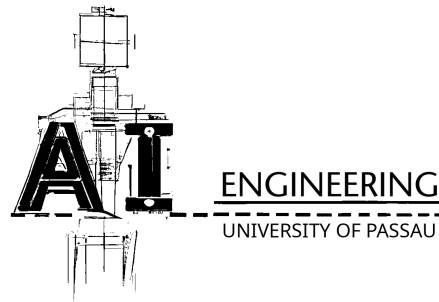


# Principles of AI Engineering

## Exercise 5

Prof. Dr. Steffen Herbold



Author: Lukas Schulte

## Project task

You have now created an API, implemented a modular text pre-processing and trained a random forest model. This exercise is about including the model in the API and storing data from its predictions.

Export the model from your Jupyter Notebook and integrate it into the API. Then implement the functionality for your two endpoints and re-use your text pre-processing. You will also need to decide on a database to store data.

`\api\predict`: Generate an ID for the issue, make a prediction based on the input, and return both the predicted label and the ID. Make sure to store important data in the database.

`\api\correct`: Access the stored prediction from the database and compare the result to the corrected label. Make sure to store important data in the database.

Hint: The file `api_ui_test_sample.zip` contains the title and description of some real-world bugs, enhancements and questions extracted from Mozilla repositories. You can use them to test your API manually. If you need more issues to test, check the JIRA or GitHub repositories of popular open source applications.

## Questions

1. Justify your decision for the type of database.
2. Describe how you will modularize your model to be integrated into the project's application (outside the Jupyter notebook).
3. Given the problem description, describe whether, in your opinion, misclassifications of bug, enhancement, question are equally bad or not and why.
  - Describe a setting, e.g., a project on GitHub where it might be the other way around.
  - Sketch a way to mitigate the case where misclassifications of certain classes matter.