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

Carefully read the instructions below!

For the following exercises we work with the `Fashion-MNIST` dataset, you can download it with `sklearn.datasets.fetch_openml`.

(-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 on 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) Decision Trees

We continue working on the palmerpenguins dataset. Use a `DecisionTreeClassifier` for this dataset to separate along two features (see exercise 1) for all three species.

(a) With the help of a visualization of the tree explain how the separate into the two classes is done.

(b) Visualize the separation in a plot

(c) Try the fit for rotated data, i.e. exchange the order of the two features.

(2) LINEAR REGRESSION 2

Compare the findings from Sheet 2 Exercise 5 for E_2 with

- The results from `sklearn.linear_model.LinearRegression`
- Using the Moore-Penrose pseudo-inverse $(A^T A)^{-1} A^T$ that can be computed by `np.linalg.pinv` as described in the lecture with *closed-form* solution.
- Not all models are linear, is there a way to handle the following hypothesis function?

$$h(x_i, \Theta) = \Theta_2 \exp(\Theta_1 x_i) = y_i$$

Find/create a suitable set of data points and test your method.

(3) REGRESSION WITH SVM AND DT

Recall the polynomial regression of Sheet 2.

(a) Use a SVM to create a polynomial fit for different polynomial degrees. Show/visualize your results. Play around with the parameters to create a good fit.

(b) Use a decision tree to create a fit for the regression show/visualize results for different depths. Play around with `min_samples_leaf` to create a good fit.

(4) RANDOM FOREST We continue working on the palmerpenguins dataset. Use a `RandomForestClassifier` for this dataset to separate along two features (see sheet 2 exercise 1) for all three species.

(a) With the help of a visualization of the tree explain how the separate into the two classes is done.

(b) Try the fit for rotated data, i.e. exchange the order of the two features.

(c) Visualize the feature importance.

(5) RANDOM FORESTS 2

We continue working on the palmerpenguins dataset.

(a) Train a random forest for this dataset for all the features.

(b) Show the individual importance of the features for the trained `RandomForestClassifier`.

(6) GRADIENT DESCENT - LEARNING RATE

We recall our minimization function from sheet 2, exercise 5:

$$MSE(\Theta) = \frac{1}{m} \sum_{k=1}^{m} (\Theta^{\mathsf{T}} x^{(k)} - y^{(k)})^2.$$

(a) Compute

$$\frac{\partial}{\partial \Theta_i} MSE(\Theta). \quad \text{for} \quad i = 0, 1.$$

The closed form to compute this looks like

$$\nabla_\Theta MSE(\Theta) = \frac{2}{m} X^{\mathsf{T}} (X\Theta - y).$$

for

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 4 \\ \vdots & \vdots \end{bmatrix}.$$

(b) Define a function that updates Θ according to the GD algorithm with a parameter η (make sure your vectors are column vectors).

(c) Compute the minimum with a learning rate $\eta = 0.01$ and 1000 epochs.

(d) Try visualize the learning rate for different η.

> Try to solve these problems on your own. If you get stuck, maybe some of your fellow students can help you. Why not write down your question in the course forum in SAKAI. Please try not to write back entire solutions or exchange solutions in the forum, this is not the idea and will backfire in the long run (also we might decide to delete such posts).