Programming assignment 3: Stance classification

In this assignment, you will solve a supervised machine learning task and write a report that describes your solution. The data that you will use for training and evaluation will be annotated collectively by all participants in the course.

The machine learning task that will be addressed in this assignment is to develop a text classifier that determines whether a given textual comment expresses an opinion that is positive or negative towards COVID-19 vaccination.

The first two parts of this assignment deal with data annotation and are solved **individually**. In the third and final part, you will implement the classification system, and here you will work **in a group of two or three people**.

Didactic purpose of this assignment:

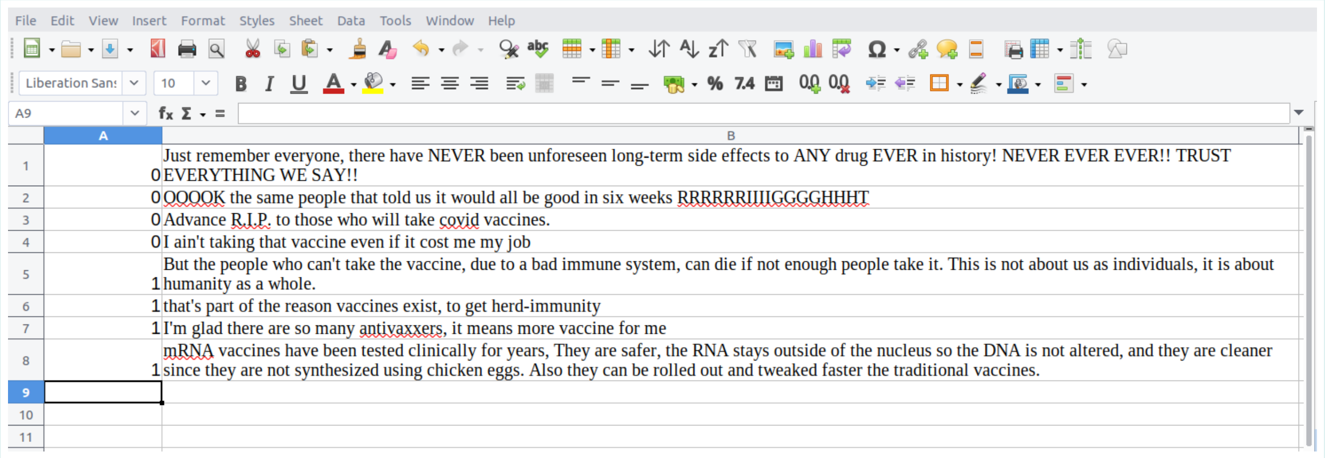
- Getting some practical understanding of annotating data and inter-annotator agreement.
- Practice several aspects of system development based on machine learning: getting data, cleaning data, processing and selecting features, selecting and tuning a model, evaluating.
- Analysing results in a machine learning experiment.
- Describing the implementation, experiments, and results in a report.

Part 1: Crowdsourcing the data

Your task here is to collect at least 100 comments in English relating to COVID-19 vaccination from social media or the comment fields from online articles. Good places to trawl for comments include social media sites such as Youtube or the website of any English-language newspaper (that allows readers to comment).

Collect comments that express a pro- or anti-vaccination stance. We will create a balanced dataset, so you should try to collect about 50 instances of each stance. **Do not include comments not expressing an opinion about vaccination. Also, since other annotators will see each comment in isolation, don't include comments where you need to read previous comments to understand the opinion (e.g. *"You're wrong!"*).** Try to select comments from a variety of sources.

Store all the comments you collected in an Excel file. This file should have two columns. The first column will store your **annotation** of whether this comment is pro-vaccination (represented as **1** in the spreadsheet) or anti-vaccination (**0** in the spreadsheet). The second column should store the text of the comment. Make sure that the text of each comment is stored in a single cell. The following figure shows an example.



	A	B
1	0	Just remember everyone, there have NEVER been unforeseen long-term side effects to ANY drug EVER in history! NEVER EVER EVER!! TRUST EVERYTHING WE SAY!!
2	0	OOOOK the same people that told us it would all be good in six weeks RRRRRRIIIIGGGGHHHT
3	0	Advance R.I.P. to those who will take covid vaccines.
4	0	I ain't taking that vaccine even if it cost me my job
5	1	But the people who can't take the vaccine, due to a bad immune system, can die if not enough people take it. This is not about us as individuals, it is about humanity as a whole.
6	1	that's part of the reason vaccines exist, to get herd-immunity
7	1	I'm glad there are so many antivaxxers, it means more vaccine for me
8	1	mRNA vaccines have been tested clinically for years, They are safer, the RNA stays outside of the nucleus so the DNA is not altered, and they are cleaner since they are not synthesized using chicken eggs. Also they can be rolled out and tweaked faster the traditional vaccines.
9		
10		
11		

Important. We will receive a large number of Excel files and they will need to be processed automatically. For this reason, it is important that you format the Excel file ***exactly*** as described above. That is, if you insert column headers, change the labels to something other than 0/1, change the order of columns, or add "helpful" comments, you are introducing errors that have to be fixed manually.

