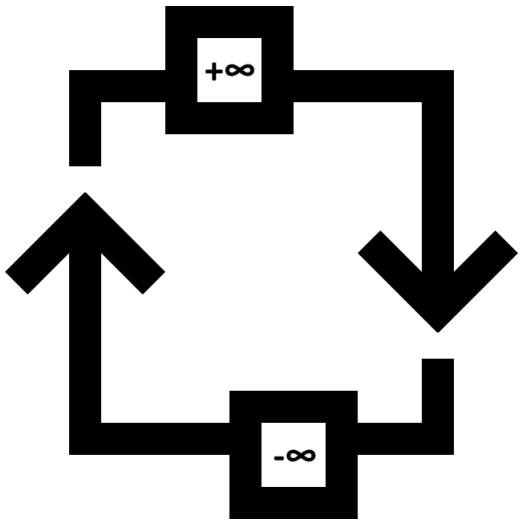
CONVERGENT ANALYTICS

HEALTH AI SUITE

VERSION: 1



TRACKING

EDITS

Edit	Date	Author(s)	Role(s)
Initial edit	2025-03-24	Sergey Lukyanchikov	Project Owner

APPROVALS

Approval	Date	Approver(s)	Signature(s)
Initial approval	2025-03-24	Sergey Lukyanchikov	NA

DISCLAIMER

This document provides information required for installation and use of Health AI Suite. The functionality of Health AI Suite is a template to be adjusted to the needs of the user. The user installs and applies Health AI Suite at their own discretion and risk.

TABLE OF CONTENTS

GENERAL

Health AI Suite is a bundle of open-source components configured to add AI-centric client-server computations to EMR/HER software. Health AI Suite is currently tested under the following components (a non-exhaustive list of the most important items):

- OpenEMR (openemr-7.0.2)
- XAMPP (xampp-portable-windows-x64-8.2.12-0-VS16-installer)

Prerequisites:

- AI Orchestration Platform with Stroke Prediction deployed

IMPORTANT: the above links are indicative, the components available under those links are not controlled by us. Download at your discretion and risk.

COMPONENTS INSTALLATION

Before downloading and configuring Health AI Suite, we must download and install open-source components on which Health AI Suite depends:

OPENEMR

Download and installation instructions: OpenEMR (openemr-7.0.2.zip)

XAMPP

Download: XAMPP (xampp-portable-windows-x64-8.2.12-0-VS16-installer)

Installation video: XAMPP

Check that the following additional (next to the ones required for XAMPP setup) PHP extensions are activated:

- cURL

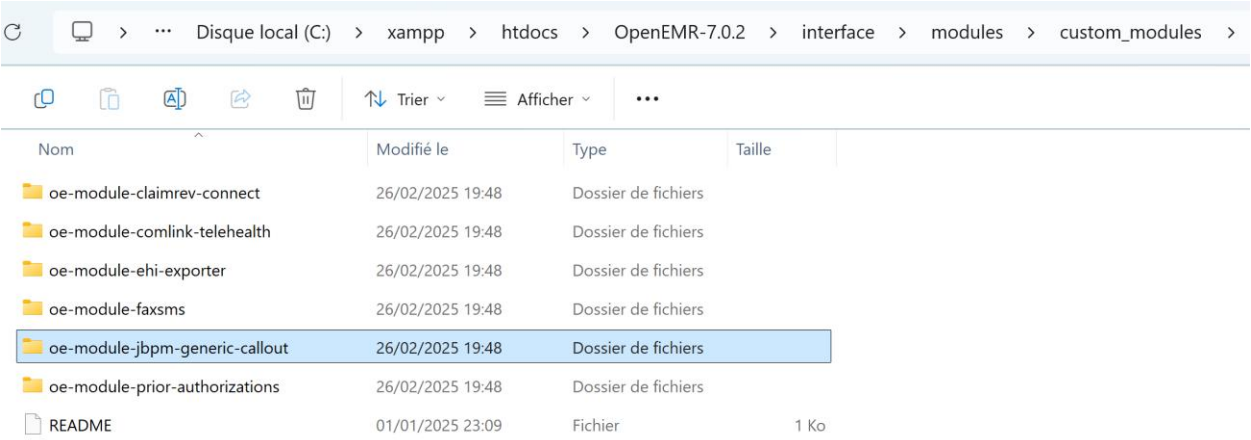
HEALTH AI SUITE DOWNLOAD, IMPORT AND SETUP

