Assignment 1: text categorization

Text mining course

This is a **hand-in assignment for groups of two students**. Send in via Brightspace **before or on Tuesday October 7**:

- Submit your report as PDF and your python code as separate file. **Don't upload a zip file containing the PDF** to enable reading the PDF directly in Brightspace.
- Your report should **not be longer than 3 pages** (being concise is an important lesson!)
- Do not copy text from external sources. Reuse of code is no problem (and intended because we build on existing packages and tutorials).

Goals of this assignment

- You can perform a text categorization task with benchmark data in scikit-learn.
- You understand the effect of using different types of feature weights.
- You can evaluate text classifiers with the suitable evaluation metrics.

Preliminaries

- You have completed the sections 7.2.3. Text feature extraction and Classification of text documents using sparse features of the scikit-learn user guide (**exercise week 4**)
- You have all the required Python packages installed

Tasks

1. The tutorial classifies between only four categories of the 20newsgroups data set. Change your script so that it addresses all 20 categories.
2. Compare three classifiers in sklearn on this multi-class classification task, including at least Naïve Bayes.
3. Compare three types of features for your classifiers: counts, tf, and tf-idf. Keep the best combination of a classifier and a feature type for the next task.
4. Look up the documentation of the `TfidfVectorizer` function (which has the same parameters as the `CountVectorizer` function discussed in lecture 4) and experiment with different values for the following parameters. For each of these parameters compare different values and store the results.
    a. Lowercasing (true or false)
    b. stop_words (with or without)
    c. analyzer (in combination with ngram_range), try out a few values
    d. max_features, try out a few values
5. Write one script or notebook for running these experiments and printing the results.

Report writing

Write a short report (3 pages is the hard maximum) with the following structure:

1. Introduction: briefly introduce the task
2. Methods: describe which comparisons you made (classifiers, features)
3. Results: show the results tables (Precision, Recall, and F1) for the classifiers and features
4. Discussion: write a brief discussion on which classifier performs the best, with which features
5. Reflection (this part can be on the 4th page):
    - Briefly describe the work division between the two team members.
    - Did you use AI assistants in research or writing? If yes, please specify briefly how: (a) which assistant did you use, (b) for which tasks (understanding, programming, debugging, writing, editing), (c) and your reflection on the use: was it helpful, did it work, did you notice any errors in the output? (maximum 1 paragraph)

Note 1: You can choose any template/style for the report but please work in Overleaf with your university account.
Note 2: Please be referred to the slides of lecture 1 for the rules regarding use of AI assistants. For this first assignment, we want you to think critically about this, and therefore we will not report AI use to the Board of Examiners.

Grading rubrics

Maximum 2 points for each of the following criteria:

- General: length correct (maximum 3 pages) and proper writing + formatting
- Experiments on 20 newsgroups
- Results table for 3 classifiers x 3 feature weights (counts, tf, and tf-idf)
- Results for a number of different settings for a. lowercase; b. stop_words; c. analyzer (in combination with ngram_range); d. max_features
- Brief discussion on which classifier performs the best, with which features