Learning from Data – Week 1
Assignment 1

General remarks

The aims of this assignment are the following:

- to get you acquainted with running a basic learning experiment with scikit-learn
- to understand the principles of a learning setting
- to interpret results according to commonly used measures
- to start reflecting on how certain learning algorithms work (Naive Bayes)
- to start getting used to writing up experimental results in a research-oriented way

Practicalities:

- assignment files are on Nestor
- you should complete this assignment in **groups** of 3 people
- what to hand it (please, upload on Nestor):
  - modified code (`LFD_assignment1.py`)
  - a report written as a short research paper including experimental details, comments, and answers to questions (`pdf`). For the report, you should use the template that we have prepared, and that you can find on Nestor. The code (and comments) are 25% of your grade, the full report the other 75%.
- deadline: **13th of September, 10:59 AM**.

Data

You are given two files for this assignment:

- `reviews.txt`: a corpus of reviews. Each review is on one line, and is headed by two meaningful tags:

  - a tag that specifies one of six topics: `books, camera, dvd, health, music, software`
  - a tag which indicates the sentiment expressed by the review, in terms of a positive or negative value: `pos, neg`.

  A third column contains the id, and the rest is the review's text. The text has already been tokenised.

- `LFD_assignment1.py`: a script to run a Naive Bayes classification on this data, using the scikit-learn libraries. **Note**: you should run this using **Python 3**. Of course, not all functionality you might need is already in this script. You'll likely have to use Google and the scikit-learn documentation to find what you need.

Setting up

It's a good idea to set up a virtual environment for this class. I suggest doing this using Conda. This example code (for Ubuntu) should take care of the set up for this assignment.

```
conda create -n lfd python=3.7
conda activate lfd
pip install sklearn
```

Now check if you can run the script succesfully. Also run it once with the -h argument, to see what your options for command line arguments are. Try to understand what they mean and what you can do with them. It's encouraged to add your own arguments throughout the assignment, if you find a need for them.

```
python LFD_assignment1.py
python LFD_assignment1.py -h
```

Exercise 1.1 – Settings and Comments

The python script `LFD_assignment1.py` contains several functions. You will have to add comments to this script to show that you understand what each function is doing, and why. All places where you have to add a comment are marked with a comment like this:

```
# TODO: comment
```

Exercise 1.2 – Binary vs Multi-class Classification

As you can see from the data, each review is tagged with a sentiment label and with a topic label. You have to run the script so that you can use both types of classification. Note that you will have to work on Exercise 1.3 and Exercise 1.4 in both settings (namely with two classes and with six classes). It won't really matter so much for the script, but it will matter for what you observe in terms of results and thus what you discuss in the report (Exercise 1.5). There also the option to shuffle the data set before splitting in training and test. How does the influence performance? And what about the performance of the baseline?

Exercise 1.3 – Measures

The script as it stands only outputs the general accuracy of the system. Checking the scikit-learn documentation, find out how to import and use the measures we have seen in class, and print them out. You will have to produce: *precision*, *recall*, and *f-score*.

Remember that these have to be calculated (and printed) *per class*. You should also output a confusion matrix. In the report (Exercise 1.5), please include a few comments on the results. For example: Is performance on one class better than performance on the other(s)? Can you speculate on why this is the case and potentially what could be done to change things?

Exercise 1.4 – Probabilities

The algorithm we are using for this little experiment is Naive Bayes, whose properties we have seen in class. For this portion of the assignment, you have to modify the script so that you can output probabilities. For the method to use, check the scikit-learn page on Naive Bayes:

`scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html`

In the report (Exercise 1.5), you will have to comment on how prior and posterior probabilities are calculated. You can also report some example reviews and discuss the probabilities assigned to them (why are posteriors different from priors?).

Bonus: this allows for some error analysis: can you find some examples from your test set in which the classifier was wrong by a large margin (i.e. correct class had a very low probability), for the sentiment analysis task? Was this to be expected by looking at the reviews? Can you spot any features that lead to this? Please show these (shortened) examples in a nice way in the report.

Exercise 1.5 – Report

You are asked to write a short report where you explain generally what the script does and what you have done to modify it. You should also describe all experiments you have run for the exercises above, and include any comments and/or answers that you were asked to provide. Additionally, you will have to answer the following questions:

(a) Why should one not look at the test set?

(b) What baselines could you use for the binary sentiment classification and for the six-class topic classification? What would their performance be?

(c) What happens with cross-validation? Did using it influence your results?

(d) Why is it useful to look at the confusion matrix?

(e) What features are used in the classification task you ran?

(f) What is the difference between using accuracy and using F-score?

(g) What is the difference between macro F-score and micro F-score and what different functions and implications do they have?

Note that you do not need to literally copy and paste the questions in the report. Try to answer them in a more organic way, explaining things as they come up throughout the report.

Important — Please note that we have prepared a template that you should use for completing your report. The template is structured along the lines of a research paper, and you can fill each appropriate section with the relevant information. The idea is that you get used to using the standard format adopted in research to report on experiments. At times this might feel a little stretched in the context of homework and the exercises you are asked to complete, but give it a try. Don't get too hung up about what should go where: try make decisions, and we will give you feedback. Additional questions can be answered in the final section. Since you will be working in a group, I suggest using Overleaf.