Getting Started with MLflow

MLflow is an open-source platform designed to manage the entire machine learning lifecycle. It provides tools for managing experiments, tracking metrics, packaging models, and deploying them into production environments.

What is MLflow?

MLflow offers four main components to simplify ML lifecycle management:

- **MLflow Tracking**: Log parameters, metrics, and artifacts to track experiments.
- **MLflow Projects**: Standardize machine learning code in reproducible formats.
- **MLflow Models**: Manage and deploy models with a consistent format.
- **MLflow Registry**: Centralized repository to store and manage model versions.

Using MLflow

1. Install MLflow

> Note: you are recommended to setup an environment dedicated for MLFlow, either with conda or pipenv.

```
pip install mlflow
```

Verify the installation:

```
mlflow --version
```

You can open the MLFlow ui by running

```
mlflow ui
```

this sets up a local server and opens a browser window to the MLFlow ui.

2. Set Up an Experiment

follow the instructions in `MLFlowSetup.ipynb`

this notebook mentions:

- how to create a new experiment
- how to log parameters and metrics
- how to register models into MLFlow registry

- how to load models from MLFlow registry
- how to connect to a remote repository (remote tracking URI section)

---

3. View Experiment Results

Launch the MLflow tracking UI to visualize the logged parameters, metrics, and artifacts:

```
mlflow ui
```

By default, the UI is hosted at `http://127.0.0.1:5000`. Navigate to your browser and explore the experiment details.

---

4. Connect to a remote repository

to collaborate with others, setup a remote repository.

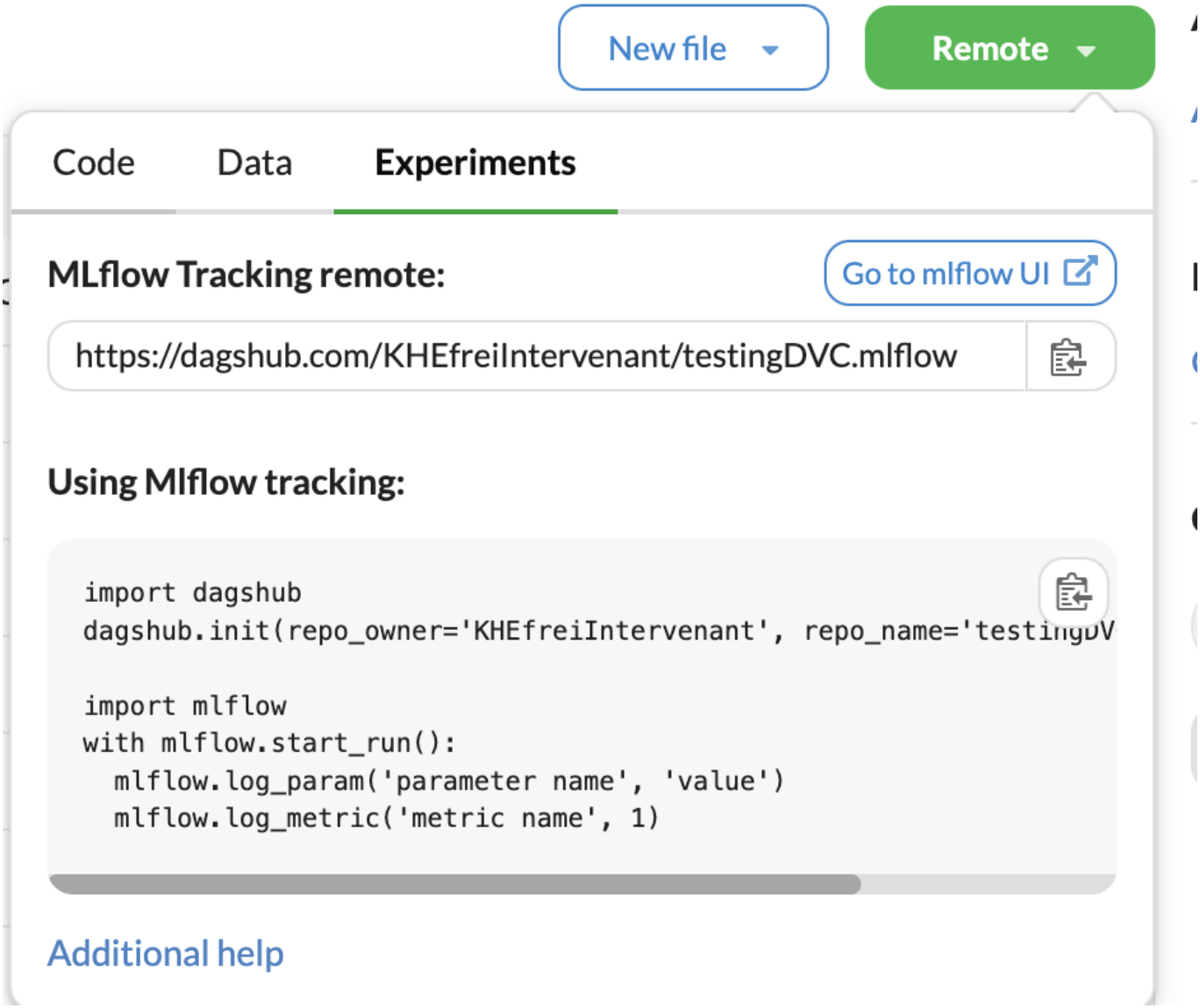
Firstly, create an account at dagshub. It allows for creating a free account, which we can then link to our github Repo that contains the MLFlow experiment code. Follow the instructions here to connect your github repo to your dagshub account.

Now, we first need to install dagshub from pip to access it in our Jupyter Notebook:

```
pip install dagshub
```

and then configure the tracking uri to point to our dagshub account.

Reffer to the `MLFlowSetup.ipynb` notebook for the steps to configure the tracking uri. There is a section in the notebook mentioning code that can be fetched from the dagshub dashboard. The image below shows where to find the code.



---

5. Serve Models

To serve a model locally from MLFlow, use the command:

```
mlflow models serve -m runs:/<run_id>/<model_name> -p 5003
# or choose a different port
```

You might face an issue asking you to install `virtualenv`. If so, you can install it by running:

```
pip install virtualenv
```

MLFlow supports deploying models on a variety of platforms, AWS, Azure, Kubernetese, databricks, ... Refer to the MLFlow Models documentation for more details.