DOWNLOAD

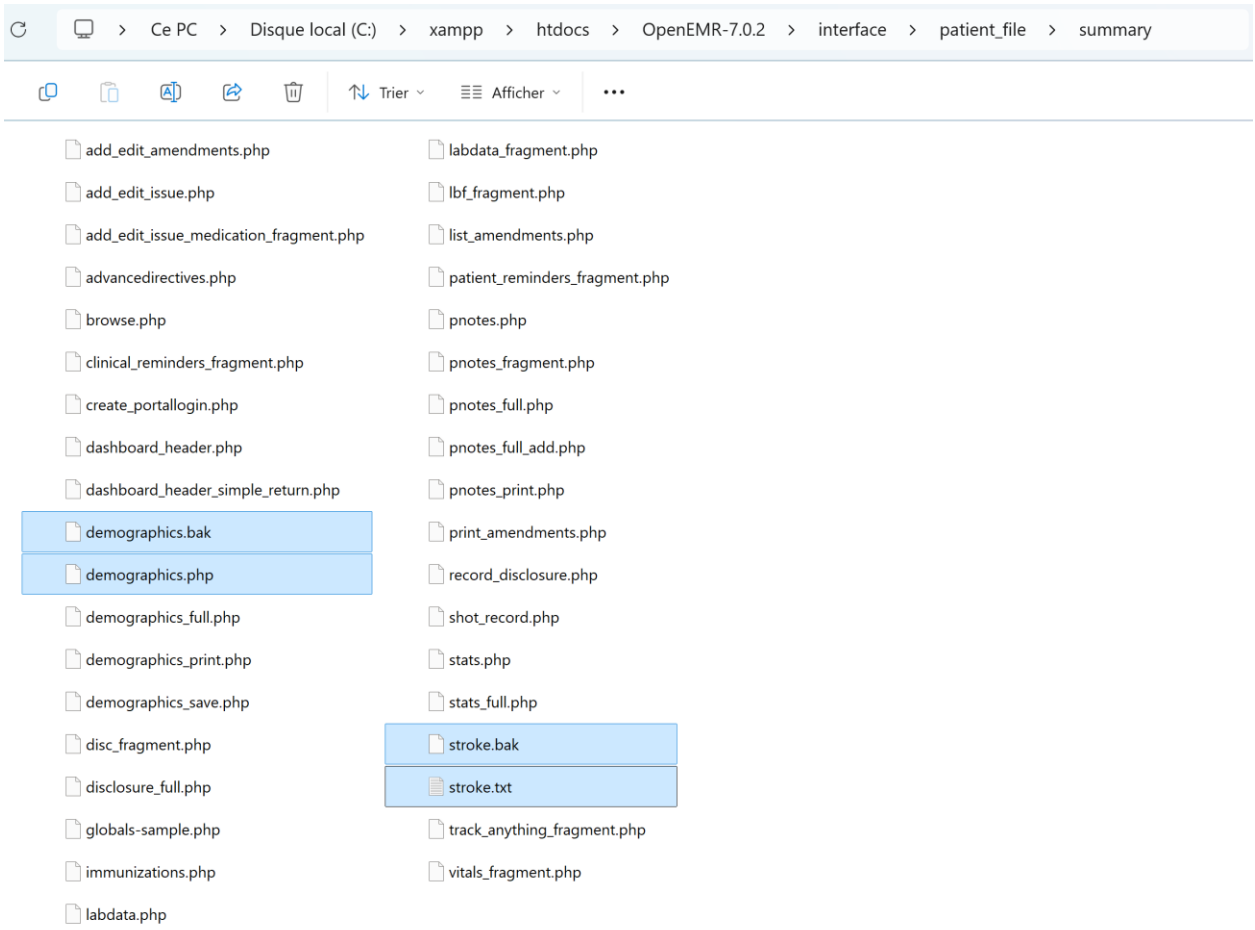
IMPORTANT: the below chapters describe a somewhat simplified procedure that is intended to get you up and running as quickly as possible. In particular, importing the downloaded classes in an IDE with subsequent version control setup should be considered for a more elaborate implementation.

We can now download Health AI Suite (Health_AI_Suite_v1.zip). Extract it from the zip file, and:

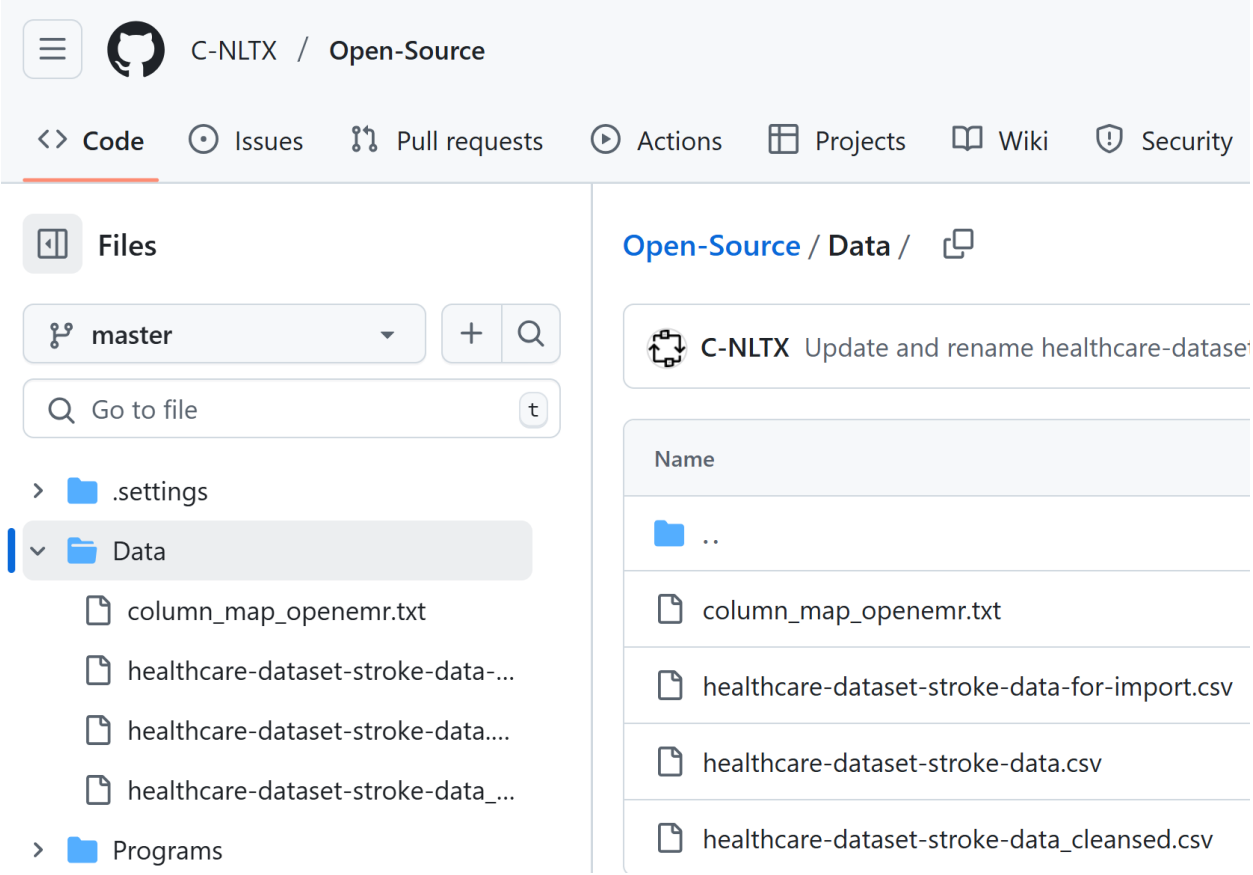
- move `oe-module-jbpm-generic-callout` folder from the download to your OpenEMR site's custom modules folder, e.g., the folders in `C:\xampp\htdocs\OpenEMR-7.0.2\interface\modules\custom_modules` path should look like below:



