

Swagger:

Prévision Pathologies API 0.1.0 OAS 3.1

/openapi.json

default ^

POST /predict/linear Predict Linear

POST /predict/random_forest Predict Rf

POST /predict/gradient_boosting Predict Gb

POST /predict/cluster Predict Cluster

GET /health Health

```
{  
    "annee": 2027,  
    "region": 11,  
    "patho_niv1": "Cancers"  
}
```

server response

Code	Details
200	<p>Response body</p> <pre>{ "model": "Linear Regression", "log_Ntop": 12.342674518429163, "predicted_Ntop": 229273.3312062632 }</pre> <p>Response headers</p> <pre>Content-Type: application/json Content-Length: 123 Date: Mon, 12 Mar 2024 14:23:45 GMT Server: Apache/2.4.41 (Ubuntu)</pre>
200	<p>Response body</p> <pre>{ "model": "Random Forest", "log_Ntop": 16.04742402679356, "predicted_Ntop": 9317677.119300008 }</pre> <p>Response headers</p> <pre>Content-Type: application/json Content-Length: 123 Date: Mon, 12 Mar 2024 14:23:45 GMT Server: Apache/2.4.41 (Ubuntu)</pre>

```

Code Details
200 Response body
{
  "model": "Gradient Boosting",
  "log_Mtop": 15.807934548845632,
  "predicted_Mtop": 7343287.058874868
}
Download

Response headers
content-length: 94
content-type: application/json
date: Fri,27 Feb 2026 19:55:39 GMT
server: uvicorn

Responses
Code Details
200 Response body
{
  "model": "KMeans Clustering",
  "assigned_cluster": 0
}
Download

Response headers
content-length: 50
content-type: application/json
date: Sun,01 Mar 2026 12:40:33 GMT
server: uvicorn

```

La Régression Linéaire est complètement à l'écart (environ 229k vs environ 7-9M), ce qui est cohérent avec son R² de 0.156 elle ne sert à rien ici. RF et GB convergent vers le même ordre de grandeur (environ 7-9M), ce qui confirme leur fiabilité. L'écart entre RF et GB (environ 2M) est normal deux algorithmes différents, GB étant légèrement plus conservateur.

Docker:

The screenshot shows the Docker Desktop interface. At the top, there's a navigation bar with 'Local' selected, showing 'My Hub' and '8.32 GB / 24.77 GB in use 10 Images'. Below this is a search bar and a table with columns: Name, Tag, Image ID, Created, Size, and Actions. A single image named 'patho-api' is listed, with 'latest' as the tag, 'b68b1cfbd9e' as the image ID, '2 days ago' as the creation date, '1.78 GB' as the size, and a 'Docker' icon in the Actions column. Below the table, the Windows taskbar is visible with icons for File Explorer, Task View, Start, and Task Manager. The system tray shows battery level and signal strength. At the bottom, a terminal window displays the command 'C:\Users\ervin\Documents\data science python\projet\projet>docker run -p 8000:8000 patho-api' followed by several INFO log messages from Uvicorn indicating the server has started and is listening on port 8000.

```

C:\Users\ervin\Documents\data science python\projet\projet>docker run -p 8000:8000 patho-api
INFO:     Started server process [1]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

```