# Lab Class ML:I, ML:II

Until **Monday, Nov. 11th, 2019, 11:00 am CET**, the following exercises must be submitted:

**Students receiving 4.5 ECTS:** 2, and 3.
**Students receiving 6 ECTS:** 1, 2, and 3.

**Lab Class General Instructions.** Exercises should be completed in groups of three students; which and how many exercises you must submit depends on the number of ECTS credits awarded for the course, based on which degree programme you're studying:

| | 4.5 ECTS | 6 ECTS |
|---|---|---|
| Degree programmes | CS4DM, CSM, MI,<br>HCI *(enrolled 2018 or earlier)* | DE,<br>HCI *(enrolled 2019 or later)* |
| For admission to the final exam: | | |
| Must reach at least | 80 points | 120 points |
| Out of maximum | 120 points | 180 points |

For each exercise sheet, all required exercises must be submitted until the stated deadline. Each individual exercise will receive a score between zero and ten. Submissions received after the deadline will not be considered for grading. In order to be admitted to the final exams, you must reach the minimum total score shown.

Upload your solutions to Moodle as a single `.pdf` file, plus one `.ipynb` file for any programming exercises. Make it clear which solution corresponds to which exercise. Programming exercises are marked with $\boxed{\text{P}}$ – their solutions must be documented extensively with docstrings and/or inline comments. ml.weimar.webis.de

## Exercise 1 : Machine Learning (general)

(a) Define the terms "supervised learning" and "unsupervised learning."

(b) Sketch for each learning paradigm a typical problem.

(c) Which design decisions are to be made during the development of a learning system?

(d) A basketball trainer wants to design a learning system that can predict how high players trying out for his team will be able to jump. What is the learning paradigm in this case? What could be a useful model formation function?

(e) Name an example of a problem which cannot be solved by learning. Explain your answer.

## Exercise 2 : Linear Regression

The table below describes four cars by their age and stopping distance for a full braking at 100km/h till stop:

| Car | Wartburg | Moskvich | Lada | Trabi |
|---|---|---|---|---|
| Age (year) | 5 | 7 | 15 | 28 |
| Mileage (km) | 30 530 | 90 000 | 159 899 | 270 564 |
| Stopping distance (meter) | 50 | 79 | 124 | 300 |

(a) Determine (by hand) the weights $w_i$ for the linear regression for the age variable.

(b) Extrapolate the expected average stopping distance for a 15-year old car. Note: use the model from (a).

(c) Consider the mileage of the cars as an additional variable and repeat (a) and (b) under this setting.

(d) Draw a scatter plot of the data and the linear regression for a variable of your choice.

(e) Discuss the problems and pitfalls of extrapolation.

## Exercise 3 : P Basic Data Analysis and Linear Regression

In this course, we will use the Python programming language, version 3, to work with data sets and implement fundamental machine learning algorithms. The following exercises will help you prepare for subsequent programming assignments.

(a) Log into the Moodle page for the course; the required enrolment key will be announced at the first lab class. On the first page, follow the link labeled “Jupyter Notebook Server” – this will start a server with a Python environment just for you, with which you can interact through your browser. Read Section 1 from www.scipy-lectures.org. If you are new to Python, consult more basic sources first, such as docs.python.org/3/tutorial and www.diveintopython3.net. While looking at this material, create a Python notebook in the Jupyter notebook server, and try a few things.

(b) Download Fisher’s *Iris* data set from archive.ics.uci.edu/ml/machine-learning-databases/iris; the file `iris.data` contains the actual examples, and `iris.names` contains meta-information including the column names. You can download the file using your browser, and then upload it to your notebook server, or you can try to figure out how to make the notebook server download the file directly, for example with a small Python program. Once you have the file, write a Python program that reads the data set into memory and computes the mean, minimum and maximum of the *petal width*, *petal length*, *sepal width* and *sepal length* attributes for each of the three species of flower. Based on your results, which of the flower species do you think will be easy to distinguish, and which will be hard?

*Hint: the data file is in comma-separated-value (CSV) format – try to find a module in the Python standard library that can make reading the file easy.*

(c) Using the `matplotlib` library, draw a scatter plot that shows the *petal length* attribute on the x-axis, and the *sepal length* attribute on the y-axis. Use different colors for the three different species and label the axes.

(d) Create a subset of the Iris data that contains only the *sepal length* attribute, and only the *setosa* and *virginica* classes. Draw a scatterplot showing the attribute on the x-axis and the class on the y-axis. Using the LMS algorithm given in the lecture, compute the weight vector $(w_0, w_1)$, and add the line of best fit to your plot. What is the residual sum of squares (RSS) for your weight vector?