- overwrite (having previously backed up `demographics.php`) the files from the download's summary folder to its counterpart in your OpenEMR site, e.g., the files in `C:\xampp\htdocs\OpenEMR-7.0.2\interface\patient_file\summary` path should look like below:

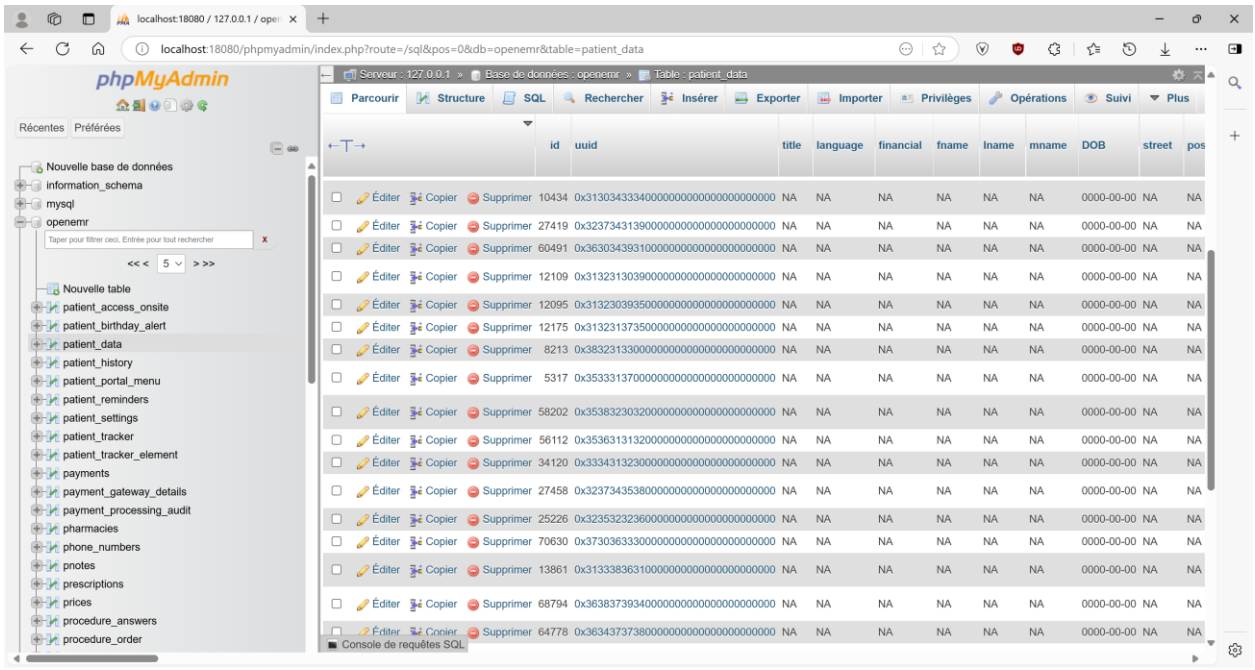


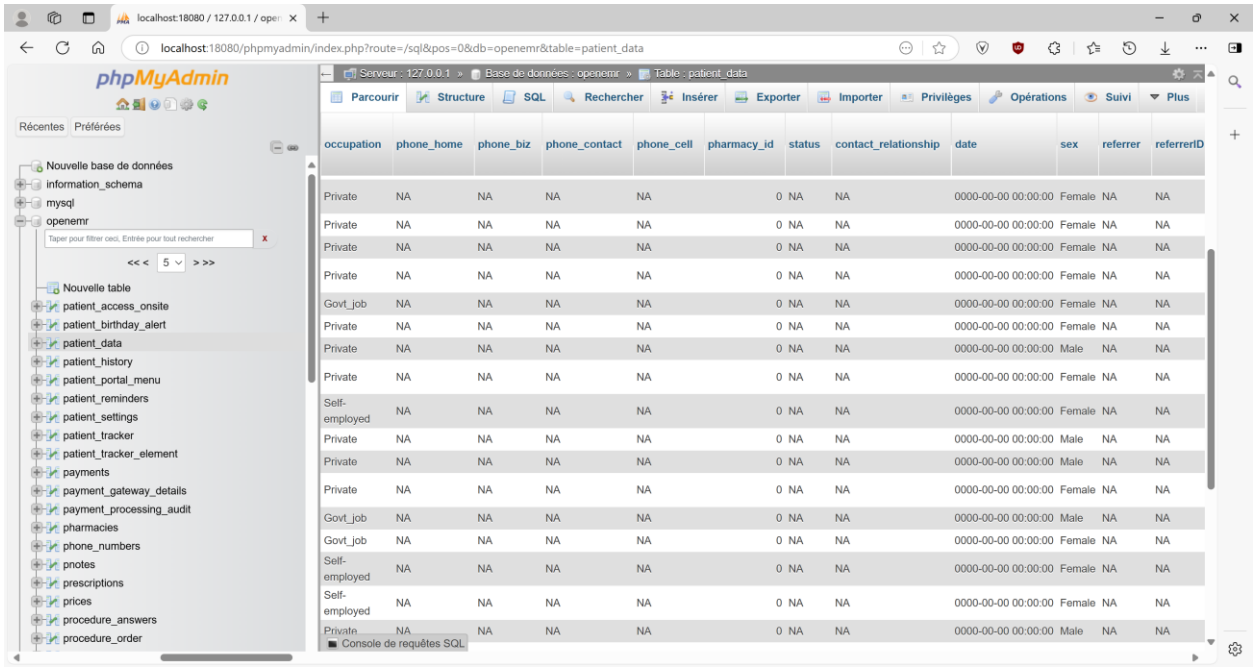
From Data folder of the same repository, download all the files to a local folder of your choice:



