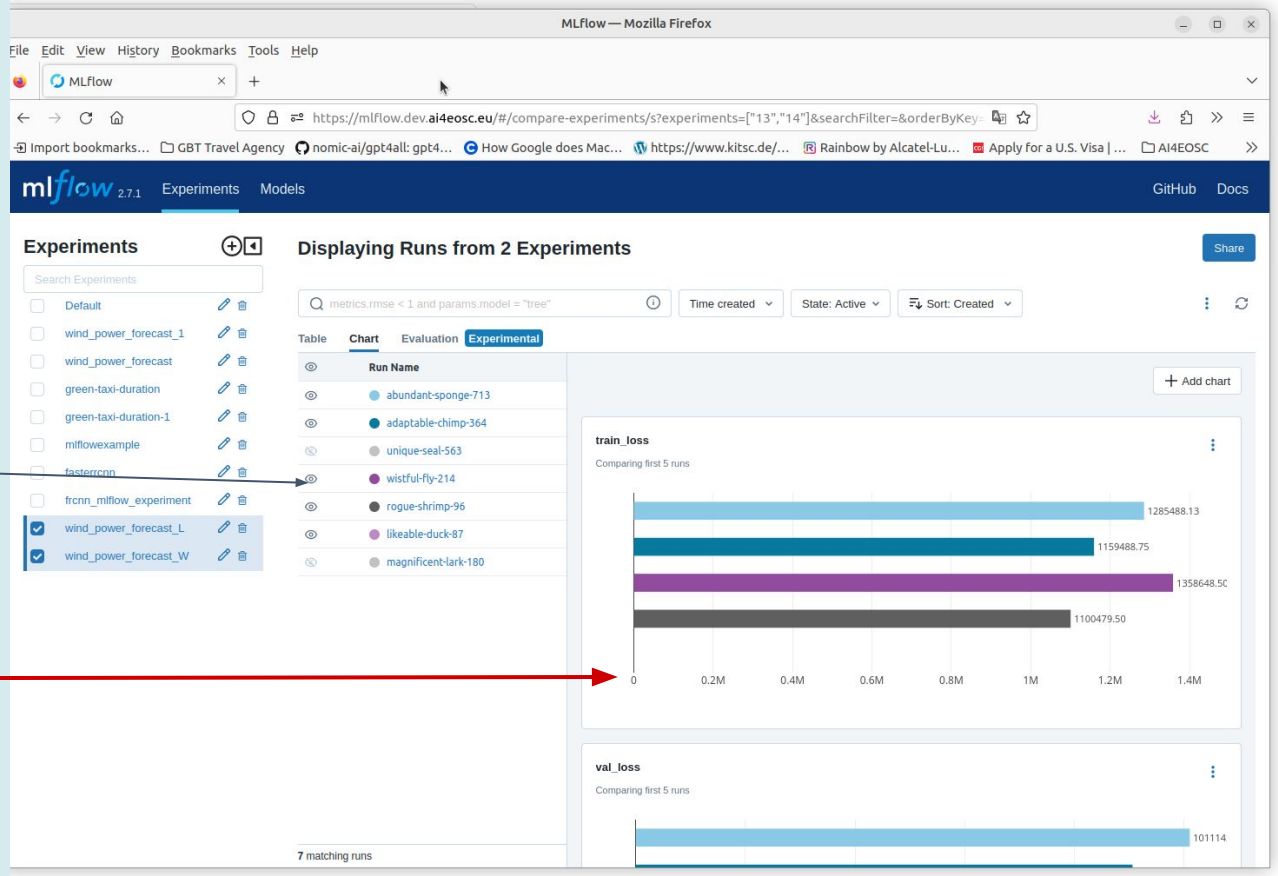
new run

Comparing Experiments runs

- Select experiments you want to compare and the runs you want to show in the chart

runs (hide)

