**Building a Machine Learning API v1.0**

**I – Introduction**

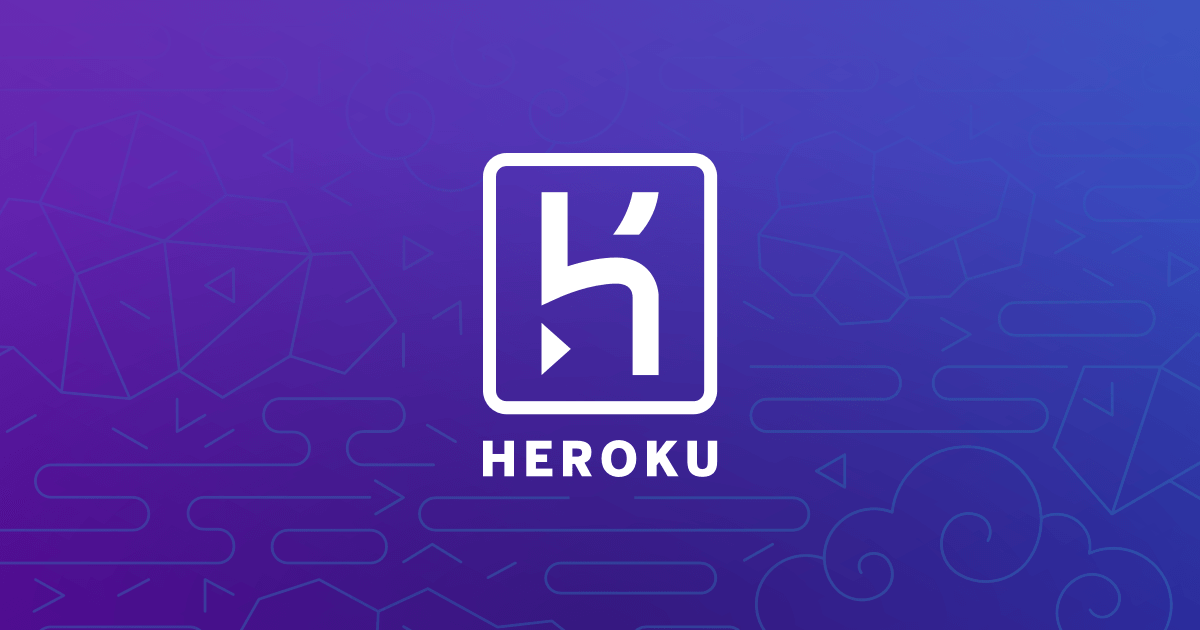
From this point forward we will be building the API using Flask. Flask is a framework to deploy web services and make actions over services.



For the API part we will be using FastAPI. FastAPI will bring us the flexibility to make a web application in an easy way.



Finally we will deploy it using a service called Heroku. Heroku will let us deploy applications on the web to people consume the application.



And for testing the API or backend we will be using PostMan. PostMan is a great tool and is the de-facto standard tool for testing web applications due to the versatility.



Now that we understood the devops tools, let’s break it down and start constructing the backend application.

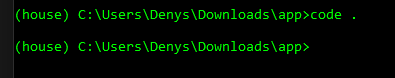
**II – Install Packages**

* [Download and Install PostMan](https://www.postman.com/downloads/) on your local machine and open an account.
* [Download, install and validate the steps to follow Heroku CLI](https://devcenter.heroku.com/articles/heroku-cli) on your machine and if necessary open an account.

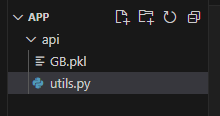
**III – Creation of the Backend App**

**III.1 – Visual Studio Code and Utils**

* Open VSCode or any editor you want over the environment using code .



