Readme2

2026-01-18

InformaticaBiomedica – Final Project

Diabetes Prediction API – Reproducible Workflow

Important Notice

The trained model are **not included in this repository** because they exceed GitHub size limits.
As a result, the Dockerized API **cannot be built or executed immediately after cloning**.

To reproduce the project, the model must be **trained locally** and the resulting folder **manually copied into the `api/` directory** before creating the Docker containers.

This document describes the **exact required steps**, based only on the files present in the original repository.

Repository Files Used in This Workflow

The following files are required and already present in the original repository:

- `diabetes_012_health_indicators_BRFSS2015.csv` (dataset)
- `train_models.R` (model training)
- `plumber.R` (API definition)
- `run_api.R` (local API execution)

Additionally, these other two (and some of the previous files) are present in this repository: - `api/Dockerfile` (API container) - `docker-compose.yml` (container orchestration)

Step 1 – Clone the Repository

Download the repository from GitHub and work from the project root directory.

No files should be modified at this stage.

Step 2 – Train the Model Locally

Run the script `train_models.R`.

This script: - Loads the diabetes dataset included in the repository - Performs preprocessing - Trains the predictive model - Saves the trained model to disk as an output file inside the folder “artifacts”

This step is **mandatory**.
If the model is not trained, the API cannot function.

Step 3 – Copy the Model into the API Directory

Manually copy the trained model folder into the `api/` folder.

After this step, the `api/` directory must contain: - `plumber.R` - `Dockerfile` - artifacts folder

If the model file is missing from this directory, Docker execution will fail.

Step 4 – Run the API Locally

Before using Docker, the API can be tested locally by running `run_api.R`.

This step verifies that: - The model loads correctly - The API endpoints function as expected

Any error at this stage must be resolved before continuing.

Step 5 – Build and Run Docker Containers

Once the trained model is correctly placed inside the `api/` folder, Docker can be used to build and run the containers.

The Docker process: - Uses the pre-trained model already present in `api/` - Does not perform model training - Exposes the API according to `docker-compose.yml`

Common Causes of Failure

- The model was not trained
- The trained model file was not copied into `api/`
- The model filename does not match what `plumber.R` expects
- Attempting to build Docker before completing the training step

---

Final Remarks

This workflow ensures reproducibility while respecting repository size constraints.
Model training and deployment are intentionally separated and must be executed in the correct order.