chart visualisations



Packaging projects

AI4 | eosc

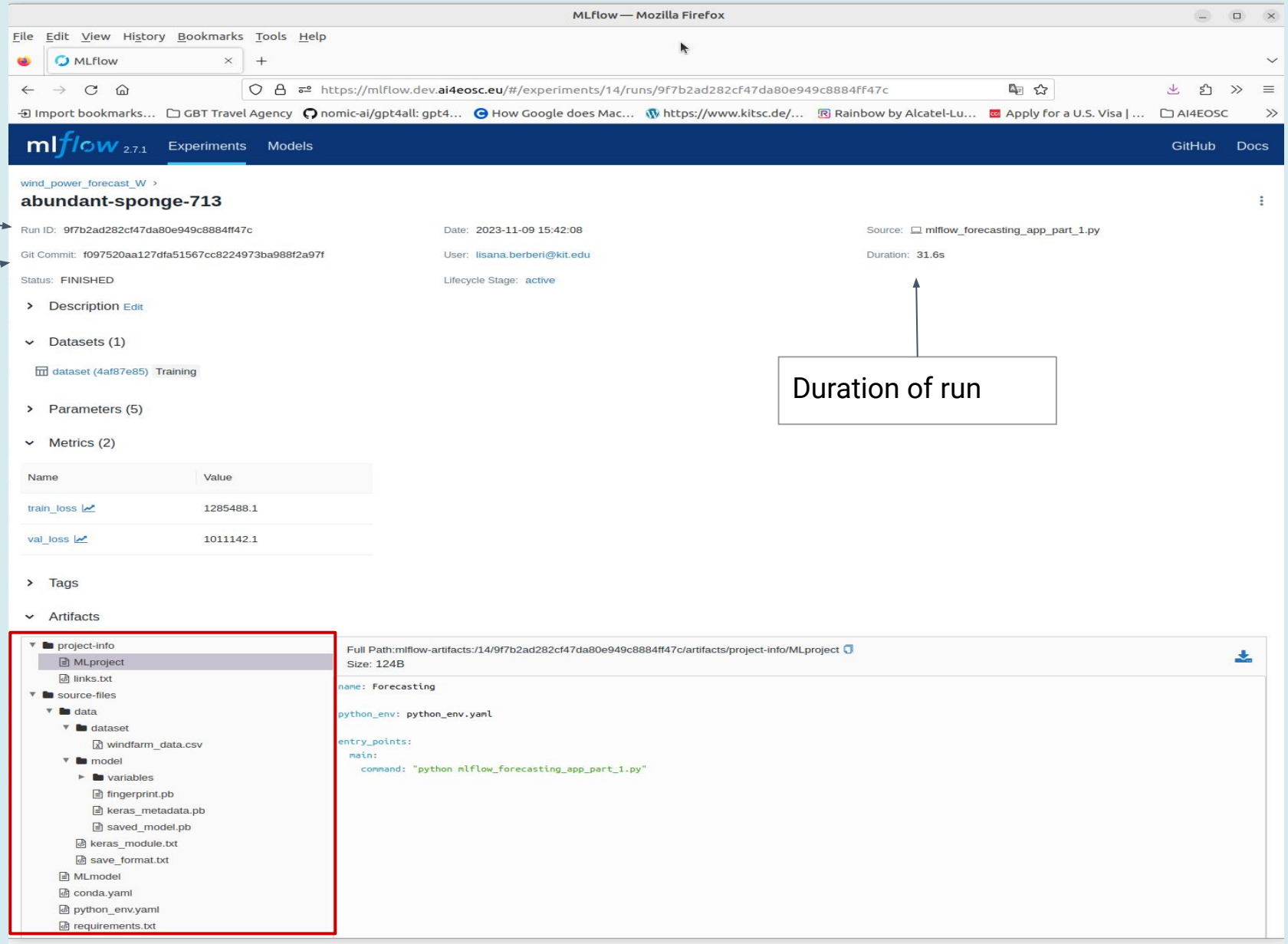
Co-funded by the European Union

package ML code
in a format to reproduce
runs on any platform

Run_ID

Git Commit hash

MLproject.yaml file



Duration of run

`mlflow.projects` → module provides an API for running MLflow projects locally or remotely[7]