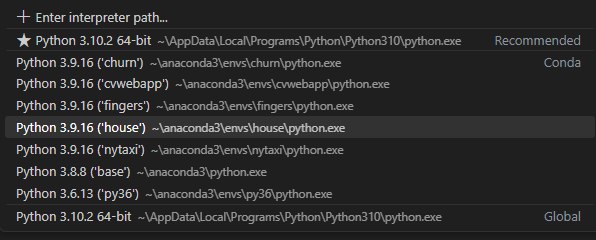
* When opened the folder try to copy the GB.pkl



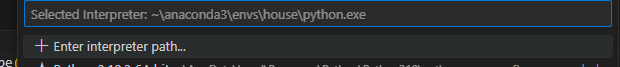
* If your interpreter is not selected press Ctrl+P and select interpreter



* Over all interpreters select the one that is correct with your project, my cas is ‘house’



* On the other hand, if your interpreter is not in the list, search for the folder (usually in ~\anaconda3\envs) and look for the python.exe in your enviroment

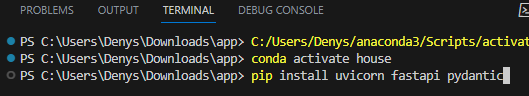


* Results must be shown below on the bottom right

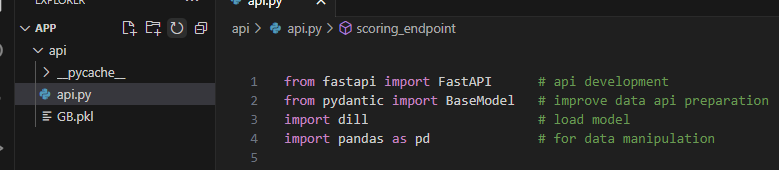


**III.2 – Visual Studio Code and Utils**

* On VSCode open a terminal (on the toolbar) and install the dependencies

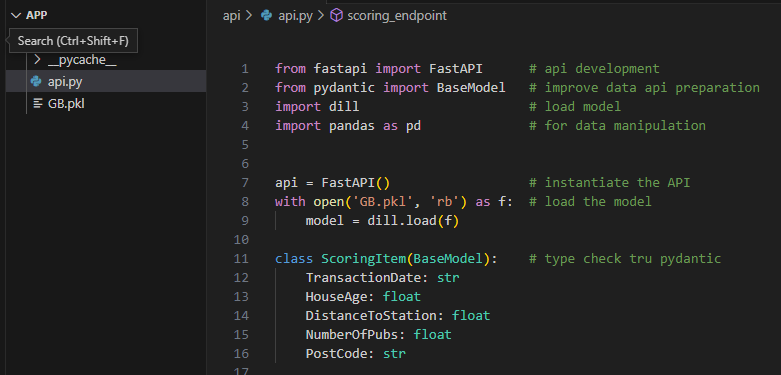


* Pydantic will help us on the preprocessing of data itself
* On the api folder create an api.py file and import the necessary libraries



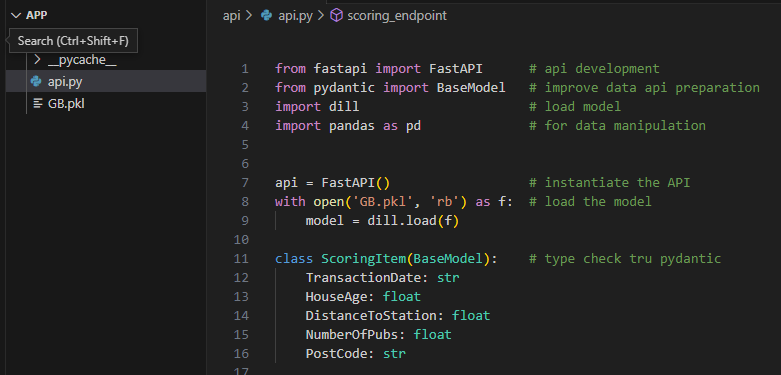
**III.3 – Create the API and Load the Trained Model**