Submit the Excel file via the Canvas page. If you have trouble using Canvas, please send your solution by email to Richard directly, with the subject line *Applied Machine Learning: Programming assignment 3 part 1*.

This part of the assignment is solved **individually**.

Deadline for Part 1: February 5

Part 2: Second round of annotation

After you have submitted your annotated comments, you will receive back a set of about 100 other comments. You will find these comments as an attachment to the feedback comment in Canvas. Annotate them as well, and submit the file containing your annotations. **If you think it's impossible to understand a comment as pro-vaccination or anti-vaccination, you can enter the value -1, which will mean "I don't know".**

This part of the assignment is solved **individually**.

Deadline for Part 2: February 9

Again, submit the second Excel file using the Canvas page. And again, use email if you have trouble with Canvas, this time using the subject line *Applied Machine Learning: Programming assignment 3 part 2*.

Part 3: Implementing your stance classification system

Write the code to implement a classifier that determines whether a given comment expresses a pro-vaccination or anti-vaccination stance. Initially, you will work with a small sample that you can use to get things set up. Eventually, you will receive the full dataset: first including the result of the first annotation, and later the result of the second round. Please note that your results may change (e.g. which model performs best) when you switch from the small sample to the full dataset.

In your implementation, you are free to use any machine learning approach you think could be useful: the only restrictions are 1) that you are not allowed to use existing implementations that carry out exactly this task (that is: classifying whether a text is pro- or anti-vaccine); 2) that your models should be possible to run in a stand-alone fasion (i.e. they should not use an external service). You may take some inspiration from the document classification examples shown in Lecture 3. However, it is probably useful to try to improve over this solution. For instance, you may read more about the TfidfVectorizer and see what you can do with it. Optionally, try out a powerful modern text representation model such as BERT.

Then write a report detailing your implementation, your experiments and analysis. In particular, some useful issues to discuss might include:

- How much consensus is there between annotators of the dataset? Do you think the data is reliable?
- How do you represent your data as features?
- Did you process the features in any way?
- How did you select which learning algorithms to use?
- Did you try to tune the hyperparameters of the learning algorithm, and in that case how?
- How do you evaluate the quality of your system?
- How well does your system compare to a trivial baseline?
- Can you say anything about the errors that the system makes? For a classification task, you may consider a confusion matrix. It is also probably meaningful to include selected errors and comment on what might have gone wrong.
- Is it possible to say something about which features the model considers important? (Whether this is possible depends on the type of classifier you are using.)

The submitted report should be around 3–4 pages. Use the following template to write the report. It should be written as a typical technical report including sections for the introduction, method description, results, conclusion. The report should be a pdf, Word or LibreOffice document. Please include the names of all the students in the group.

The code should be a Jupyter notebook.

Please use the Canvas page to submit your solution (the code and the report). If you have trouble using Canvas, please send your solution by email to Richard directly, with the subject line *Applied Machine Learning: Programming assignment 3, part 3*.

Grading. Grading will be based (1) on whether the report is insightful and lives up to professional standards of technical writing, including decent clarity, spelling, grammar, and structure, and (2) on whether the technical solutions are justified and the code well-implemented.

Clarification. Your report should not be in the form of a bullet list that just goes through the discussion points listed above. It should be a typical technical report, written in a clear and readable manner.

Deadline part 3: February 16

The datasets and their format

Please check this page regularly, since we will post the new datasets after they have been collected.

The **training data** will be stored in a text file consisting of tab-separated columns, where the first column contains the output labels (1 for pro-vaccination, 0 for anti-vaccination) and the second contains the comments. To exemplify, here are some of the examples in the sample data:

```
0       the vaccine is a hoax, based on lies.
1       I will get the vaccine as soon as I can.
1       stupid anti-vaxxers!
0       don't believe what they tell you, it's not safe!
```

Here is a preliminary sample (1000 instances) that you can work with when you start coding. After the first round of annotation has been completed, you'll get a larger set.

After the second round of annotation has been carried out, we will distribute the data once again, including the annotations from all the annotators. Here is an example of how this will look. As you can see, the different annotations are separated by a slash (/). In some cases, as in the last example below, annotators may disagree.

```
0/0       the vaccine is a hoax, based on lies.
1/1/1       I will get the vaccine as soon as I can.
1/1       stupid anti-vaxxers!
0/1       don't believe what they tell you, it's not safe!
```

There will also a separate **test set**, which will also be published when we have the final annotation. As usual, the development of your system should not use the test set in any way, and you should only compute the test-set score after finalizing your system.