15-388/688: Tutorial

October 16, 2016

> Docendo discimus (we learn by teaching)
>
> *Latin proverb*

Important Due Dates (due 11:59pm)

- **9/26** – One-sentence proposal
- **10/19** – Check-in draft submission
- **11/2** – Tutorial due (max ***two*** late days allowed, so final deadline **11/4**)
- **11/9** – Student tutorial evaluation due

Introduction

When studying data science (though the same applies to virtually any other disciplines too), the best way to learn a particular topic is often to teach it to others. With this in mind, the tutorial assignment of this course is meant to give you the chance to create a mini tutorial on some subject related to to data science. The potential topics are quite broad: you can describe how to use a particular algorithm, library, methodology, or data set (there has to be some amount of data collection or substantial processing involved if you go the data set route, you cant simply run a bunch of algorithms on a pre-curated data set). Your tutorial should provide an introduction to this topic suitable for the level of other students taking this course (you will ultimately be graded partly by other students taking the course, so keep them in mind when developing your document).

Requirements for final submission

Your tutorial should be in the form of a Jupyter notebook, which mixes together written markdown and code portions. You will walk students (your readers) through the algorithm, library, methodology, or data that you are presenting, explaining both the high-level concept and including examples of code. You want to make the tutorial read like an actual explanation of the process or technique, not just as a listing of code: as a general rule of thumb, you should have a paragraph or two of prose explaining any function or class method you include in the code.

Your notebook should contain between 1000-2000 words of prose in markdown sections, and between 100-250 lines of code (not counting comments). The script provided on the web site: `http://www.datasciencecourse.org/count_length.py` will check the size of your file and let you know the length of the assignment as we will be evaluating it.

For an example of the right type of balance, we have released an example tutorial on geospatial analysis, available on the course webpage: `http://www.datasciencecourse.org/GIS%20Tutorial.ipynb`. or viewable as a static webpage (this link directs to the nbviewer web app, which will render a static version of any notebook by passing the proper url): `https://nbviewer.jupyter.org/url/www.datasciencecourse.org/GIS%20Tutorial.ipynb`.

This last link also illustrates a key point: your tutorial must be fully readable as a *static web page* (that is, you should pre-generate any output that is required to understand the content, such as figures, etc). You don't necessarily need to render it in the nbviewer web app, but your tutorial should be readable to students as soon as they open it in jupyter notebook, without running any cells. This is extremely important, because there is a good chance that some initial call in your notebook may fail on someone else's machine, and if you don't pre-generate all the relevant outputs, the tutorial won't be useful in this case. Of course, students *should* be able to run the code if they want, but the tutorial needs to serve its purpose even if they are unable to do so. In fact, *the instructor and TA evaluations will not run any code in the tutorial notebooks, we will just be reading the notebook as a static web page* (without this condition, it would take to realistically evaluate all tutorials).

Peer evaluation

In addition to submitting your tutorial, each student will be required to two other tutorials. These peer evaluations will factor partly into your final grade for the tutorial. When evaluating other students' submissions, you will use the rubric described in the next section. Although two is a relatively small number, we will make all the tutorials available within the class, so that students can form a baseline about what they believe to be a good tutorial, and help to contextualize their grading.

Grading rubric

Both the peer evaluations and the instructors will use the following rubric for grading the tutorials. All the following questions are answered on a 1-10 point scale.

- **Motivation.** How well does the tutorial motivation the importance of the topic to the data science setting?
- **Understanding.** After reading the topic, how well do you feel you understand the general ideas behind the tutorial? If you already knew the subject: how well do you think the tutorial was able to describe the basic ideas as you understand them?
- **Further resources.** Does the tutorial helpfully suggest where to look for additional resources on the problem. Do these resources seem like they would help you further your understanding of the topic or ideas after you have completed the tutorial?
- **Prose.** Did the prose in the markdown cells contribution to your understanding of the material?

- **Code.** Did the included code provide helpful examples or otherwise solidify your understanding of the topic?
- **Subjective evaluation.** If you were reading this tutorial as a blog post linked from <insert your favorite CS-themed news site here>, how likely would you be to read the whole thing?
- (No points assigned) Write, in 100-200 words, one thing you particularly liked about the tutorial
- (No points assigned) Write, in 100-200 words, one thing you think could be improved about the tutorial

Proposals and draft submission

In addition to the final tutorial, you will have a one-sentence proposal submitted, as well as a mid-way tutorial draft, that should be roughly half the length of the final submission. Both of these are intended largely as check-ins, and will factor only slightly into the final grade. Students will receive full credit as long as they make a reasonable attempt to propose a topic and provide an initial draft of their tutorial (where the definition of "reasonable attempt" is ultimately up to the instructors' discretion). Must have at least 500 words of text and 50 lines of code in checkin submission

Grading

Grading will be done on a point scale, (0-100), and will be factor in the different elements as follows:

- 3% - tutorial proposal
- 7% - tutorial draft submission
- 10% - providing feedback on other students tutorials (we are not evaluating the feedback except to verify it meets the criteria of the rubric).
- 20% - student evaluation: peer evaluating of your tutorial, averaged over the two student evaluators of your tutorial
- 60% - instructor evaluation: evaluation of your tutorial by the instructors according to the rubric above.