* Next thing is to preparate the FastAPI interface and load the model



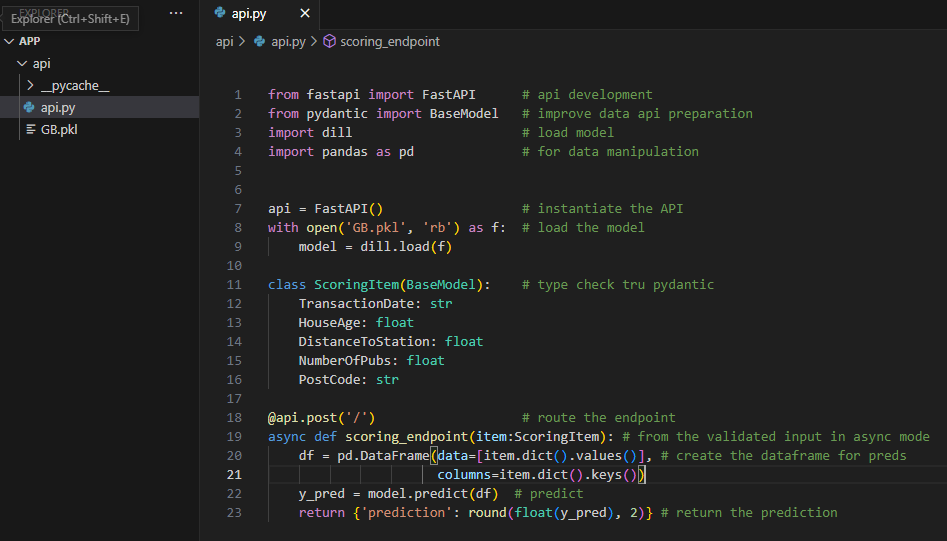
**III.3 – Validation Class**

* Next create a base class to consume the correct data using pydantic

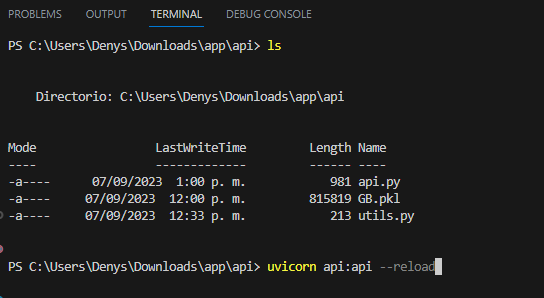


**III.4 – Develop Route Endpoint and Initial API Tests**

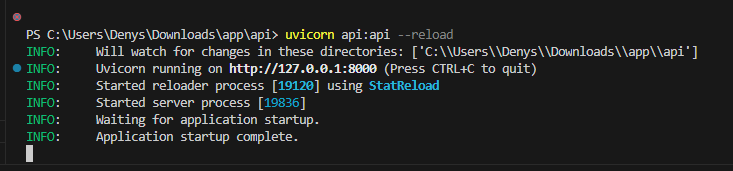
* Create the function to test the endpoint, we will use an async transaction and create the dataframe for finally make predictions and return the result



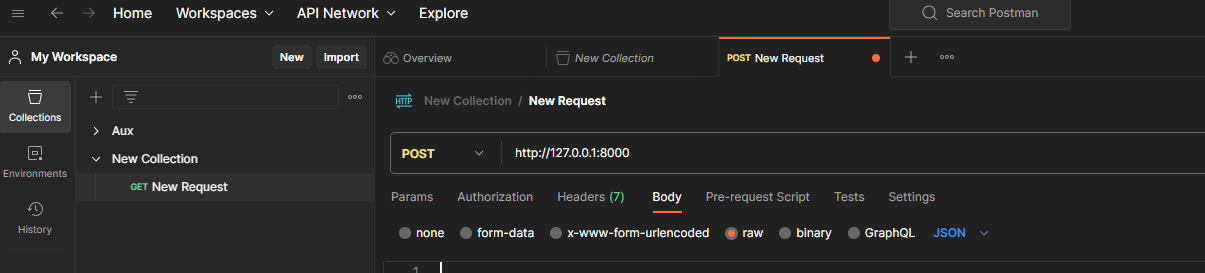
* Run vunicorn from the folder app/api executing the application via the terminal.



