

Reminders to finish last exercises and to our way of working: -1, 0
Prepare before class such that you can present your result: 1, 2, 3, 4
Worked on in class, but you can prepare them at home: 5, 6

Carefully read the instructions below!

(-1) FINALIZE LAST EXERCISE SHEET

Finalize all exercises of the last exercise sheet and upload the results to your defined GitHub repository.

(0) SETUP A GIT PROJECT ON GITHUB

In order to get a bit of `git` training done we work for all the exercises on GitHub.

- (a) Use the project created for the previous exercises or create a new **private** project in GitHub
- (b) Give the instructor (kandolfp) access to the project (see GitHub documentation for help)
- (c) Create a `pdm` project in your repository and commit the necessary files.
- (d) Create an appropriate structure in your repository for the rest of the exercise sheet (maybe have a look at the exercises to have a better idea first), not everything should be in the main folder.
- (e) Try to structure your work on the exercises with `git`, i.e.
 - Don't commit things that do not belong together in one single commit. Each exercise can be considered as a separate thing. Subparts of an exercise might be independent as well.
 - Use meaningful commit messages <https://www.conventionalcommits.org/en/v1.0.0/>
 - Make sure that you do not commit something that does not work - produces an error. If you have difficulties with an exercise you can also commit your best effort in this case.
- (f) Add a `README.md` that explains what you are doing, how to run the exercises and anything else that is necessary (quick guide to `pdm`), maybe note your name somewhere.
- (g) Optional: Work with issues, you can reference the issue in the commit message, GitHub documentation

(1) LAW OF LARGE NUMBERS We have seen the *law of large numbers* in action in the Monte Carlo Method computing π but how do we simulate a biased coin toss as discussed the lecture?

- (a) Create a way to simulate a slightly biased coin toss with 51/49.
- (b) Use `numpy` to have a fast algorithm
- (c) Illustrate simulations for a growing number of coin tosses and include the reference of the bias.
- (d) Can you make it such that the bias can be prescribed between 50 and 100?
- (e) Visualize your results, including a line for the expected result.

(2) ENSEMBLE LEARNING I

We continue working on the palmerpenguins data set. Select two classes to work on for an `VotingClassifier` from `sklearn.ensemble`.

- (a) Split your dataset into a training and testing set.
- (b) Define three different classifiers with the classifiers we saw last time (e.g. `SVC`, `DecisionTreeClassifier`, `LogisticRegression`).
- (c) Train and evaluate the `VotingClassifier`.
- (d) Show the scores of the individual classifiers and the final result.
- (e) Switch to soft voting and see if you get better results.

(3) ENSEMBLE LEARNING II

We continue working on the palmerpenguins data set. Select two classes to work on for an `BaggingClassifier` from `sklearn.ensemble`.

- (a) Split your dataset into a training and testing set.
- (b) Train a `BaggingClassifier` with the three different classifiers you used above
- (c) Can you visualize the results?
- (d) Combine the three `BaggingClassifiers` via an `VotingClassifier` and check your results.

(4) RANDOM FORESTS

We continue working on the palmerpenguins data set.

Recall the training with the Random Forest on this example from the last exercise sheet.

- (a) Train a random forest for this dataset for all the features.
- (b) Show the individual importance of the features for the trained `RandomForestClassifier`.

(5) GRADIENT BOOSTING

Recall the regression of Sheet 4, Exercise 5 and Sheet 5, Exercise 5. Lets work on this with gradient boosting by successive training with `DecisionTreeRegressor`.

- (a) Follow the steps from the lecture to train on the residuals with 5 levels.
- (b) Improve on it with the correct class from `sklearn.ensemble`

(6) QUESTION AND ANSWERS

Look through all the exercises and prepare questions you have on them after working on them. We try to find answers together in the lecture.

(7) WORKING ON NATURAL LANGUAGE

In this exercise we discuss approaches to work with natural language tools in python.