Assignment 4: High Performance Computing Message Passing Interface (MPI)

In this exercise, we will implement the ML prediction service using a parallel distributed approach. To speed up the calculations, the calculation service has access to a number P of processors (which can be in the same machine or distributed among several machines). To set up this processing cluster, we will use the *Message Passing Interface* and its implementation OpenMPI. The service will satisfy the following requirements:

- The service will start by reading the pre-trained model `fraud_rf_model.pkl` that can be found in the examples repository.
- Then it will read as many requests from the `transactions` queue (that was implemented in Assignment 3) as processors available. If the queue is empty, the service will block until a message is available. If there are fewer than P requests in the queue, the service will proceed further with the available requests.
- For each request, it will pass a message to a worker processor to calculate the result of the final prediction.
- The service will gather the results of its workers to collect the final predictions, which will be sent to the `results` queue as individual requests.
- Now the service will take the next batch of messages from the queue.
- The number of processors will be configurable and default to 5.

We will see MPI examples in the exercises and sample code will be available in the `DS_Examples/mpi` folder.

Optional: Create a *docker-compose.yml* file that starts the service, creates a local Docker network and starts 5 worker nodes.

Deadline: 11/06/2025, 23:59 CET.

Additional requirements

- All files shall be submitted in a single `zip` file.
- A `README.MD` file will also be included in the submission with a short description of the submitted files.

Assessment

Total: 15 points.

- All requirements are satisfied: 10 points.
- The documentation is concise and technically correct: 5 points.
- **Optional exercise:** additional 5 points.