Model Registry

centralized model store

-Register a new model

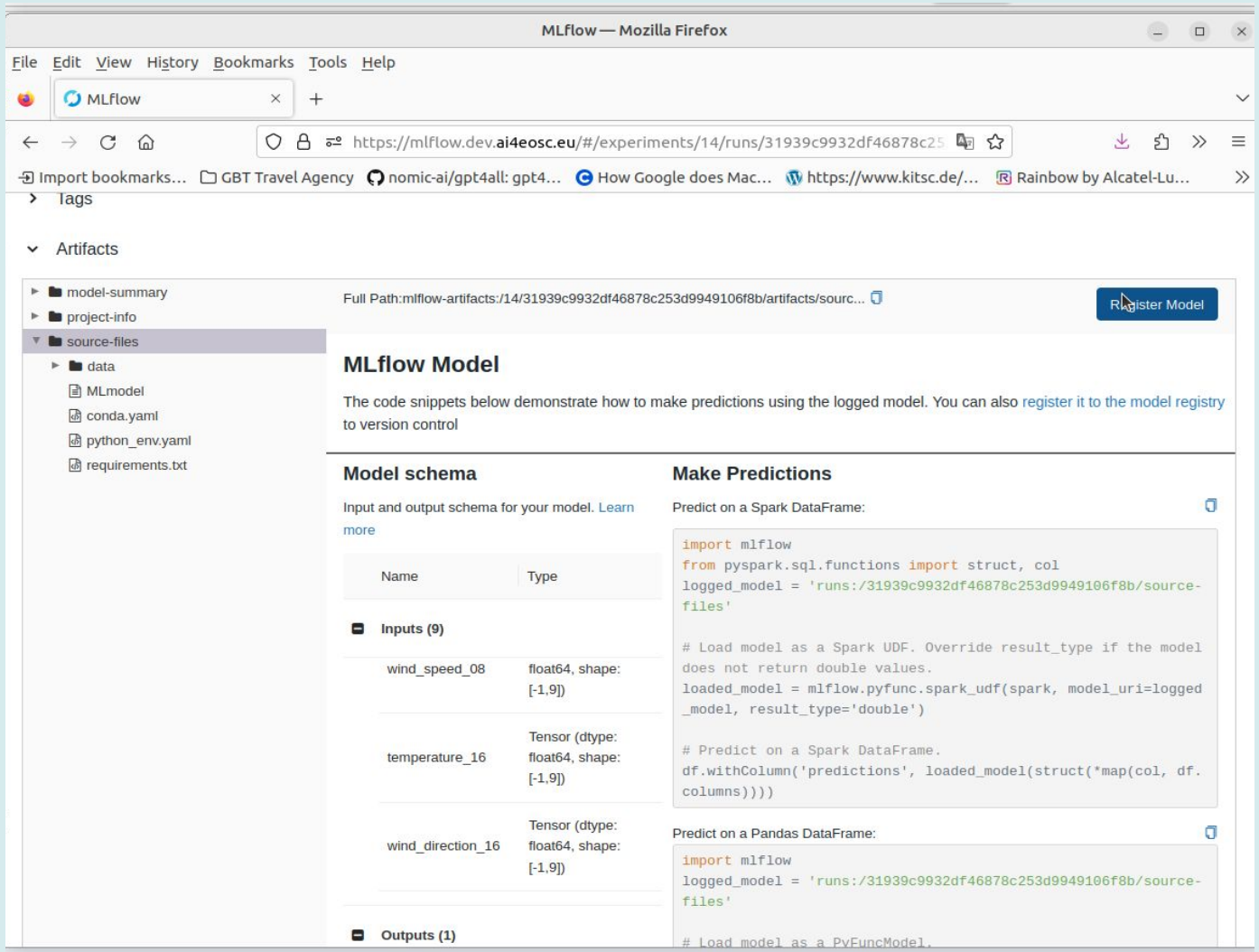
Model_name must be unique
!!! a new version of that model will be created (auto increment version nr)

- during MLflow experiment run or

```
mlflow.<flavor>.log_model(model,
artifact_path="source-files",
                                signature=signature,
            registered_model_name=MLFLOW_MODEL_NAME)
```

- after your experiment runs.

```
mlflow.register_model(
    f"runs:/{run_id}/artifacts/source-files", MLFLOW_MODEL_NAME
)
```



Model Registry

model tags

MLflow — Mozilla Firefox

File Edit View History Bookmarks Tools Help

MLflow

https://mlflow.dev.ai4eosc.eu/#/models/wind-forecast-seq-model-v3.0

Import bookmarks... GBT Travel Agency nomic-ai/gpt4all: gpt4... How Google does Mac... https://www.kitsc.de/...

