**Setting environment with Pycharm and Conda**

Ctrl-alt-s

Project – interpreter – select conda

Now we must install the dependencies or libraries

Tools -> Sync Python Requirements

**Way to install modules without pycharm**

pip list this command list all modules installed in the actual environment

pip freeze > requirements.txt this command get the list of modules in the actual environment and build a txt file

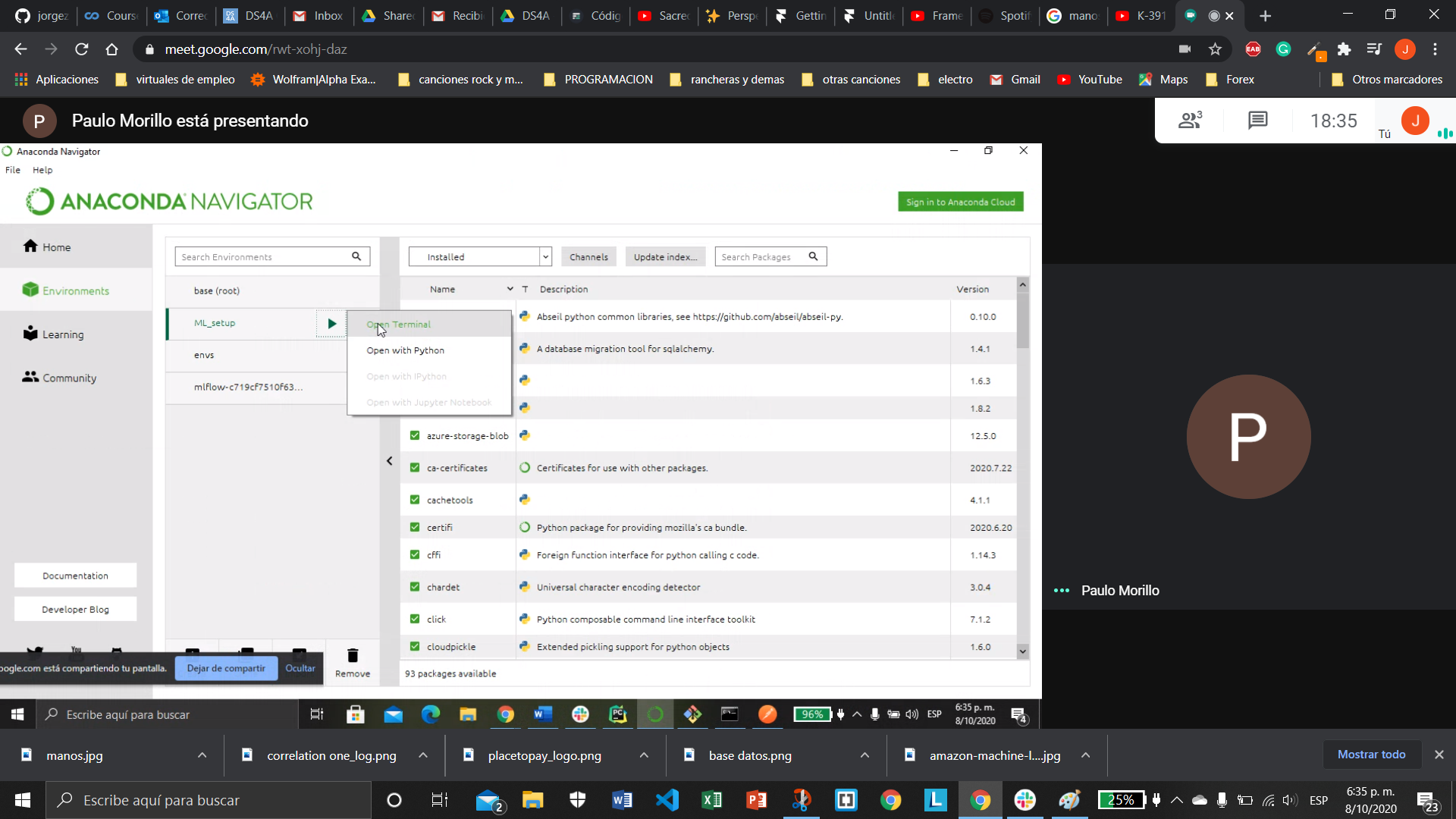
pip install -r requirements.txt way to install the modules in new environment