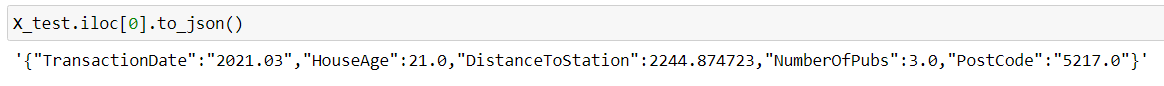
* Once completed successfully you must see something like this that the server has started.



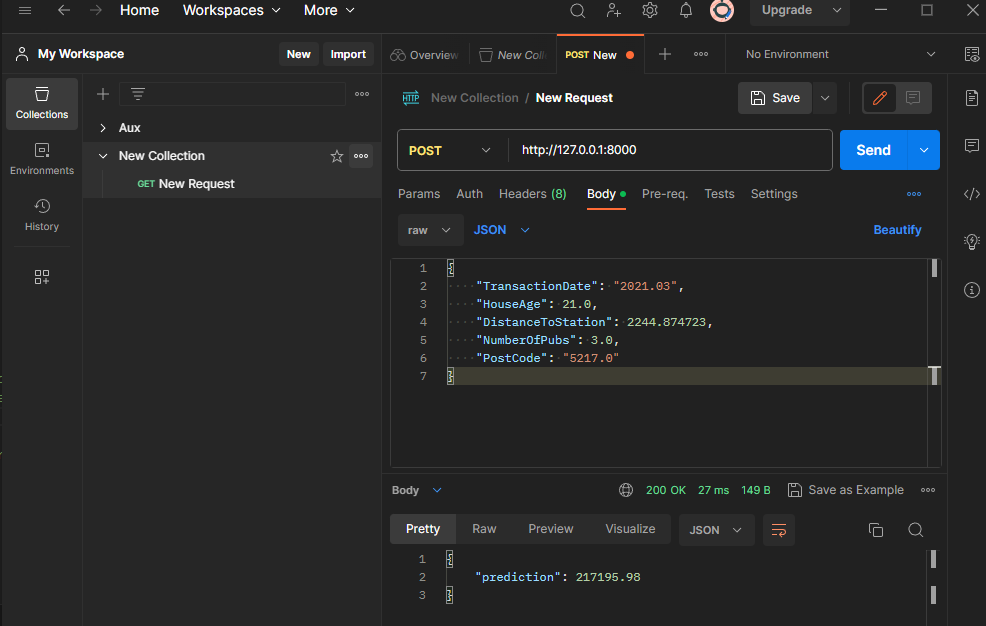
* For testing predictions open PostMan and copy the <http://127.0.0.1:8000> address and prepare the test data in PostMan using a POST method.



* Recover the dictionary of the notebook (copy that) and paste it to the postman window

