**Project Documentation**

The project focuses on learning to use tools for conducting data set analysis.  
The selected dataset was downloaded from the Kaggle platform

* [Indicators of Heart Disease (2022 UPDATE)](https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease/data)

The task was implemented using VSCode, Kaggle, MLFlow, and a local installation on the computer's hard drive. (The rationale was access to a laptop with good technical specifications: processor, RAM, fast high-capacity SSD drives.)

On GitHub, the repository [Pawel20240101/MLOps\_PZ](https://github.com/Pawel20240101/MLOps_PZ) contains all the necessary information to reproduce and run the project.

Project Structure

* Data – directory for storing the dataset downloaded from Kaggle
* Doc – project documentation in .docx and .pdf formats
* Eksploracja – folder for .png files generated during input data analysis
* Models – directory for locally saved classifier models and result files (best parameters) from training on the test set
* Notebooks – required Jupyter notebook files for executing the project
* Reports – .csv files containing results for each classifier
* Results – graphical presentation of results, i.e., confusion matrix, ROC curve, and for models supporting SHAP – SHAP results visualization

Additional Files

* settings.json – file specifying the environment used in the project.  
   Due to occasional issues with incorrect environment selection or configuration, this file was prepared to solve the problem.  
   The environment is automatically selected when launching the project.
* requirements.txt – file listing the required libraries
* README.md – general project information displayed on the main GitHub page
* Przydatne\_polecenia.txt – file containing commands necessary for project execution

——————————————————————————————————————————————

The project consists of three files located in the notebooks directory.

### **Project Reproduction**

1. **Download the project from GitHub** In the VSCode terminal, type: **git clone https://github.com/Pawel20240101/MLOps\_PZ**
2. **Create the appropriate environment on your local drive** Refer to the file Przydatne\_polecenia.txt
3. **Activate your environment** Refer to the file Przydatne\_polecenia.txt
4. **Install the required libraries from the requirements.txt file** In the VSCode terminal, type: **pip install -r requirements.txt**
5. **Start the MLflow server** In the VSCode terminal, type: **mlflow ui**
6. **Run the files in the notebooks directory in the following order:** a. 1\_Import\_i\_eksploracja\_danych\_ML.ipynb  
    b. 2\_Przygotowanie\_danych\_ML.ipynb  
    c. 3\_ML\_Workflow\_pipeline.ipynb
7. **Check the results in MLflow**

Open a browser and go to: **http://localhost:5000**

### Description of the .ipynb Files Used in the Project

1\_Import\_i\_eksploracja\_danych\_ML.ipynb  
This notebook is responsible for downloading data from Kaggle and performing exploratory data analysis on the .csv dataset.

2\_Przygotowanie\_danych\_ML.ipynb  
This file prepares the dataset for the classification process through preprocessing and data formatting.

3\_ML\_Workflow\_pipeline.ipynb  
This is the core notebook that automates the full machine learning workflow: training models and logging results to an MLflow server. The process is executed fully automatically and consists of the following stages:

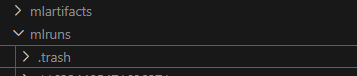
1. Installing and importing necessary libraries
2. Loading previously prepared data
3. Data preparation – identifying feature types
4. Preparing a pipeline for data preprocessing
5. Training models and logging results to MLflow
6. Comparing model performance
7. Hyperparameter tuning of the best model (using Optuna)
8. Saving the best model to a file
9. Prediction pipeline – generating predictions
10. Project summary
11. Completion

You can modify classifier parameters, experiment names, and more by editing the notebook:

**Step 1**, you can change the experiment name.

**Step 5**, you can include or exclude classifiers from training by commenting them out with #, or you can add new ones. In the latter case, corresponding sections later in the notebook must also be updated.

**Note:**Sometimes, the .trash subdirectory is not automatically created inside the mlruns directory. If this happens, you should manually create the .trash folder to avoid runtime errors.



Visualization of Results Using MLflow (Examples)

