ASSIGNMENT 2

Joaquín Velarde Noval, a24joave, 20010425-T533

MLFLOW

In this document, we will describe how MLFlow has been used in the developing of a Neural Network model. Also, we will detail how the MLFlow setup that we will show is accesible from the reader's own computer.

To access MLFLow from computer:

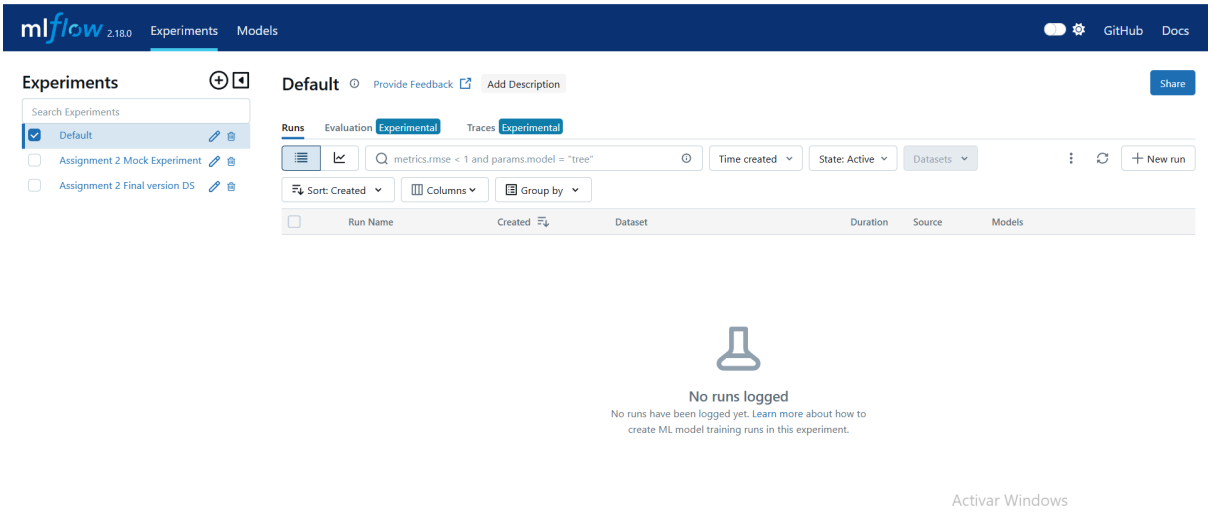
The reader must access in their terminal the folder in which he or she opened this document. In this folder there should be other directories, such as mlruns or mlartifacts.
Once in the folder, the user must type:

```
python -m mlflow server --host 127.0.0.1 --port 8080
```

This will activate a server in the user's computer containing all the MLFlow information that has been used. The user can access it just by typing http://127.0.0.1:8080 in his computer.

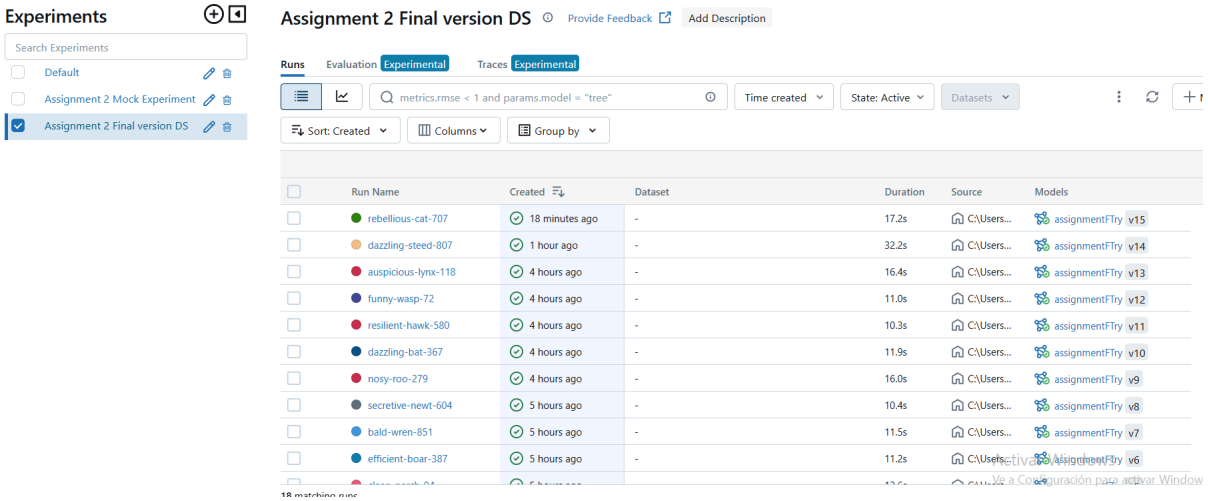
MLFlow setup:

