Some recommendations and guidelines on how to write reports for assignments and projects in Data Science

Layout and overall presentation

- Make sure that the document you are producing has reasonable margins, like the margins of this document or slightly larger (2.5-3 cm for the left and right margins, and 4cm for the top and bottom margins). The default setting of Latex typically produces margin that are too wide. You can use the package fullpage to fix this. You just need to add `\usepackage{fullpage}` in the header of your latex document. There is no particular reason to use a conference or journal paper format, but if you would like to do this, you may.

Figures

Figures are a frequent source of migraines for readers and reviewers in scientific publications. Make sure

- that your figures are not too small (preferably no less than half of the width of the space between the margins on the page)
- that the lines in your plots are sufficiently thick. The default setting in several programming languages produces lines that are enough for screen visualization but yield printed figures that have lines that are too thin.
- the same holds for the markers of points in point clouds. Please make them sufficiently big and if there are different classes, make sure you represent them with different colors.
- A figure must have
  - axis labels
  - a precise title
  - a visual legend in or next to the plot specifying which curves are represented in which color, with which line style etc.
  - and last but not least and probably even *most importantly: a caption* that describes what the figure represents and reminds the reader of which dataset, what dataset size, what method, what hyper-parameters values, etc have been used to obtained the figure. As much as possible this caption must help the reader to understand the figure, without that he would have to search for details in the rest of the report, or possibly indicate where the details are provided. So if you use names of variables such as M or φ or acronyms, make sure that they are defined in the caption. In particular, when you report error curves of statistical performance, you should always make sure that you state whether the curve you are plotting is obtained with the training set or the validation set.

  If possible do not export a figure that consists of curves in an image format such as JPEG. This is a terrible idea because the compression algorithm will try to approximate blocks of the “image” of the curve with a discrete cosine transform. Given that the image has very sharp edges the resulting image will smear out your curves and exhibit fairly ugly *ringing artifacts* due the *Gibbs phenomenon.* A fairly good format for curves is EPS. For heatmaps or cartoonish illustrations, you might consider the PNG format.

Sanity checks, and reporting of crucial experiments

In machine learning, there are two absolutely crucial questions that need to be evaluated before starting to interpret the behavior of an algorithm on a given data set, namely

- **Convergence** Is you algorithm properly implemented and bug free, in particular, if it contains an iterative procedure, does it converge and does the objective function decrease or increases as it is supposed to.
- **Overfitting** Have you chosen well the hyperparameters of the algorithms or the procedure to chose them so as to avoid overfitting?

Please include in you reports all the curves and figures that illustrate that these two questions can be answered positively (or, if not, which issues you encountered) such as curves showing how the likelihood or the objective function increases or decreases with the iterations and provide systematically results on the training set and on the testing or validation(s) sets.

Questions of interpretation of the results of experiments

When prompted to provide comment results of experiments, keep in mind that mere factual descriptions of the results are uninteresting.

While some factual elements are necessary, you should always try to provide some interpretation by relating what it happening to main concepts such as overfitting, generalization, curse of dimensionality, adequacy of the model (is the model well-specified or not), sample size, choice of the hyperparameters, characteristics of the data such as separability, correlation, sparsity, presence of outliers, cluster structure, intrinsic dimensionality of the cloud of datapoints, etc.

Also, before you get into complicated interpretations of the strange behavior of your algorithm, question first whether the algorithm has been properly executed, whether the results meet a certain number of sanity checks that should reassure you or not on whether the algorithm has converged, could contain a bug, etc.

Structure of a project report

This section discusses the structure that will usually be expected for a project report. A project report generally has the following template

1. Introduction
   (a) context
   (b) general problem studied, general methodology
   (c) focus and goal of the work presented
   (d) structure of the rest of the report
2. Methods
   (a) specific problem studied
   (b) related work

    (c) model, main idea
    (d) specific methodology
    (e) algorithms

3. Experiments
    (a) Description of the dataset(s) considered / general problem associated with the data
    (b) Description of the protocol of the experiments (setting of the hyperparameters/cross-validation procedure/evaluation methodology)
    (c) Factual description of the type of results reported (explanation pertaining to the Figures, tables, etc)
    (d) Interpretation and discussion of the results (comparison with the baselines, advantages of each algorithm, etc)

4. Conclusion
    (a) summary
    (b) main conclusions and take home messages
    (c) Remaining questions/ future directions (only if relevant)

Of course, this structure is just to give you a general idea and should not to be followed by the book or to the letter. In particular, it should not be reproduced as specified above in your reports and you should replace the general titles that are given above with specific personalized titles that correspond to your project and which give a idea to the reader of what he will read in the corresponding sections.

The outline above is essentially meant to give you a general idea of the progression of the report. You should certainly not add a section if it seems artificial and if you do not know what to put in it. Depending on the work that you are presenting some sections can disappear and others can be subdivided or added. For example, if your work requires to first present a general theory which it would be too long to develop in the introduction, you may add a section 2 on general methods before you introduce the line of work that is more closely related to what you have been working on in a section 3 and postpone the experiments to a section 4.

Appendix, technical sections and code

The report may contain an appendix, beyond the number of pages of the report requested. Put differently, if the report should, for example, be a six page report, this does not include the appendix pages. The appendix could count more pages than the report, but keep in mind however that we will focus mainly on the report: we will not read in minute details the appendix, and if it is too long, we are very likely to give up before we reach the end.

The main reports should only contain some of the main equations that are necessary to understand your models and algorithms. Precise derivations and calculations leading to a specific result should be deferred to an appendix at the end of the report.

The same holds for code: if you feel relevant to include some reasonably small portions of your code, then it should be included in an appendix. You can however include an algorithm in pseudocode in the body of the paper if it is not too complicated. To include code you can use the following package `http://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings` `ftp://ftp.tex.ac.uk/tex-archive/macros/latex/contrib/listings/listings.pdf`