The Mathematics of Machine Learning
Homework Set 4
How to Become a Successful Spammer

Due 12 March 2025 before 13:00
via Canvas

- You are allowed to work on this homework in pairs. One person per pair submits the answers via Canvas. Make sure to put both names on the submission.
- You have to submit both a Jupyter Python notebook, and the answers to the questions below. You may either answer the questions in a separate document or by inserting text into the notebook directly.

The goal of this exercise is to see naive Bayes in action for spam classification. We will be taking the perspective of a spammer, who wants to get their spam message to pass the naive Bayes spam filter. The data come from the Kaggle machine learning competition website.

- Download and unpack the Ham-and-Spam data from Canvas to get a `hamnspam/` folder with e-mail files.
- You will extend the `Homework4-start.ipynb` Jupyter notebook, which is available from the course website. This notebook already contains the code to read in and prepare the data, and to train an accurate naive Bayes spam filter. Read through the notebook and make sure you understand all the steps.

1. [2 pt] The notebook is a bit sloppy about preprocessing the data: it selects the set of words that will be included in the dictionary based on the combined train and test sets. What is wrong with that?

Near the end of the notebook you will find an `email`. This is the e-mail that you, the spammer, want to sneak past the spam filter. Unfortunately the email is currently classified as spam by the naive Bayes classifier. Your goal is to modify the email so that it is classified as ham, but you want to change as few characters in the email as possible, so you should do it in a principled way that exploits your in-depth understanding of the spam filter.

2. [4 pt] Describe a principled strategy to add and/or remove words from the e-mail in a way that will have a strong effect on the classification of the naive Bayes classifier.

3. [4 pt] Use your strategy to modify the email such that it is classified as 'ham'. Report the number of characters that you had to change. This should be less than 500. Count as 1 change: adding a character, removing a character, or changing a character. This is known as the *edit distance*. The notebook already contains code to automatically calculate this number of changes from the original email. NB Make sure not to modify the `origemail` variable.

The best solution, i.e. with the smallest number of characters changed, will be announced on Canvas in the "homework 4 hall of fame"!

Coding hints:

- You can obtain a numpy array with the logarithms of the binomial probabilities $(\theta_{k,1}, \ldots, \theta_{k,2500})$ for class k using `mnb.feature_log_prob_[k,:]`.
- For any numpy array `a`, `a.argsort()` returns the indices that would sort the array. See `https://numpy.org/doc/stable/reference/generated/numpy.argsort.html`.
- Given a list of indices in the dictionary jlist $= (j_1, j_2, \ldots, j_m)$, you can obtain the corresponding words using `np.take(cv.get_feature_names(),jlist)`.