IMPORT

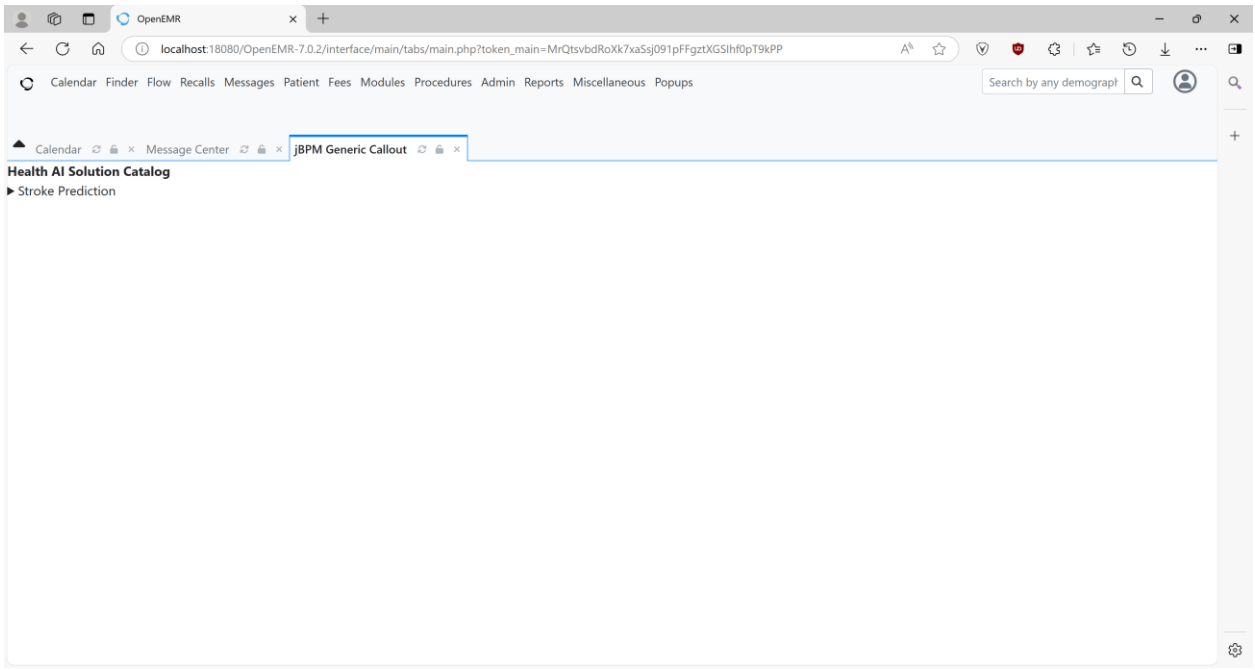
In phpMyAdmin, import in `openemr` database (or its analog in your OpenEMR site) in `patient_data` table the data from `healthcare-dataset-stroke-data-for-import.csv` file using for column mapping `column_map_openemr.txt` file, previously downloaded from Data folder of C-NLTX repository. The data in `patient_data` table will look like below:





SETUP

In OpenEMR, add and configure custom `oe-module-jbpm-generic-callout` module by watching the following tutorial. Once done, the added module's tab will look like below:

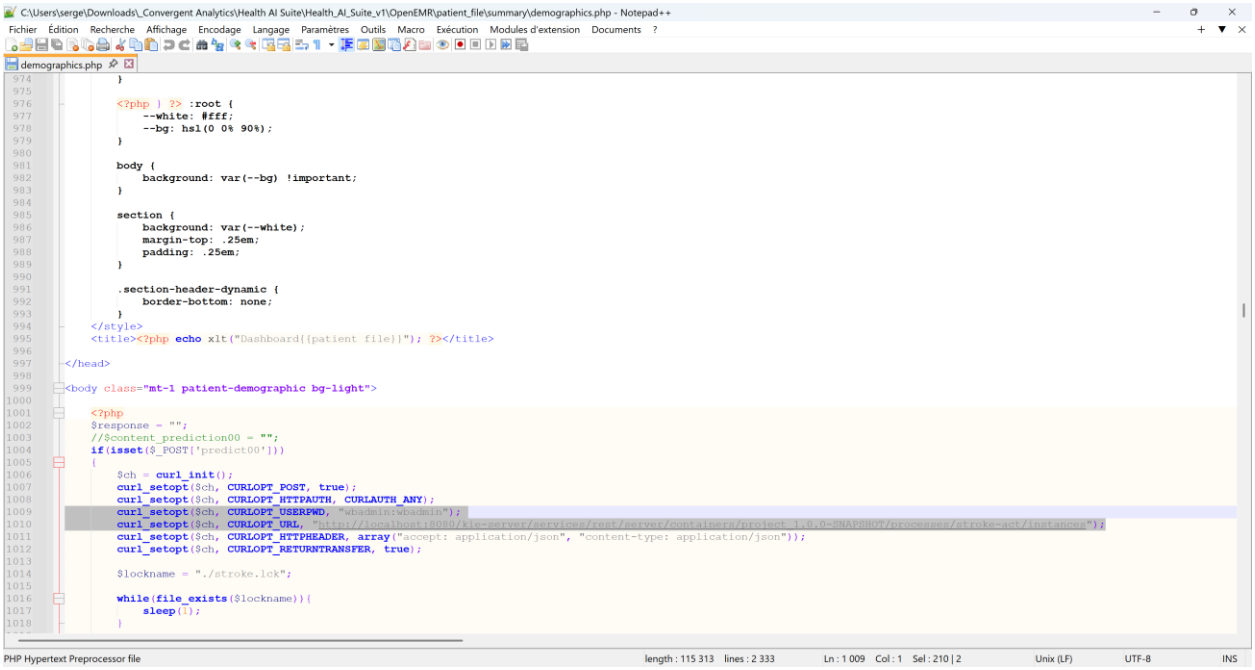


HEALTH AI SUITE CUSTOMIZING AND TESTING

CUSTOMIZING

Before we can test our Health AI Suite, let us customize the content we have just deployed. In `C:\xampp\htdocs\OpenEMR-7.0.2\interface\patient_file\summary\demographics.php`, adjust (if needed) the following lines:

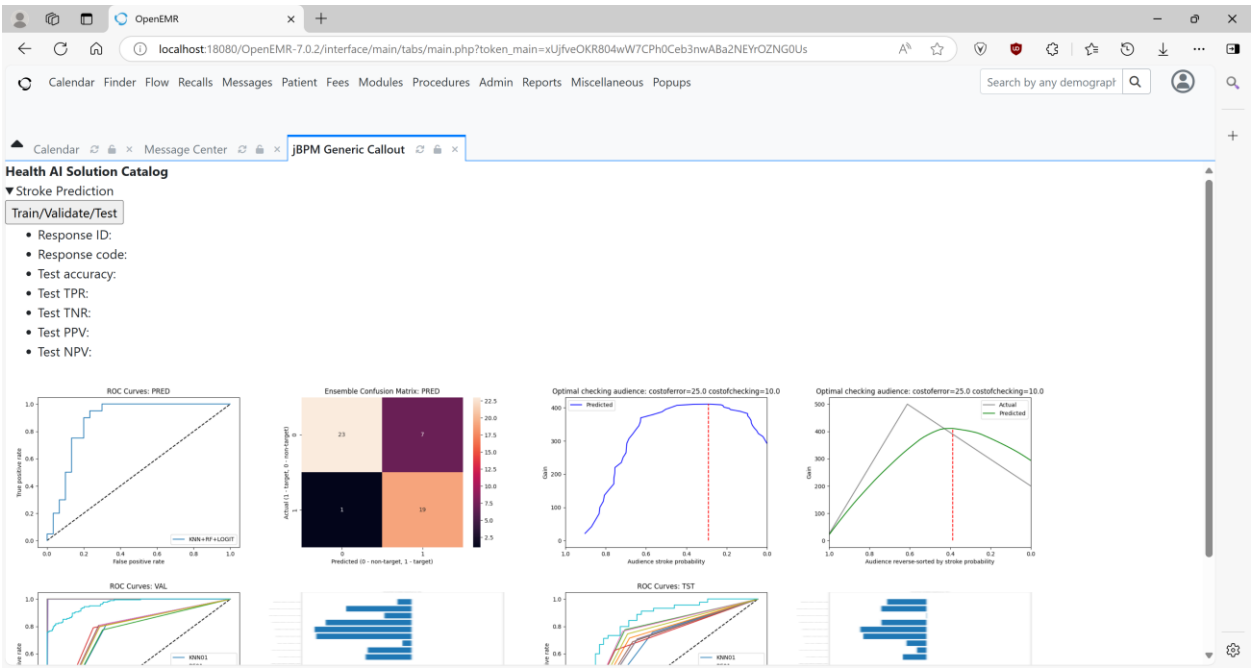
- 1009 – adjust the user password for a callout to AI Orchestration Platform
- 1010 – adjust the URL for a callout to `stroke-act` process deployed in AI Orchestration Platform
- 1079 – adjust the user password for a callout to AI Orchestration Platform
- 1080 – adjust the URL for triggering a query to a process instance in AI Orchestration Platform



