AI Project

Find a team partner and define an idea for your AI project. You both will realize this project together by the end of the course. In principle, your project needs to scrape data from the Internet, prepare it for training and testing, realize an AI-based system and provide it via separate images, while realizing the following subgoals.

Subgoal 1: Git Usage

Fork the repository MarcusGrum / **AI-CPS** and modify this repository so that you provide the required code, data material and documentation to rebuild your own project. Please try to mime the current repository structure. As you collaborate to realize your project, label each of your commits by a powerful title. I expect at least three commits by each team member. Don't forget to update the ReadMe.md by your ownership and clarify, that your repository is part of the course 'M. Grum: Advanced AI-based Application Systems'. Push your final code, so that your project can be rebuilt by further researchers as well as by the course evaluator.

Subgoal 2: Data Scraping and Data Preparation

Scrape data from the Internet in accordance with your project idea and store them on your local file system. Clean your data (incl. algorithmic outlier dropping and algorithmic normalization) and store it as csv file called 'joint_data_collection.csv'. Split your data, so that you have one csv file called 'training_data.csv', that contains 80% of your joint dataset. The csv file called 'test_data.csv' should contain the rest of your dataset. Now, you have two disjunct data files. Finally, create a csv file called 'activation_data.csv' that contains one data entry of your test file.

Subgoal 3: Training, Testing and Application Material Docker Provision

Create the following two docker images, the first called 'learningBase_*yourIndividualProjekctTitle*' and the second called 'activationBase_*yourIndividualProjekctTitle*'. The first provides the 'training_data.csv' at the path called '/tmp/learningBase/train/' and the 'test_data.csv' at the path called '/tmp /learningBase/validation/'. The second image provides the 'activation_data.csv' at the path called '/tmp/activationBase/'. Further, each of your images should be based on a 'busybox' image and contain the following two: first, your corresponding data file, second, a Readme.md that indicates a) ownership, and clarifies, that your image was b) created as part of the course 'M. Grum: Advanced AI-based Application Systems' by the 'Junior Chair for Business Information Science, esp. AI-based Application Systems' at University of Potsdam, c) the origin of your data scraped and d) the commitment to the 'AGPL-3.0 license'.
Please be aware of the fact, that your individual dataset organization depends on your AI app implementation.
You might want to test the functioning of your images, which can be realized by an image specific 'docker-compose.yml' file using the external volume 'ai_system' to mount the image internal path '/tmp' by 'ai_system:/tmp'. Don't forget to publish your images at your public dockershub profile and document the docker pull command at your report.

Examples can be found here:

- https://github.com/MarcusGrum/AI-CPS/tree/main/images/activationBase_apple_blurred_defect_01

- https://github.com/MarcusGrum/AI-CPS/tree/main/images/learningBase_apple_01

Subgoal 4: AI Model PyBrain/Tensorflow Creation and Data Visualization

Create an AI model in accordance with your project idea (this includes wiring, training with your training material prepared and testing with the testing/validation material prepared) using PyBrain or TensorFlow programming libraries and AI-CPS code and store your trained AI model called 'currentAiSolution.xml' (or whatever file type needed) on your local file system. Please document training and validation performance by storing corresponding files at '/learningBase/', such as number of training iterations, final loss and accuracy values of your corresponding training iteration as well as the corresponding visualizations, namely (1) the training and testing curves up to your corresponding training iteration, (2) the diagnostic plots and (3) the scatter plots as we have discussed in the exercise. Do not forget to document these at your report, too.

Examples can be found here:

- PyBrain: https://github.com/MarcusGrum/AI-CPS/tree/main/code/annRequests_PyBrain
- TensorFlow: https://github.com/MarcusGrum/AI-CPS/tree/main/code/annRequests_TensorFlow

Subgoal 5: OLS Model and Data Visualization

Create an OLS model realizing the very same task than your AI model (this needs to be set up based on the training material you prepared) using Statsmodels programming libraries and AI-CPS code and store your trained AI model called 'currentOlsSolution.xml' (or whatever file type needed) on your local file system. Please also realize testing routines with the testing/validation material you prepared so that you are able to compare OLS performance with AI performance. Please document training and validation performance by storing the relevant performance indicators and visualizations, namely (1) the diagnostic plots and (2) the scatter plots as we have discussed in the exercise. Do not forget to document these at your report, too.