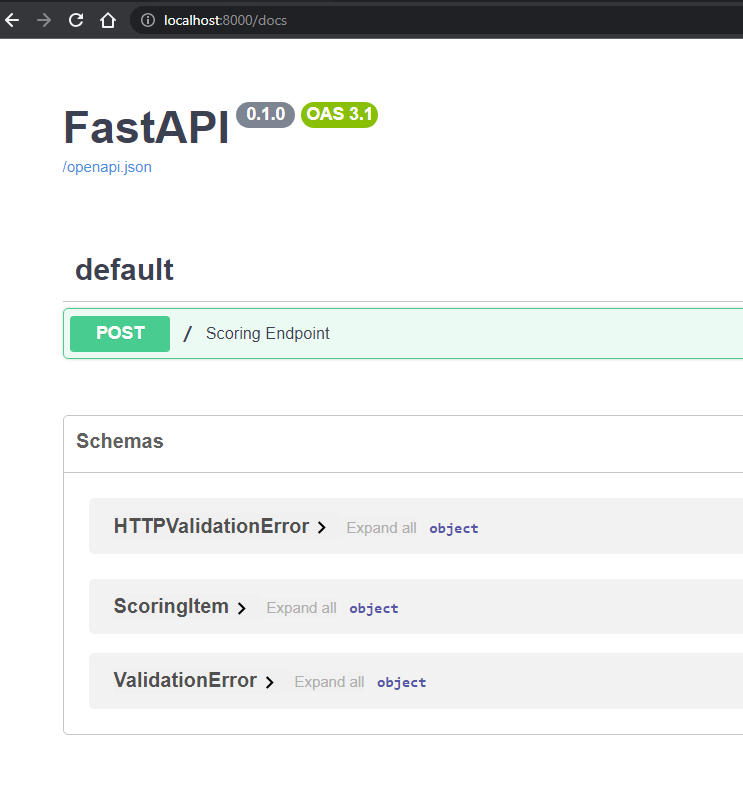
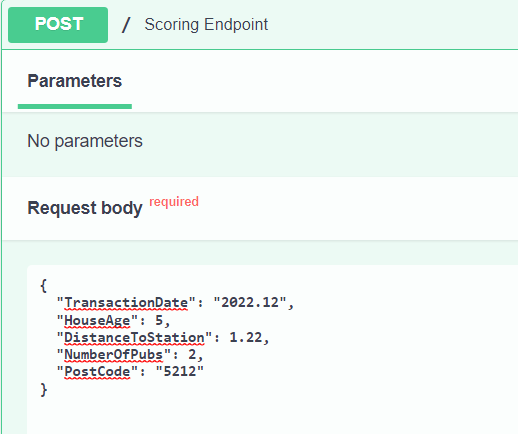
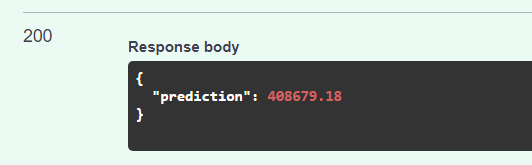


* Test the prediction



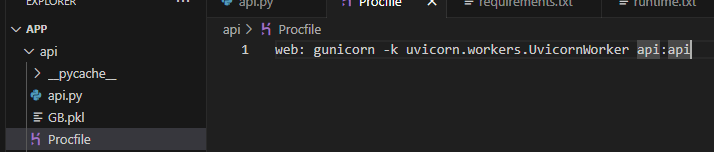
**III.5 – [OPTIONAL] – Look at the FastAPI Interface**

* You can also validate the data without PostMan and only FastAPI just looking at it at http:127.0.0.1:8000/docs, send information in the proper format and you will get a response.

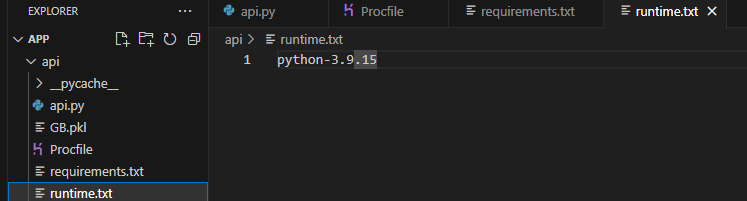
  

**III.6 – Deploy at Heroku**

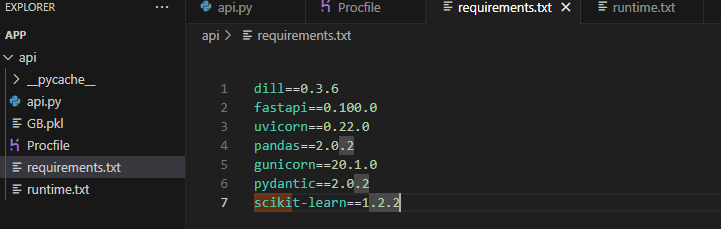
* Create a Procfile, runtime.txt and requirements.txt
* Define the Procfile contents



