Demo: container with PyCaret environment

Objective

Our goal is to run a Docker container to enable a simple machine learning project using the PyCaret package. Installing the package on a local system requires a lot of care - the official Docker image definitely makes this task easier.

For our purpose we will use:

- The official 'lightweight' Docker image of the Pycaret library available here
- A notebook with a sample project `binary classification - CHURN.ipynb`.
- Training data `customers_churn.csv`.

Checklist

- □ Running Jupyter notepad to load custom notepad and activate `pycaret` kernel.
- □ Loaded files:
  - □ `binary classification - CHURN.ipynb`.
  - □ `customers_churn.csv`.
- □ Successfully executed commands from notepad `Binary classification - CHURN.ipynb`.

Solution

1. run the official `light' docker image of the Pycaret library:

`docker run -p 8888:8888 pycaret/slim`.

2. go to the jupyter server address indicated on the screen (127.0.0.1:8888...).
3. import the `Customers binary - CHURN.ipynb` and `customers_churn.csv` files (*Upload* button in the top right corner). 4 Start notepad `Binary Classification - CHURN.ipynb` and run the individual cells.

Tips

In case the Jupyter Notebook you are running requires you to enter a password (or token), and the one indicated by Docker does not work, the problem is most likely another Jupyter server that was previously running. Solution:

1. check the currently running Jupyter servers:

   `jupyter notebook list`.
2. stop the previously running server: `jupyter notebook stop 8888`.

3. restart the image.

Sometimes the source of problems can also be previously running, and still active, containers. To check which ones are running, run: `docker run ps`.

To remove all running containers (note: this is an extreme solution), run:

`docker container prune`.