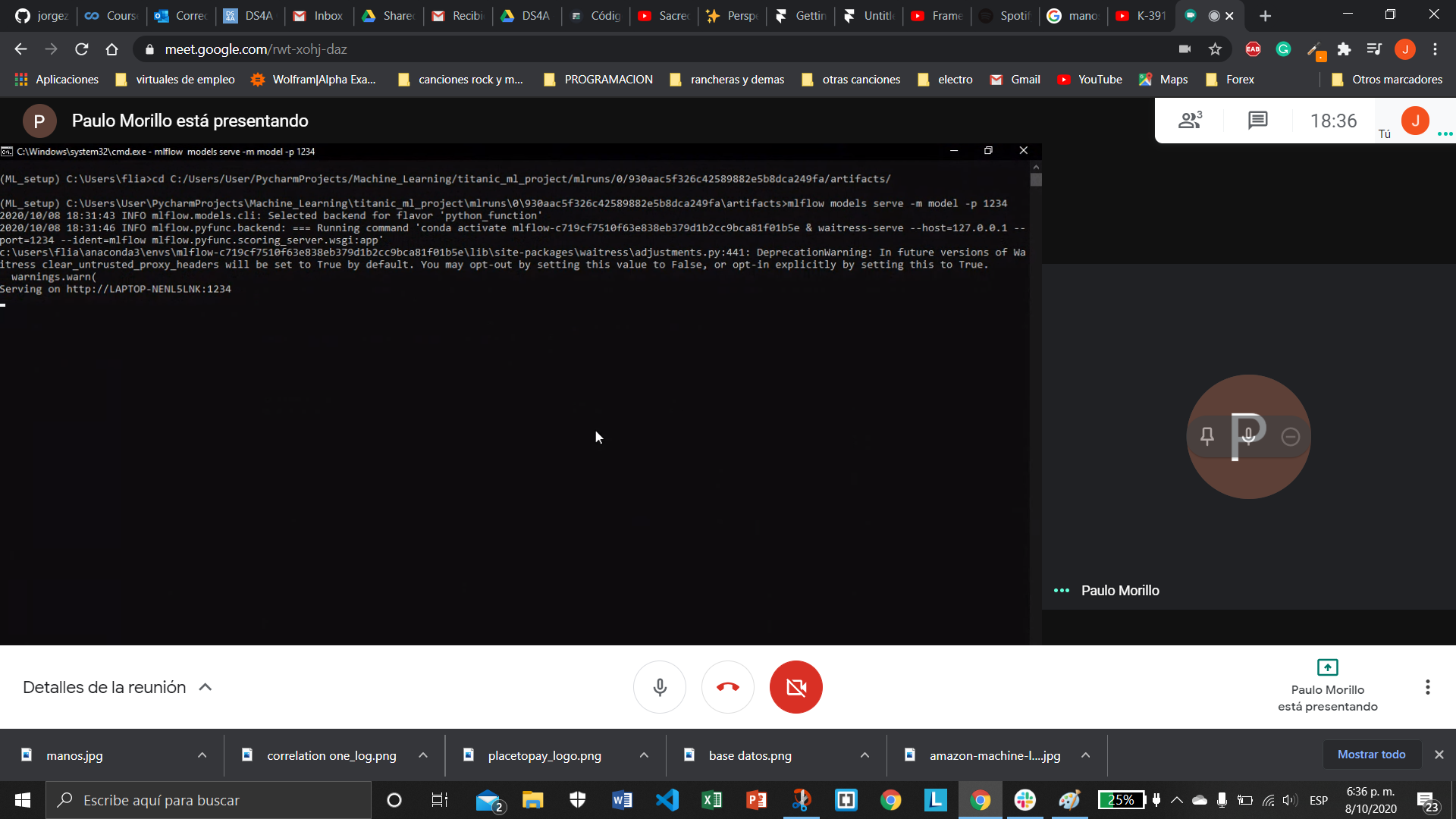
Possible mistakes:

Anaconda doesn’t run pycharm as administrador and if we are using OS module we are going to have troubles. To solve this, close pycharm and reopened as admin

Scikit-learn module couldn’t be installed directly with pip install, This is so weird but I remove the version and pip install the same version of the requirements file. (No reason about it)