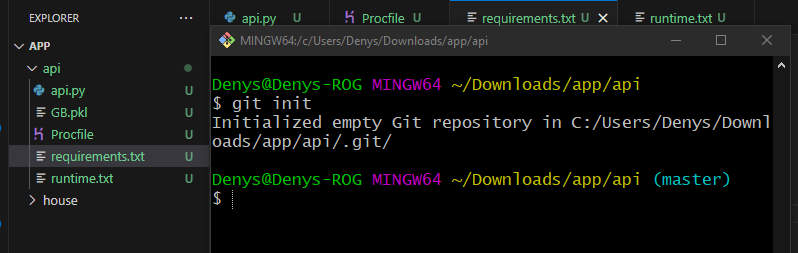
* Define Heroku python supported version



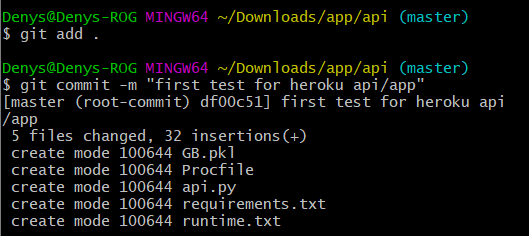
* Define the requirements file (you can see the dependencies on your environment using pip freeze on a terminal)



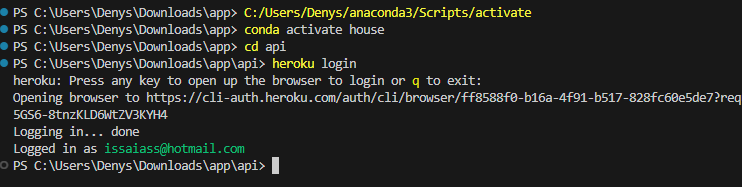
* Init the repository to apply de deployment using git init inside api folder (install git if not installed)



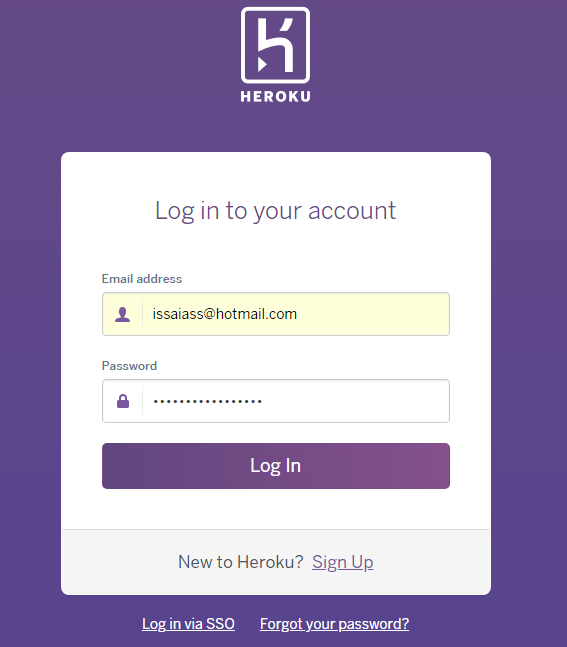
* Continue using git add . and git commit –m “some message”