Subgoal 6: AI Model Docker Provision

Create the following two docker image types, the first called 'knowledgeBase_*yourIndividualProjekctTitle*' and the second called 'codeBase_*yourIndividualProjekctTitle*'. The first provides your AI/OLS model (e.g. xml/pickle/h5 file), such as 'currentSolution.xml', at the path called '/tmp/ knowledgeBase/'. The second image provides the 'activation_data.csv' at the path called '/tmp/activationBase/'. Further, each of your images should be based on a 'busybox' image and contain the following two: first, your corresponding model file, second, a Readme.md that indicates a) your ownership, and clarifies, that your image was b) created as part of the course 'M. Grum: Advanced AI-based Application Systems' by the Junior Chair for Business Information Science, esp. AI-based Application Systems at University of Potsdam, c) a short characterization of your AI model and d) the commitment to the 'AGPL-3.0 license'.
Please be aware of the fact, that your individual folder organization depends on your AI app implementation.

You might want to test the functioning of your images, which can be realized by an image specific 'docker-compose.yml' file using the external volume 'ai_system' to mount the image internal path '/tmp' by 'ai_system:/tmp'. Don't forget to publish your images at your public dockershub profile and document the docker pull command at your report.

CodeBase examples can be found here:

- PyBrain: https://github.com/MarcusGrum/AI-CPS/tree/main/images/codeBase_ai_core_for_transport_classification_x86_64
- TensorFlow: https://github.com/MarcusGrum/AI-CPS/tree/main/images/codeBase_ai_core_for_image_classification_x86_64

KnowledgeBase examples can be found here:

- PyBrain: https://github.com/MarcusGrum/AI-CPS/tree/main/images/knowledgeBase_cps1_transport_system_01
- TensorFlow: https://github.com/MarcusGrum/AI-CPS/tree/main/images/knowledgeBase_apple_banana_orange_pump_01

Subgoal 7: Docker Builds and Docker-Compose Utilization

Create two 'docker-compose-yml' files that enable the application of your AI model trained on your example activation file prepared as well as the application of your OLS model trained on the very same example activation file prepared. Please be aware, for this, your docker-compose files need to use the following of your publicly available Docker's Hub images:
1) your image called 'knowledgeBase_*yourIndividualProjekctTitle*',
2) your image called 'activationBase_*yourIndividualProjekctTitle*',
3) your image called 'codeBase_*yourIndividualProjekctTitle*'.
Please, use the external volume 'ai_system' to mount the image internal paths '/tmp' by 'ai_system:/tmp' at all images used and delete any existing content here beforehand.

Docker-Compose application examples can be found Here:

- PyBrain: https://github.com/MarcusGrum/AI-CPS/blob/main/scenarios/apply_annSolution_for_transport_classification_with_pyBrain/x86_64/docker-compose.yml (please be aware that activation component is missing here due to publication)
- TensorFlow: https://github.com/MarcusGrum/AI-CPS/blob/main/scenarios/apply_annSolution_for_image_classification_with_tensorFlow/x86_64/docker-compose.yml

Submission Details

Please submit your AI project code **via public Github**, your images via **public Docker's Hub** not later than 6th of February 2025, 6 p.m. Do not forget to document your corresponding Github and Docker's Hub hyperlinks at your individual final team report (FTR), so that we can use your docker-compose.yml files for running your project.

Final Team Report (FTR)

Extend your "Individual Interim Report" (due date was December 5, 2024) so that your subsections are doubled and show the following: your first subsections show the individual weekly technical puzzle pieces you have developed during the first phase of the course. Your second subsections, show how you have applied these components in your AI project context. These represent your work during the second phase of the course and are a team artifact. However, each team member submits an individual "Final Team Report". As you have collaboratively realized your AI project, your team subsections may be content-wise redundant but make sure that its clear that both of you wrote the report on your own. Your final report should not have more than 5 pages per team member.

FTR Submission Details

Please submit your FTR via Moodle by 6th of February 2025, 6 p.m. at the latest.

Video Presentation (VP)

Create a team video presentation that presents your AI project having at maximum 150 seconds (plus/minus 10 seconds). In the video, you should tackle the following:

1. Introduce your AI system idea.
2. Clarify its data scraped.
3. Explain the AI architecture type and structure used.
4. Visualize its performance and relevant key-performance-indicators.
5. Present your team repository forked from the AI-CPS repository.
6. Explain how it uses the AI-CPS repository.
7. Present and explain the images built and provided at Docker's Hub.

The presentations will be watched at the exercise course at February 6, 2025. Best presentations will be published at the chair's Web 2.0. presence (www.lsikiwi.de, Youtube, etc.).

Video Submission Details

Please submit your VP via Moodle by 5th of February 2025, 10 a.m. at the latest.