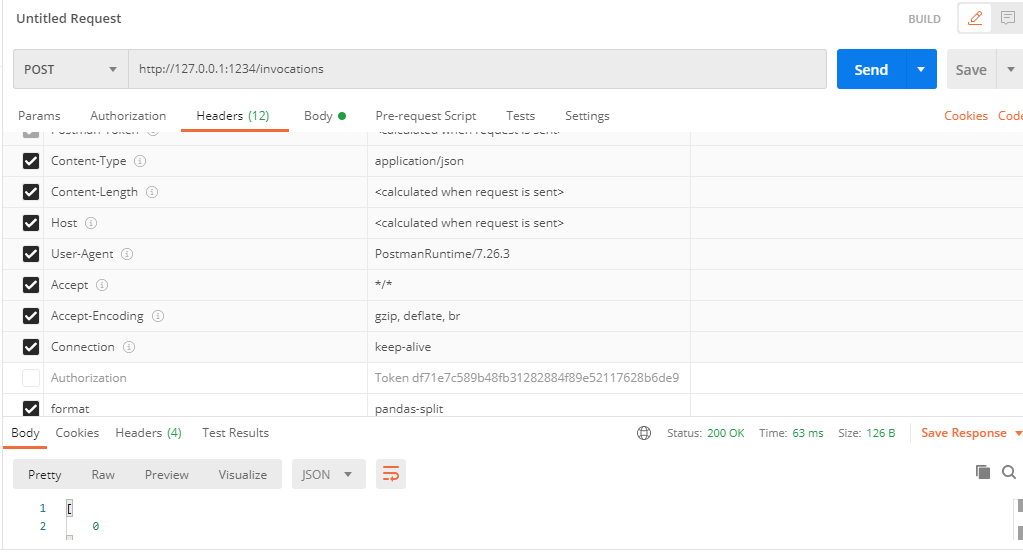
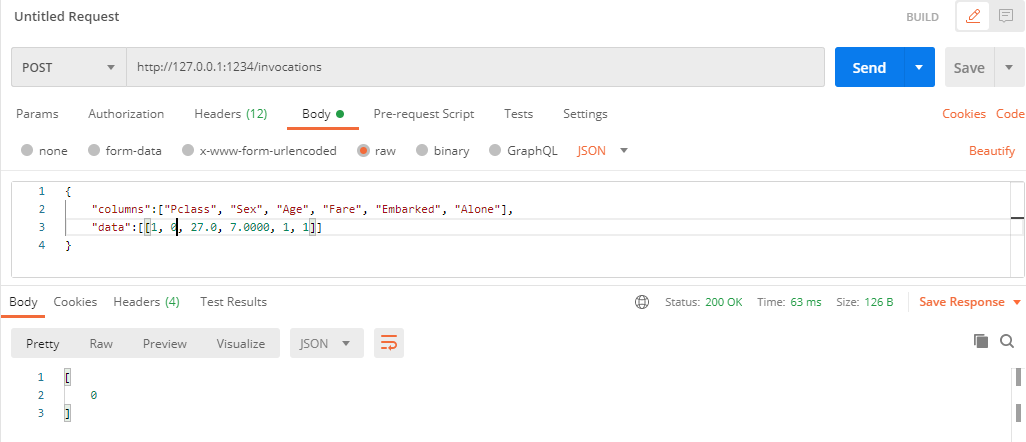
To serve mlflow for me was better opening the terminal with conda. And trying to reach the model folder with cd.





Pclass Sex Age Fare Embarked Survived Alone

curl -X POST -H "Content-Type:application/json; format=pandas-split" --data '{"columns":["Pclass", "Sex", "Age", "Fare", "Embarked", "Alone"],"data":[[3, 0, 27.0, 7.0000, 1, 1]]}' http://127.0.0.1:1234/invocations



Yaml is a superset of JSON

With pyyaml module

We can import functions, classes, dictionaries, list, variables, all that we import is going to be an object (for this reason we can have all data types). However, we should be load data with safe\_load to avoid security troubles.

Like a little overview for a yaml file the next example has a dictionary with item hola and a list as value

hola:

- mundo  
 - yaml

When we import that file, we are going to have {“hola”: [“mundo”, “yaml”]}

Installing typer

pip install typer[all]

Typer is a library which permits to get data from CLI. Instead sys.arg they only need to declare the arguments in the main function (as we did it with C)

For example this is the hello world:

def main(name: str, lastname: str, formal: bool = False):  
 *""" This is our main method """* print(name, lastname, formal)  
  
  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 typer.run(main)

$ ./main.py Paulo Morillo --formal

Paulo Morillo True

As you can see, we have 3 arguments name as a str, lastname as str too, and formal as a Boolean with a default value equal to False.

To run this program