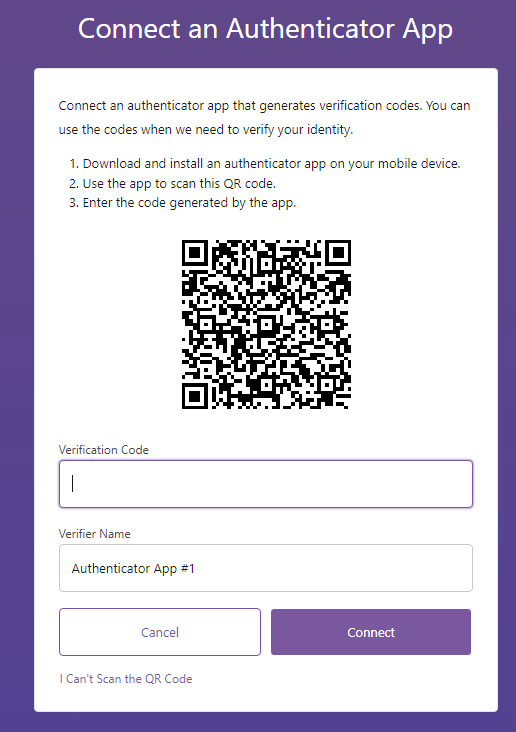
* Now login to the heroku cli and login



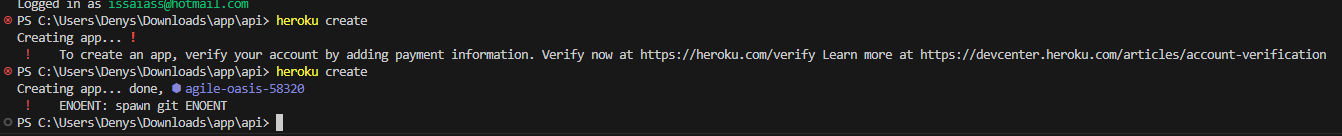
* Press Enter and login to heroku



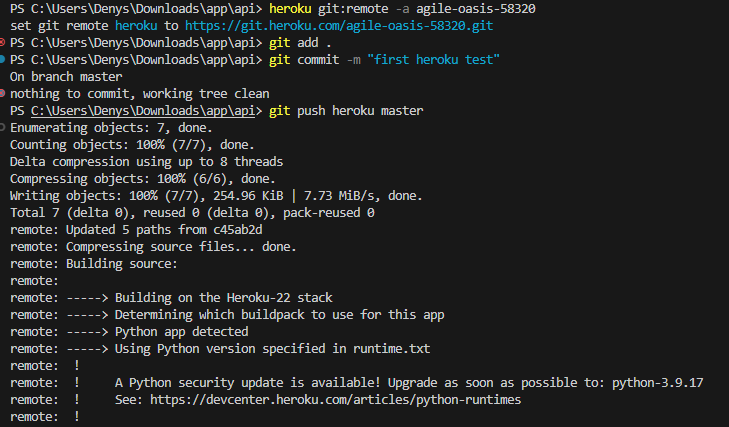
* Authenticate in any fashion way, I used the Google Authenticator, next close the window and return to the cli



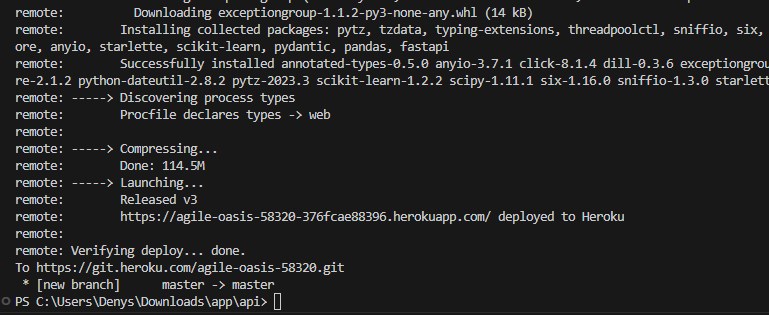
* Nowadays heroku charge you monthly if you use the system at low as $5/month. In my case I verified my CC (Credit Card) for creating a project next. I get agile-oasis-58320 as my app.



