12-748-612

Date: 28.04.2022

Name: Nicola Crimi

Keywords: test coverage, refactoring prediction service, linkage db & training service

Overall Progress: 5

Activities completed:

- US12 - Base Training Dataset: Set up an end point to trigger the training service manually. Further enhanced the training service and resolved some bugs (the function shuffle_data was not working correctly). Massively improved the test coverage of the training service. This was necessary, since the training service is now connected to the mongo DB. We had to ensure that the most recent model (aka the model with the highest accuracy is fetched from the service). Managed to achieve a test coverage above 80% according to the Sonar Cloud tool. Implemented the flag on the DB which ensures that the data is only once used for training. Implemented Swagger.
- US4 - Update Model: Made sure that each time after the model is trained by the training service, it will be saved on the db. We restructured the prediction service in an identical fashion as the training service. At the same time, we ensured that the prediction service fetches the most recent model (checks for “time_created” on the DB) such that the customer is always interacting with the most accurate model.
- General: Refactoring of the prediction service to ensure better conformity with the training service. These two micro services are now structurally identical.

Current activities:

- Documentation of code.
- Update wiki.
- Analyse the 18 code smells (according to Sonar Cloud).
- Further enhancement test coverage of training service.

Goals or Activities for next deliverable:

- Finish the “performance analysis” task as early as possible which will be tackled at the very beginning of the upcoming sprint. The task will discover potential bottlenecks which could lead to major refactoring tasks.
- US8: Documentation of Training Service.
- Proper split of training & test data (if not already implemented).
- Analyse and fix the code smells (if necessary).

Immediate attention:

- Due to the wrong implementation of the shuffle_data function in the training service, we suspect that the model is massively overfitted. We are checking this issue in the upcoming days and discuss potential solutions if necessary.