mlflow 2.7.1 Experiments Models GitHub Docs

Registered Models ›

wind-forecast-seq-model-v3.0

Created Time: 2023-11-10 13:45:18 Last Modified: 2023-11-10 13:50:06

Description Edit

Tags

Name	Value	Actions
author	lisana.berberi@kit.edu	
framework	tensorflow	
task	classification	

Name Value Add

Versions All Active 2 Compare

Version	Registered at	Created by	Stage	Description
Version 2	2023-11-10 13:48:16		Production	
Version 1	2023-11-10 13:45:20		Staging	

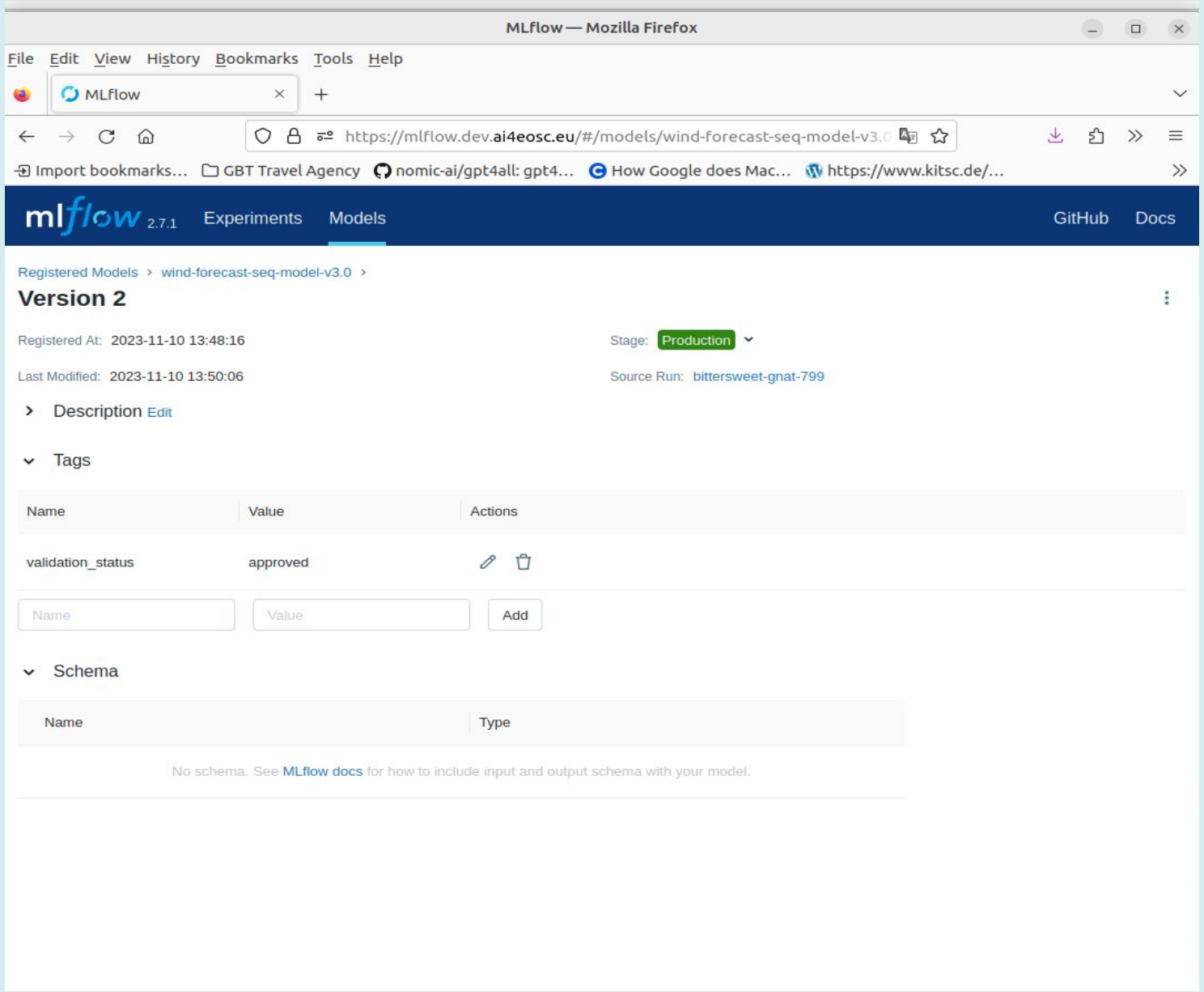
1

Model Registry

- fetch the model

`mlflow.<model_flavor>.load_model()`, or more generally, `load_model()`. You can use the loaded model for one off predictions or in inference workloads such as batch inference.