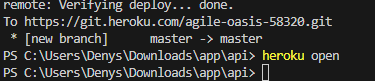
* Deploy as in the web says, that is…



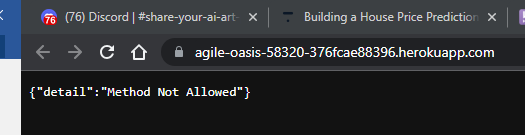
* When the deployment is finished and done you will see a window like this in the shell



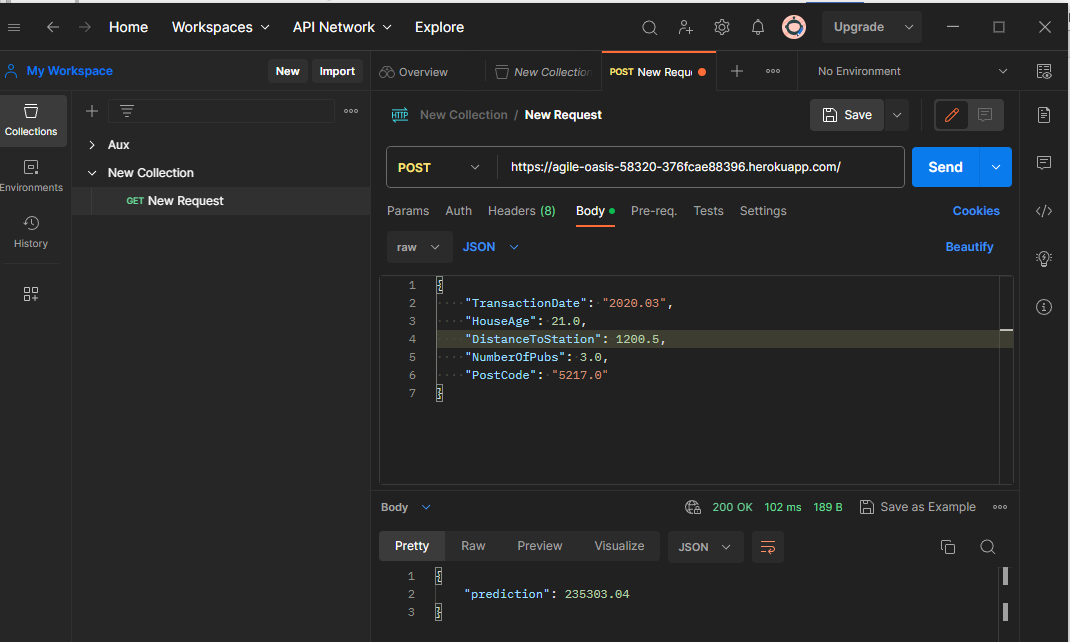
* Finally to see the application you could do on the terminal heroku open



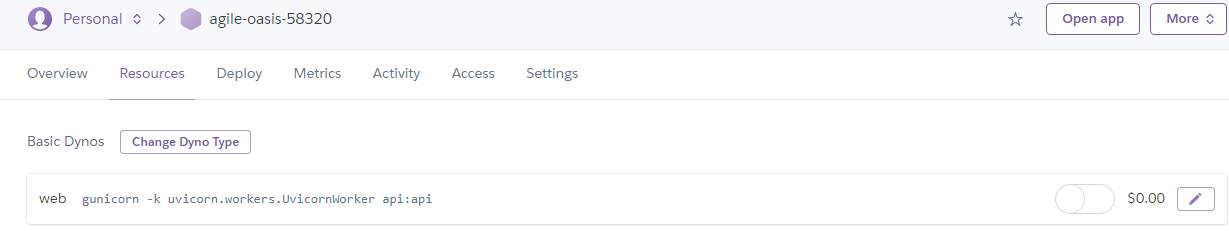
* At first you will see the method is not allowed, that is ok for now



* Now grab the link and test again in PostMan, you will have to see a response



* Finally remember to shutdown the instance and turn off the Dynos for not charge you.







You have made it so far. You did an end-to-end project for your machine learning applications, constructed a regression model, tested locally and finally deployed remotely.

