

## Paired With Hira Fatima (2211-012-KHI-DEG) Sayam Naqvi(2211-031-KHI-DEG) Rumaisa shahab

```
In [1]: !python -c "import sys; print(sys.executable)"  
/home/anooshamalik/mlflow_assignment/mlops_assignment/bin/python
```

## MLFlow lab

### Setting up MLFlow tracking server

We also specify artifact root and backend store URI. This makes it possible to store models.

After running this command tracking server will be accessible at `localhost:5000`

```
In [4]: %%bash --bg  
  
mlflow server --host 0.0.0.0 \  
    --port 5000 \  
    --backend-store-uri sqlite:///mlflow.db \  
    --default-artifact-root ./mlruns
```

### MLProject file

This file is used to configure MLFlow steps.

Using `MLproject` we can define our project's pipeline steps, called *entry points*.

Each entry point in this file corresponds to a shell command.

Entry points can be ran using

```
mlflow run -e <ENTRY_POINT>
```

By default `mlflow run` runs `main` entrypoint.

```
In [5]: %cat MLproject
```

```
name: basic_mlflow

# this file is used to configure Python package dependencies.
# it uses Anaconda, but it can be also alternatively configured to use pip.
conda_env: conda.yaml

# entry points can be ran using `mlflow run <project_name> -e <entry_point_name>`
entry_points:

  # MLproject file has to have main entry_point. It can be toggled without using -e option.
  main:
    # parameters is a key-value collection.
    parameters:
      file_name:
        type: str
        default: "day.csv"
      max_k:
        type: int
        default: 10
    command: "python train.py {file_name} {max_k}"
```

First we need to download data. We will use weather data from previous machine learning tutorial.

```
In [28]: %%bash
source mlflow_env_vars.sh
mlflow run .
```

```
2022/12/19 18:30:40 INFO mlflow.utils.conda: Conda environment mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33 already exists.
2022/12/19 18:30:40 INFO mlflow.projects.utils: === Created directory /tmp/tpmpeycxlk86 for downloading remote URIs passed to arguments of type 'path' ===
2022/12/19 18:30:40 INFO mlflow.projects.backend.local: === Running command 'source /home/anooshamalik/anaconda3/bin/./etc/profile.d/conda.sh && conda activate mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33 1>&2 && python train-Copy1.py 10' in run with ID 'caf9946624b74883ba3f297c30a9f836' ===
/home/anooshamalik/anaconda3/envs/mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33/lib/python3.10/site-packages/_distutils_hack/__init__.py:33: UserWarning: Setuptools is replacing distutils.
  warnings.warn("Setuptools is replacing distutils.")
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:43 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 28
Created version '28' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:44 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 29
Created version '29' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:45 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 30
Created version '30' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:46 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 31
Created version '31' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:47 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 32
Created version '32' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:48 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 33
Created version '33' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:49 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 34
Created version '34' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:30:50 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
```

```
Model name: sklearn_knn, version 35
Created version '35' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of
this model...
2022/12/19 18:30:51 INFO mlflow.tracking._model_registry.client: Waiting
up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 36
Created version '36' of model 'sklearn_knn'.
2022/12/19 18:30:51 INFO mlflow.projects: === Run (ID 'caf9946624b74883b
a3f297c30a9f836') succeeded ===
```

## Training

Now we can train models. See `train.py`. It contains code from supervised machine learning tutorial; we added tracking metrics and model.

We will train kNN models for  $k \in \{1, 2, \dots, 10\}$  using *temperature* and *casual* features.

After running this command you can go to `localhost:5000` and see the trained models.

```
In [33]: import sklearn
```

```
In [34]: sklearn.__version__
```

```
Out[34]: '1.2.0'
```

```
In [35]: %%bash
source mlflow_env_vars.sh
mlflow run .
```

```
2022/12/19 18:33:02 INFO mlflow.utils.conda: Conda environment mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33 already exists.
2022/12/19 18:33:02 INFO mlflow.projects.utils: === Created directory /tmp/tmp82p4nqp0 for downloading remote URIs passed to arguments of type 'path' ===
2022/12/19 18:33:02 INFO mlflow.projects.backend.local: === Running command 'source /home/anooshamalik/anaconda3/bin/./etc/profile.d/conda.sh && conda activate mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33 1>&2 && python train-Copy1.py 10' in run with ID 'b8a245a791ba47d6b7487463a88e1f60' ===
/home/anooshamalik/anaconda3/envs/mlflow-dd0fbdd40ba98798131458f29496394bd1a3fb33/lib/python3.10/site-packages/_distutils_hack/__init__.py:33: UserWarning: Setuptools is replacing distutils.
  warnings.warn("Setuptools is replacing distutils.")
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:04 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 37
Created version '37' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:05 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 38
Created version '38' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:07 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 39
Created version '39' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:08 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 40
Created version '40' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:09 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 41
Created version '41' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:10 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 42
Created version '42' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:11 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 43
Created version '43' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of this model...
2022/12/19 18:33:12 INFO mlflow.tracking._model_registry.client: Waiting up to 300 seconds for model version to finish creation.
```

```

Model name: sklearn_knn, version 44
Created version '44' of model 'sklearn_knn'.
Registered model 'sklearn_knn' already exists. Creating a new version of
this model...
2022/12/19 18:33:13 INFO mlflow.tracking._model_registry.client: Waiting
up to 300 seconds for model version to finish creation.
Model name: sklearn_knn, version 45
Created version '45' of model 'sklearn_knn'.
2022/12/19 18:33:13 INFO mlflow.projects: === Run (ID 'b8a245a791ba47d6b
7487463a88e1f60') succeeded ===

```

## Inspecting stored models

The trained models are stored in `mlruns/0`.

These directories contain artifacts and config that is needed to serve them.

```
In [37]: import mlflow
```

```
In [38]: mlflow.__version__
```

```
Out[38]: '2.0.1'
```

## Serving model

Now that we trained our models we can go to *Models* page on MLFlow UI (<http://localhost:5000/#/models>).

Click *sklearn\_knn* on this page, choose a model and move it to *Production* stage.

The following cell will serve the model at localhost on port 5001.

```
In [39]: %%bash --bg
source mlflow_env_vars.sh
mlflow --version
mlflow models serve -m models:/sklearn_knn/Production -p 5001 --env-manag
```

## Prediction

We'll load data that we can feed into prediction server.

Let's predict for first winter day and first non-winter day (first rows of previous two dataframes)

**warning: this might fail at first because the prediction server didn't spin up; in this case wait a minute**

```
In [40]: %%bash
data='[[12.93,2.81,2.70,21.0,96.0,1.54,0.50,0.53,0.75,4.600000,0.77,2.31,
echo $data
```

```
curl -d '{"inputs\": $data}' -H 'Content-Type: application/json' 127.0.0.1:5000/predict
[[12.93,2.81,2.70,21.0,96.0,1.54,0.50,0.53,0.75,4.600000,0.77,2.31,600.0]]
```

% Total Current	% Received	% Xferd	Average Speed	Dload	Upload	Time Total	Time Spent	Time Left
Speed								
100	106	100	20	100	86	8343	35878	--:--:-- --:--:-- --:--:--
53000								

```
{"predictions": [0]}
```

Voila! We see that the model outputs correct predictions.