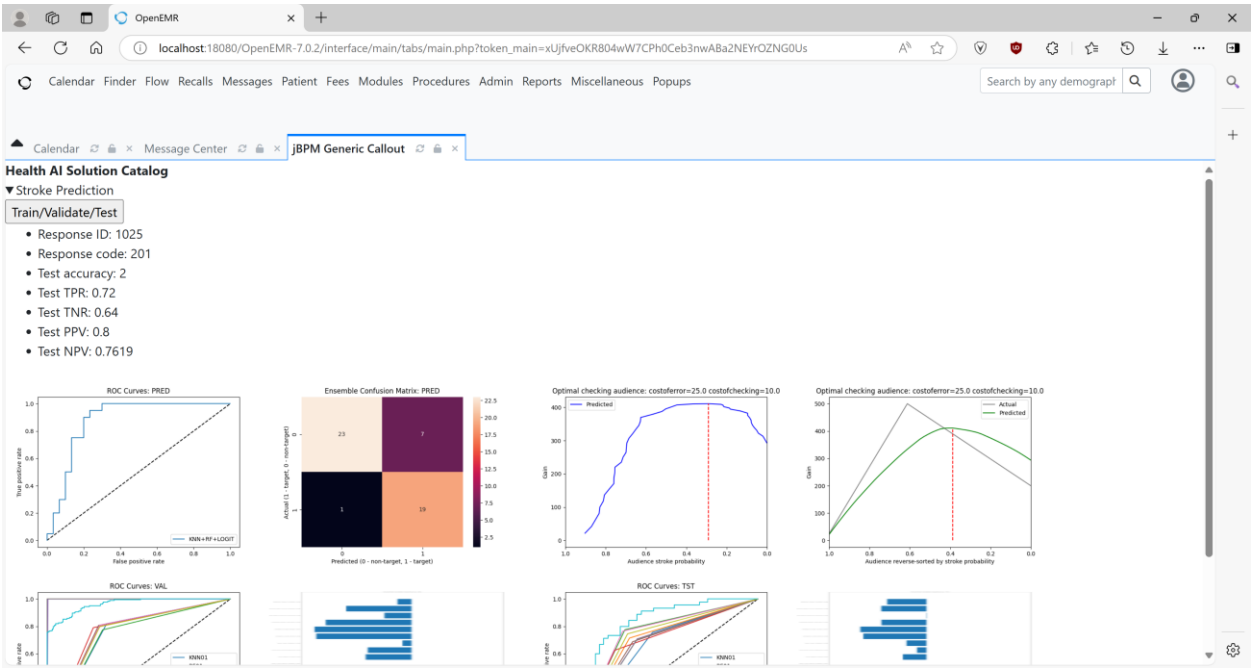
TESTING

We can now test our Health AI Suite by sending train/validate/test and predict requests from OpenEMR to Stroke Prediction solution that we assume implemented in AI Orchestration Platform.

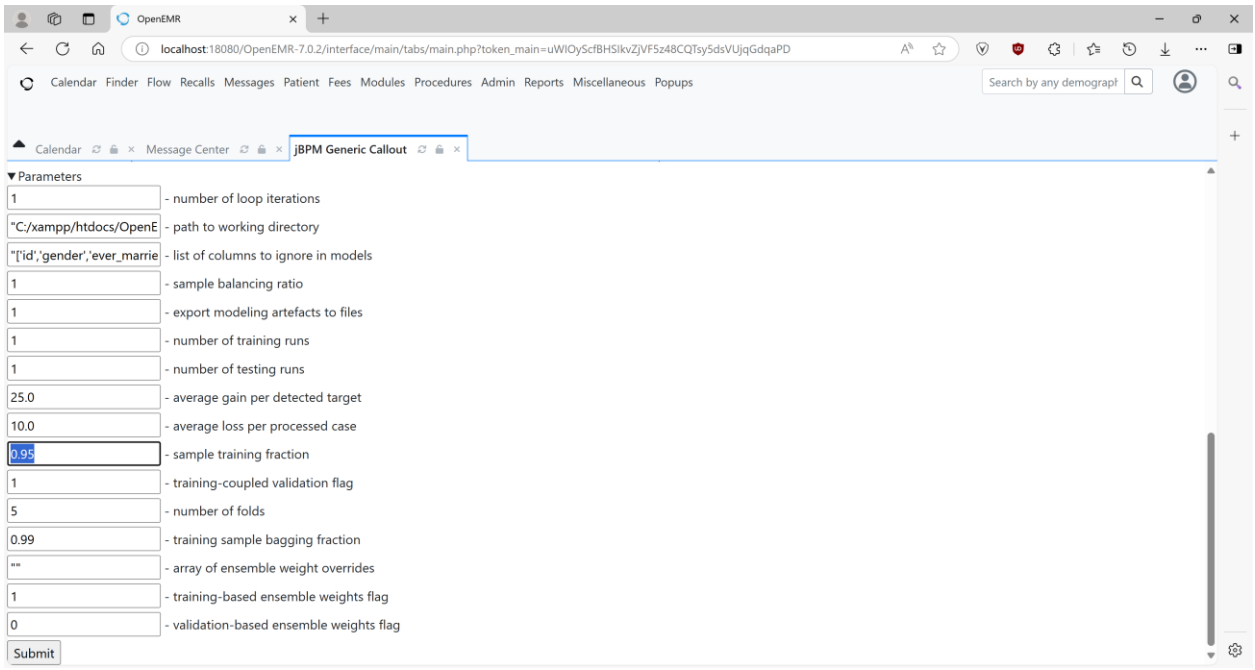
In OpenEMR, select from `Modules` menu `jBPM Generic Callout` item, and click on `Stroke Prediction` to gain access to Train/Validate/Test button (the graphs shown in the below screenshot will appear once the first callout was executed):



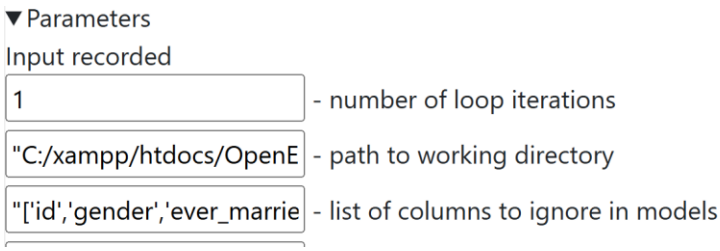
Press Train/Validate/Test button. Once the screen was refreshed (signals that the callout initiated by pressing the button is completed), click again on `Stroke Prediction` item to visualize the callout results (note the numbers that appeared next to the text labels under Train/Validate/Test button):