model version tags



Model Serving

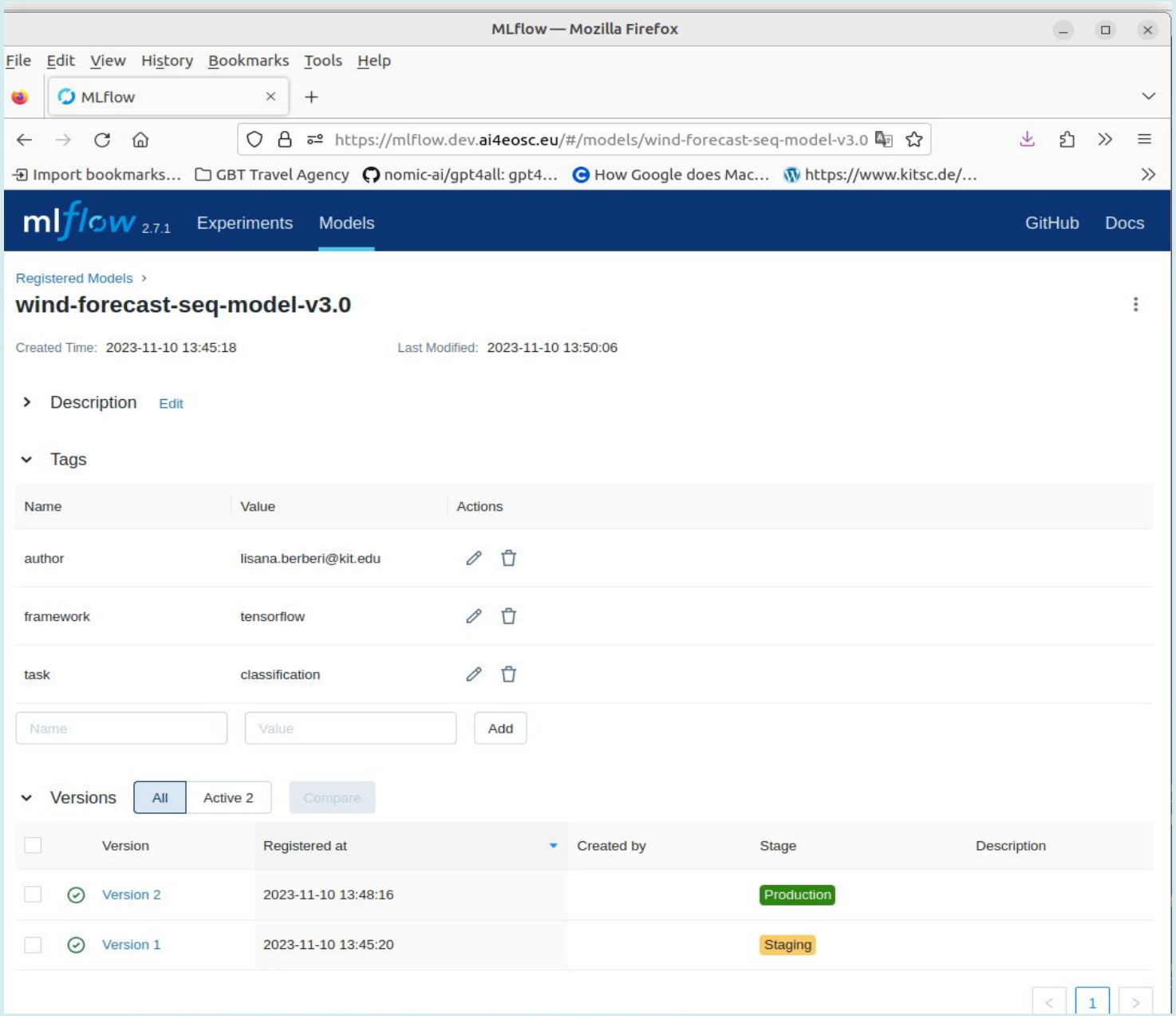
- fetch the model

  `mlflow.<model_flavor>.load_model()`, or more generally, `load_model()`. You can use the loaded model for one off predictions or in inference workloads such as batch inference.

```
client = MlflowClient()
model_version = client.get_latest_versions(model_name,
stages=[model_stage])[0].version
model_uri = F"models:/{model_name}/{model_stage}"
model = mlflow.pyfunc.load_model(model_uri)