Once in MLFlow, the user will see something like this:



Under "Experiments" the user will find a number of experiments. Each experiment, at least in our framework, is dedicated to a separate problem. The experiment that is important to us

now should be the third one, “Assignment 2 Final version DS”. If we click on it, we will see the following:



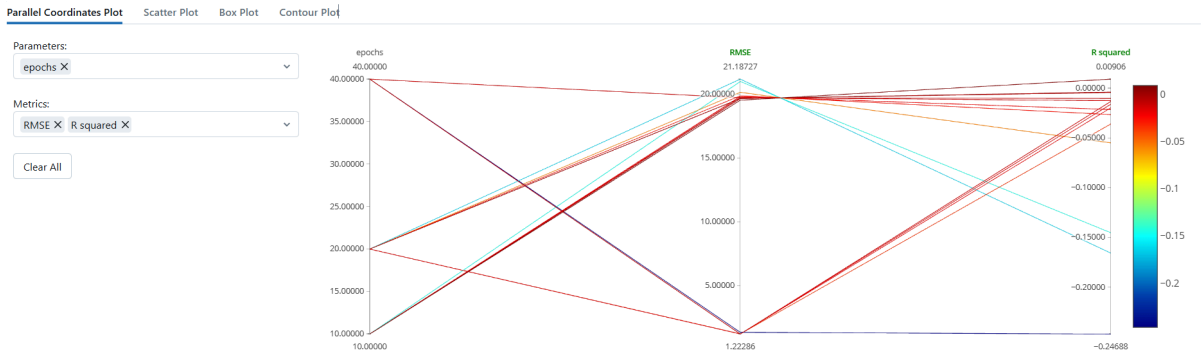
We have different versions of our model. Each version has different parameters, and, as a consequence, different performance. If we click on a run, we will be able to see both.

We are listing the following parameters:

Parameter	Value
1st layer	64
batch_size	64
dropout	auto
dropout rate	auto
epochs	20
L1 L2 regukarization	auto
learning rate	auto
loss function	mean_squared_error
n hidden layers	1
preprocessing	Robust Scaling + log transf+dropping outliers
random_state	42
weight inizialization	auto

And the following metrics:

Metric	Value
explained variance	-0.0052139588722055485
MSE	1.5048044993171978
R squared	-0.01968976628610064
RMSE	1.2267047319209288



A useful tool is the compare runs method, in which we can see what parameters performed better.

FLASK DEPLOYMENT

For deployment, we chose to build a flask API.
The API code can be found, again, in the unzipped zip folder. To run the API we need to follow the next steps:

Our MLFlow terminal needs to be running on http://127.0.0.1:8080 . Open a new terminal and navigate to our unzipped folder. Type the following command:

```
waitress-serve --host 0.0.0.0 --port 5000 depModelAss:app
```

A web server will start running then on the user's terminal. This server supports our API.

Now to use the API, in a PowerShell terminal we will have to write the following:

```
Invoke-WebRequest -Uri "http://192.168.137.1:5000/predict" ` -Method POST ` -Headers @{ "Content-Type" = "application/json" } ` -Body '{"features": [[7, 4, "oct", "tue", 90.6, 35.4, 669.1, 6.7, 18.0, 33, 0.9, 0.0]]}'
```

In "features" we have to put a list of lists, in which each row is one of the inputs we want a prediction for. The inputs have to be in the format showed above.

The API will return something like the following:

```
StatusCode        : 200
StatusDescription : OK
Content           : {"prediction":[[2.285720282305324]]}
```

Where [[2.285720282305324]] is the list of predictions.