I-NERGY Load Forecasting Service – LightGBM Model

This is a forecasting service for predicting of the Portuguese aggregated electricity load time series (15-min resolution, 24hr forecasting horizon). This service is based on a LightGBM model implemented in the context of I-NERGY project. The model has been trained on the Portuguese timeseries from 2009 to 2018 and validated on year 2019. It can be used it to produce forecasts for periods from 2020 and later. No external variables have been considered. Please have in mind that the effects of the pandemic can negatively affect the model's performance.

Download & Deployment

- Use Deploy to local option.
- Extract the downloaded solution.
- Create a Kubernetes namespace. (Kubernetes Cluster shall be available) kubectl create ns
- Inside solution folder locate and run set up script. python .\kubernetes-client-script.py -n
- Verify the status of the created pods. kubectl get pods –n

Usage

In case this service is not combined with another service and therefore it is not executed by an orchestrator as a pipeline solution, a gRPC client shall be implemented. If used as a pipeline, it is recommended to utilize our i-nergy-load-forecasting-databroker service, which is also available on this platform. To facilitate this process a simple example client implementation is provided (see load_prediction_client.py in Documents section) along with relevant guidelines.

Users may provide the path to a timeseries in the form of a .csv file (for example series.csv), that has two columns, one of which must be named Load and the other should be named Date.

In order to execute the client:

- Install required dependencies (i.e., pandas, numpy, grcpio, logging)
- Generate the imported classes. (Classes are also available inside container and therefore can be copied, if users have access to container.)

    install grpcio-tools.

    locate file model.proto inside folder microservice

    create classes: python3 -m grpc_tools.protoc -I. --python_out=. --grpc_python_out=. model.proto

- Configure request's payload as described above.
- Replace values of host & port to the ones of the deployed service.
- Run client: python3 load_prediction_client.py