```

- serve the model (deploy)
  - to run model inference

mlflow models serve --model-uri models:/<model-name>/Production -h <hostname> -p 5001



How to log your own experiment?

Example app: Predict the power output information for a wind farm in the US

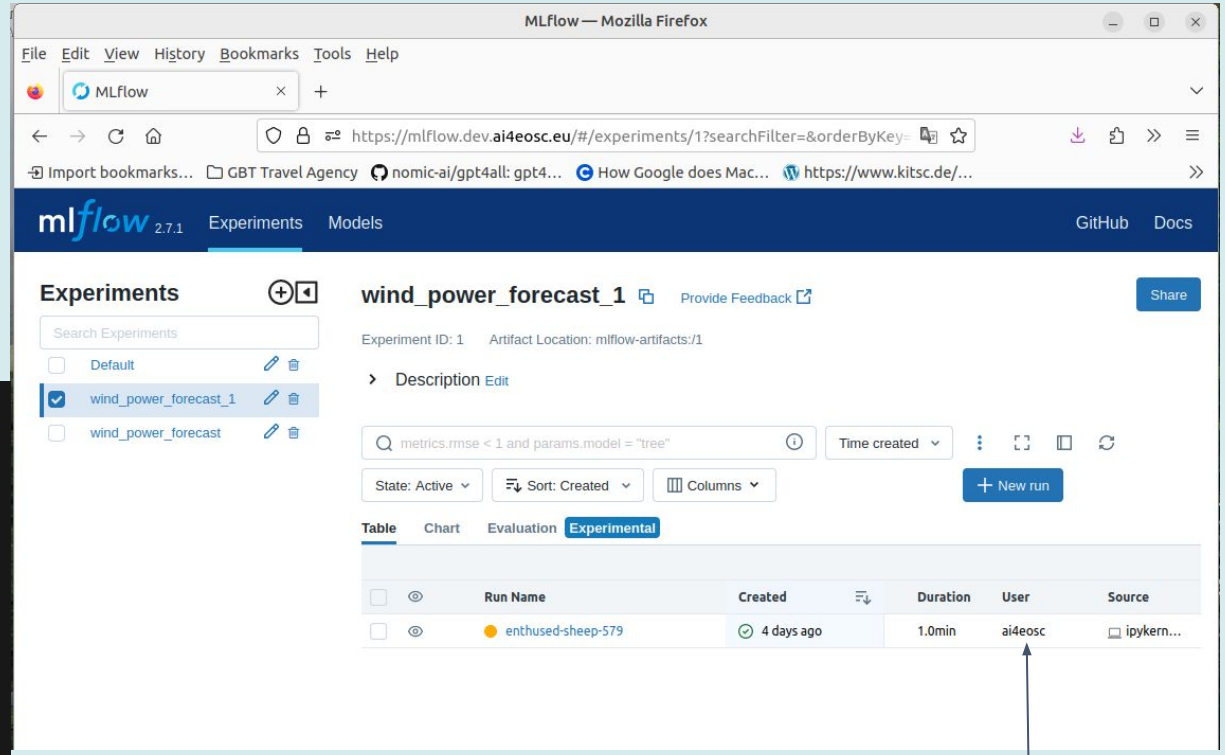
```
git url:  https://git.scc.kit.edu/m-team/ai/mlflow-tutorial.git
pip install -r requirements.txt;
python mlflow-example/mlflow_forecasting_app_part_1.py

or launch the notebook
```