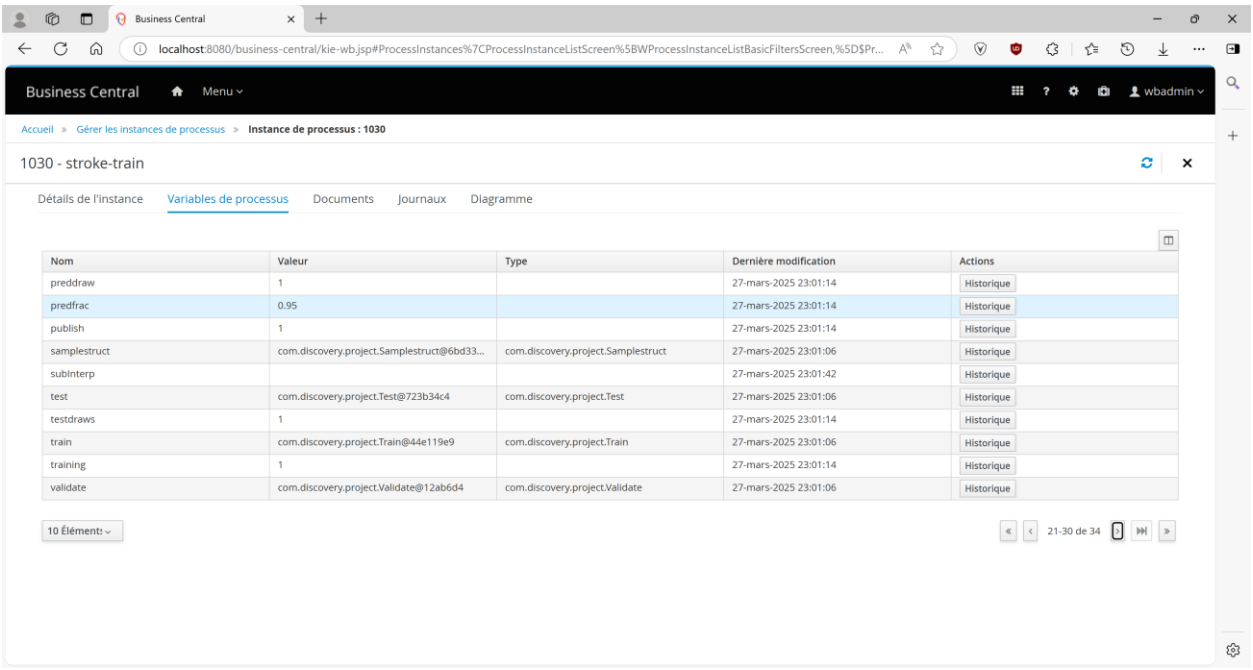
If you want to change train/validate/test hyperparameters, click on `Parameters`, modify the values, and press Submit button:



– once you reopen `Stroke Prediction` and then `Parameters`, you will see Input recorded message:

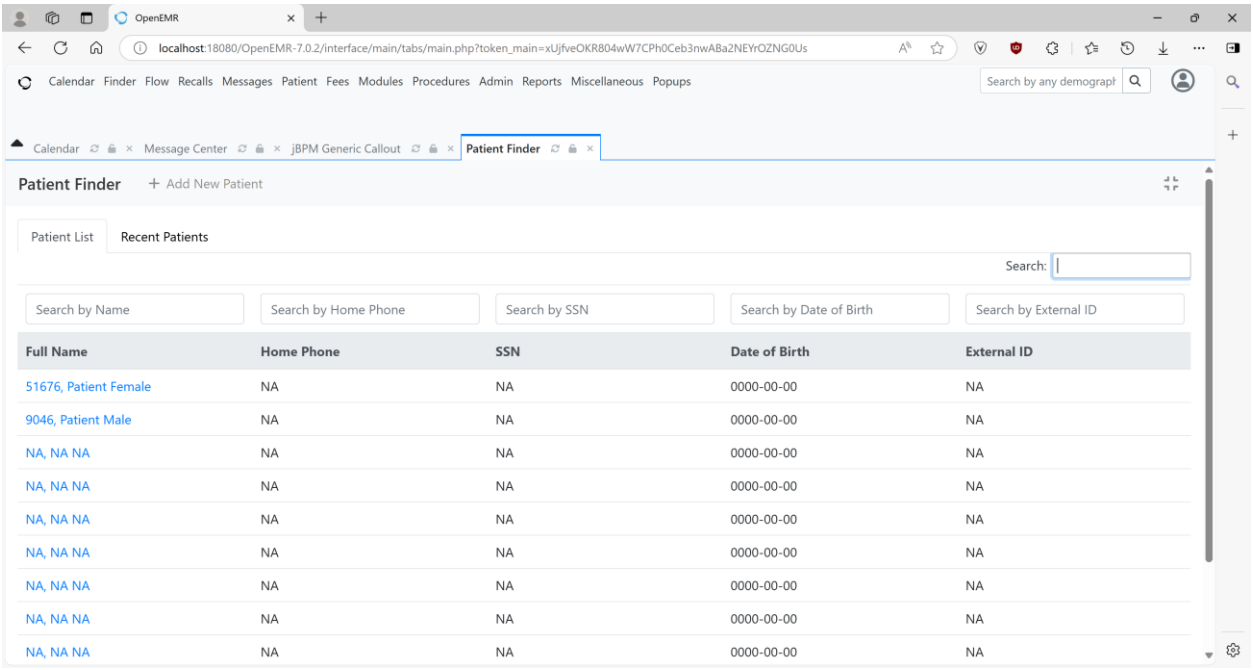


The modified values have now been saved to `C:\xampp\htdocs\OpenEMR-7.0.2\interface\modules\custom_modules\oe-module-jbpm-generic-callout\public\stroke.txt` file, and will be used by default for each subsequent train/validate/test callout to AI Orchestration Platform:

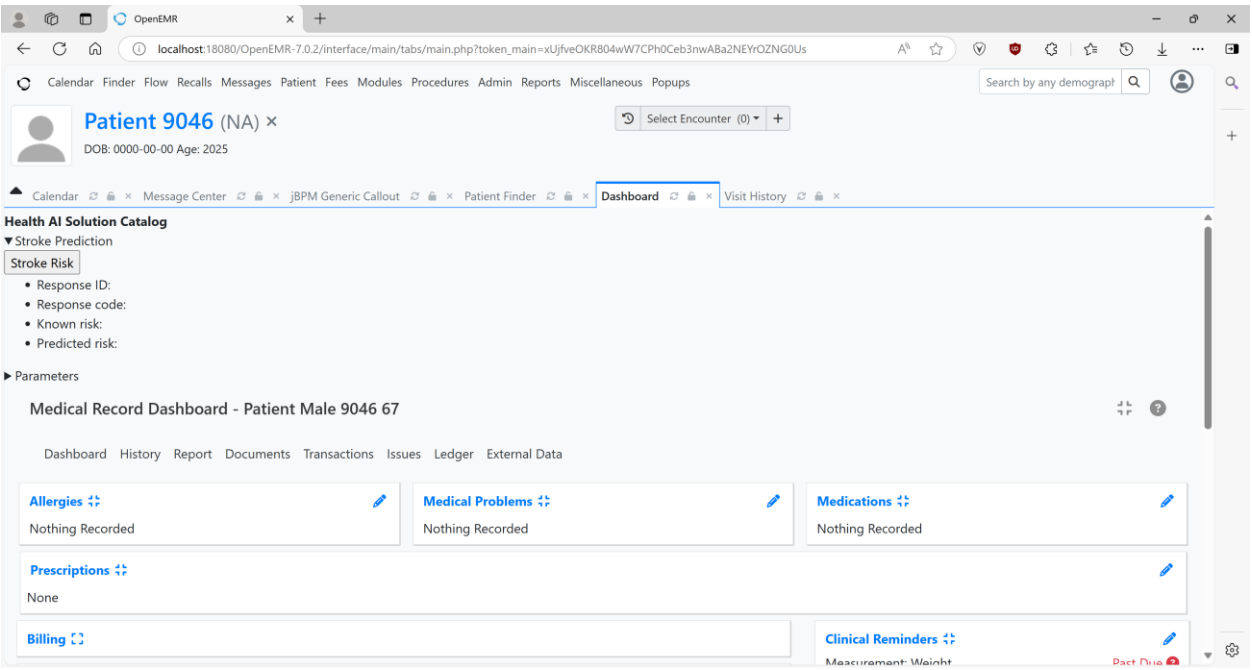


IMPORTANT: the saved parameter values can be overwritten in `stroke.txt` file by any user having access to train/validate/test callouts.

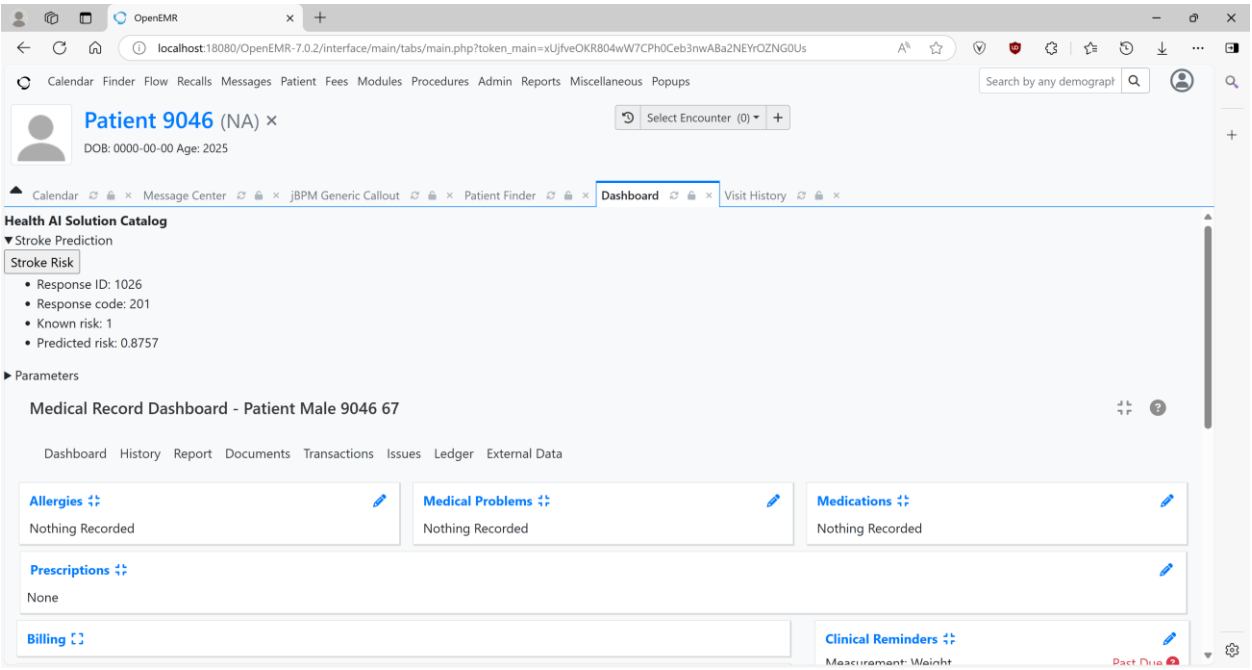
Proceed by selecting `Finder` menu item:



Click on any patient in the list, and then click on `Stroke Prediction`:



Press Stroke Risk button. Once the screen was refreshed (signals that the callout initiated by pressing the button is completed), click again on `Stroke Prediction` item to visualize the callout results (note the numbers that appeared next to the text labels under Stroke Risk button):



IMPORTANT: if testing by sending concurrent callouts (e.g., from separate OpenEMR users/computers), while a callout is processed by AI Orchestration Platform, a lock file `C:\xampp\htdocs\OpenEMR-7.0.2\interface\patient_file\summary\stroke.lck` is created making any other callouts wait till the current callout is fully processed. If for any reason `stroke.lck` is not deleted automatically and blocks callouts from execution, delete it manually.