**Guide to a successful project**

**Week 1 and 2: Basics (Everyone):**

The dataset you will work with contains diagnosis data for detecting breast cancer, it can be downloaded [here](https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data). For the diagnosis, digital pictures were produced via a fine needle aspirate (FNA). The data you receive is based on these pictures and describe various characteristics, also called 'feature' in the following, of the cell nuclei present in the image (like perimeter, concavity,…) that are used to classify (diagnose) each image into benign or malignant. The mean, standard error and "worst" (mean of the three largest values) of the 10 features are given resulting in 30 features per image.

**Prerequisites**

* Install Python and an IDE (PyCharm or Visual Studio) on your computer
* Create a virtual environment for your project and install required packages (pandas, dash, streamlit, sklearn, ..) via pip. (For windows, it is easiest to use conda or miniconda to install Python)
* Create a git repository to collaborate effectively
* For inital exploration it is easiest to work in a [jupyter](https://jupyter.org/install) notebook, make sure this is installed and working.

**Explore the dataset**

There are many blogposts availabel describing exploratory data analysis (EDA) in general and also for this dataset explicitly. Get inspired and use these resources to get started with Python and build an understanding of the data. Here are some guiding points:

* Load and explore the dataset
* Get familiar with the basics of the pandas library
* Use pandas functions like describe and plot to look at each column
* Plot the distribution of each column (also called 'feature' in the following) to see, whether there is a lot of variance, whether there are outliers, …
* Is the dataset imbalanced, e.g., are there more malignant than benign diagnoses?
* Plot and compute the correlation between each feature and the diagnosis. Are there any features which 'explain' the diagnosis to some degree?

For plotting packages like seaborn are useful, and there are also great packages to do EDA easily in python: [ydata-profiling](https://ydata-profiling.ydata.ai/docs/master/pages/getting_started/quickstart.html).

Once you're familiar with the data and Python showcase your results:

* Get familar with one app library, e.g. Dash or Streamlit
* Build a minimal dashboard with a dropdown to select the feature, and a chart which shows its distribution as well as the correlation with the diagnosis

**Week 3 and 4: Advanced features (smaller sub-teams)**

Split up in smaller teams or individual tasks to work on more advanced features, for example:

* Dashboarding
  + Improve the look-and feel
  + Add more graphs based on findings from the other teams
  + Add more interactivity
  + Play with different types of graphs
* Prediction Model
  + Build a minimal regression model, e.g., with sklearn to predict the diagnosis
* Play with LLMs:
  + Allow users to ask questions about the data using large language models like GPT using the new python package PandasAI:   <https://github.com/gventuri/pandas-ai>  (using LLM from Hugging Face Hub to avoid data security issues with openAI)
* Deployment (very advanced)
  + Deploy the webpage to the Cloud

**Remarks**

Also keep in mind following points before finishing the project:

* spend some time on code refinement - improve the final code by organizing your code into small functions and classes, try to follow clean code principles
* add a requirements.txt to your git
* and a README.md to your git, explaining repository structure, instructions to use the code,... etc.