mlflow_forecasting_app_v1.2.ipynb

```
# #### MLflow part
#
# **! Configure IMPORTANT CONSTANTS !:**

#set the environmental vars to allow 'mlflow_user' to track experiments using MLFlow
import os
import getpass

# IMPORTANT CONSTANTS TO DEFINE
# MLFLOW CREDENTIALS (Nginx). PUT REAL ONES!
# for direct API calls via HTTP we need to inject credentials
MLFLOW_TRACKING_USERNAME = input('Enter your username: ')
MLFLOW_TRACKING_PASSWORD = getpass.getpass() # inject password by typing manually
# for MLFLow-way we have to set the following environment variables
os.environ['MLFLOW_TRACKING_USERNAME'] = MLFLOW_TRACKING_USERNAME
os.environ['MLFLOW_TRACKING_PASSWORD'] = MLFLOW_TRACKING_PASSWORD
```

How to share your experiment?

- Once you logged your experiment, you can control user access and permissions to it.
- Follow instructions in [5] how to grant/revoke specific permissions to user for an experiment [scripts available in [6]]



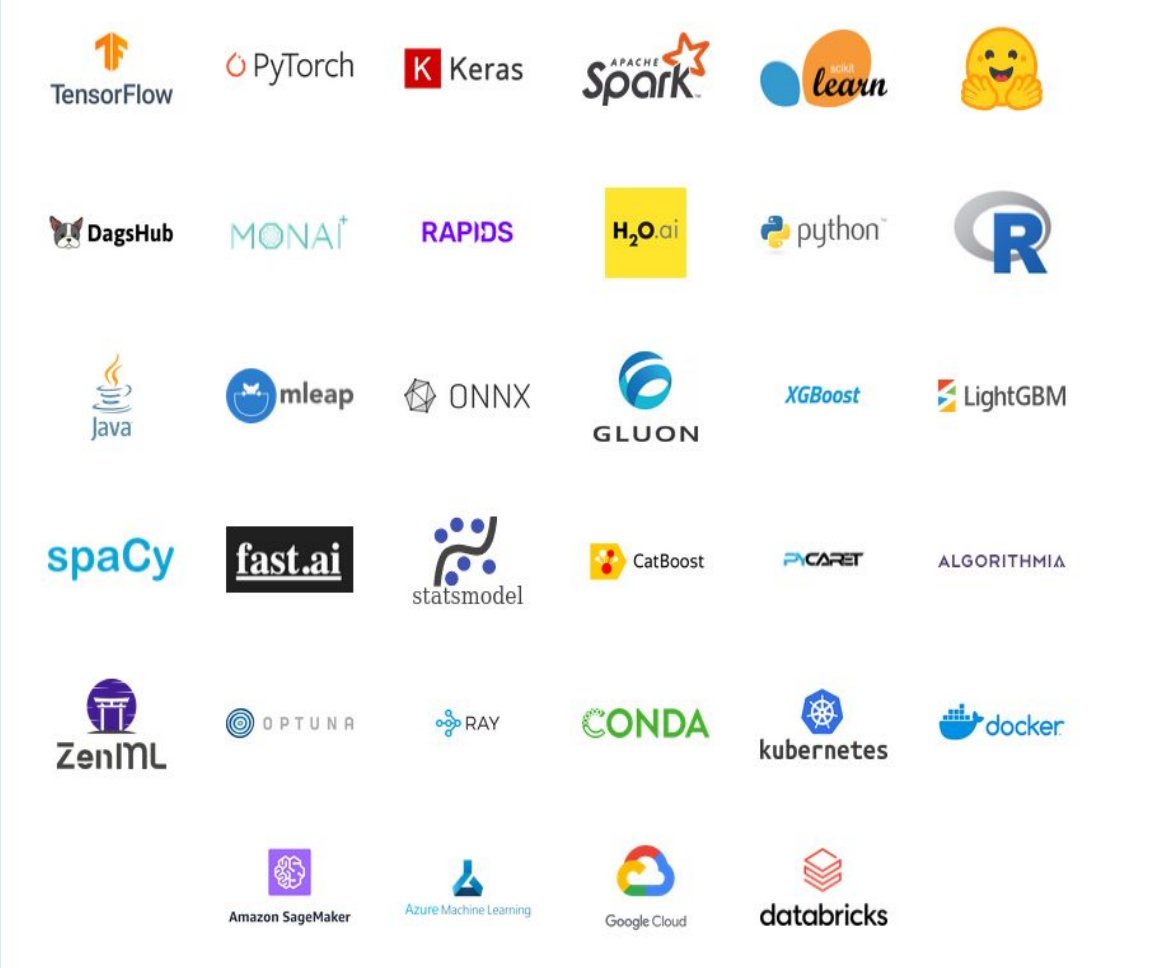

```
Enter your username: lisana.berberi@kit.edu
Enter your password:

Authentication successful.
Menu:
0. List all the experiments granted to you
1. Add Experiment-User Permission
2. Get Experiment-User Permission
3. Update Experiment-User Permission
4. Delete Experiment-User Permission
5. List Runs for an Experiment
6. Add RegisteredModel-User Permission
7. Get RegisteredModel-User Permission
8. Update RegisteredModel-User Permission
9. Delete RegisteredModel-User Permission
10. List RegisteredModel
11. Exit
Enter your choice (0/1/2/3/4/5/6/7/8/9/10/11): 0
Enter a possible experiment-name: wind

Experiment Name: wind_power_forecast

Experiment ID: 5

Artifact Location: mlflow-artifacts:/5

Experiment Name: wind_power_forecast_1

Experiment ID: 1

Artifact Location: mlflow-artifacts:/1
```

```
Menu:
0. List all the experiments granted to you
1. Add Experiment-User Permission
2. Get Experiment-User Permission
3. Update Experiment-User Permission
4. Delete Experiment-User Permission
5. List Runs for an Experiment
6. Add RegisteredModel-User Permission
7. Get RegisteredModel-User Permission
8. Update RegisteredModel-User Permission
9. Delete RegisteredModel-User Permission
10. List RegisteredModel
11. Exit
Enter your choice (0/1/2/3/4/5/6/7/8/9/10/11): 2
Enter the experiment name: wind_power_forecast_1
Experiment Name: wind_power_forecast_1
Experiment ID: 1
Artifact Location: mlflow-artifacts:/1
Enter the username you want to show what permissions (s)he have for the e
xperiment: lisana.berberi@kit.edu
Experiment Name: wind_power_forecast_1
Experiment ID: 1
Artifact Location: mlflow-artifacts:/1
{'experiment_permission': {'experiment_id': '1', 'permission': 'READ', 'u
ser_id': 4}}
Experiment ID: 1
User ID: 4 /username: lisana.berberi@kit.edu
Permission_array: READ
```

User with only “Read” permission to that experiment

User who logged the experiment run

MLflow integrations and community support



MLflow limitations

- Security concerns
- UI simple design
- Lack of user (!fixed partly in the new version) and group management
- Scalability and performance concerns
- Configuration and maintenance overhead

MLOps (paid) alternatives

- AzureML
- Weight & Biases
- Neptune.ai
- Comet ML
- etc..

Conclusions

- **Improved Efficiency**: MLflow's streamlined experiment tracking and management significantly reduce the time spent on manual record-keeping

- **Cross-Team Collaboration and Knowledge Sharing**: The centralized approach of MLflow has fostered collaboration among diverse teams involved in the ML process

- **Reproducibility**: comprehensive experiment tracking and versioning capabilities provide a robust foundation for reproducibility

References

1. MLflow server: https://mlflow.dev.ai4eosc.eu
2. MLflow core components: https://www.mlflow.org/docs/latest/introduction/index.html#core-components-of-mlflow
3. MLflow docker compose: https://git.scc.kit.edu/m-team/ai/mlflow-compose
4. MLflow GitHub repo: https://github.com/mlflow/mlflow
5. MLflow server docker instructions: https://confluence.ifca.es/x/HQDRC
6. MLflow user and control access/permissionshttps://git.scc.kit.edu/m-team/ai/mlflow_auth/-/tree/main?ref_type=heads
7. MLflow Project- Python APIhttps://mlflow.org/docs/latest/python_api/mlflow.projects.html
8.

AI4 | eosc

Co-funded by the European Union

AI4EOSC

ai4eosc-wp6@listas.csic.es

ai4eosc.eu

Thank you! Any questions?