Problem Set 0: Submission & Typesetting

Due: Friday, January 23, 2026 at 11:59pm ET

Instructions: This first "problem set" is simply a calibration problem set to get you acquainted with the resources, policies, and syllabus of the course and help you decide what typesetting software to use. **This problem set is worth zero points, but please submit it just to make sure Gradescope and everything is up and running for you.**

Course Resources (0 Points)

Welcome to the Spring 2026 semester of DS-GA 1003: Machine Learning! This problem is just here so you know where to find everything for the course. We won't be using the Brightspace webpage for this course; instead, all resources, links, and materials will be posted on the course webpage. Hop on over there and complete the following tasks:

1. Browse the course webpage for the course and familiarize yourself with course policies.
2. Know where to ask for help: (i) Ed to post questions to your classmates, the instructor, and the TA(s) and (ii) office hours, updated on the Calendar on the course webpage.
3. All lecture materials and optional readings will be posted in Course Content on the course webpage as the class goes on.
4. Make sure you're signed up for Gradescope, where all homework will be submitted.
5. Make sure you're signed up for Ed, where announcements, class discussion, and questions on the homework will be posted.

For all queries related to the course, post on Ed for the fastest response — this discussion board will be monitored by the instructors and course staff. There is also an option to post on Ed privately, which will make your question visible only to the course staff and instructors.

Ed should be your first stop, but if you believe that your question is a private query that should only be directed to the instructors, please email both of us at samuel.deng@nyu.edu and n.tomlin@nyu.edu.

In your submission for this problem, just write: "I have looked through the course resources." The following "problem" will give details on recommended ways you could typeset your assignments.

Typesetting (0 Points)

In subsequent problem sets, the following instructions will begin every problem set, along with some problem set specific instructions.

> **Instructions:** Your answers to the questions below, including plots and mathematical work, should be submitted as a single PDF file. It's preferred that you write your answers using software that typesets mathematics (e.g. LaTeX or MathJax in iPython), though if you need to you may scan handwritten work. You may find the minted package convenient for including source code in your LaTeX document.

In most problem sets, you will need to submit code for the programming portions. For the programming part of the assignments, you should start getting comfortable with the Python packages NumPy and matplotlib if you are not already.

In order to typeset your problem sets, we recommend one of two methods:

LaTeX

LaTeX is by and large the standard solution to typesetting mathematics. You are welcome to generate your LaTeX submissions however you'd like; we only need your PDF. Here's a quick resource to get you up to speed for typesetting in LaTeX if you've never used it before. A standard way to include Python code in your PDF generated by LaTeX is minted.

You should check the source file `ps0.tex` in the .zip for an example of LaTeXtypesetting. The code for the following two examples of using `minted` is included in `ps0.tex`.

Including Python Code from File Here we're extracting lines 4 through 13 from the file `code.py`.

```
def dotProduct(d1, d2):
    """
    @param dict d1: a feature vector represented by a mapping from a
    ↪  feature (string) to a weight (float).
    @param dict d2: same as d1
    @return float: the dot product between d1 and d2
    """
    if len(d1) < len(d2):
        return dotProduct(d2, d1)
    else:
        return sum(d1.get(f, 0) * v for f, v in d2.items())
```

Python Code Inline Here we're extracting lines 4 through 13 from the file `code.py`.

```
def dotProduct(d1, d2):
    """
    @param dict d1: a feature vector represented by a mapping from a
    ↪  feature (string) to a weight (float).
    @param dict d2: same as d1
    @return float: the dot product between d1 and d2
    """
    if len(d1) < len(d2):
        return dotProduct(d2, d1)
    else:
        return sum(d1.get(f, 0) * v for f, v in d2.items())
```

iPython/Jupyter notebook

Alternatively, you can submit your assignments by typesetting everything in one `.ipynb` iPython/Jupyter notebook. In this case, you'd be using Markdown and MathJax to typeset your math and you could write Python code directly in the notebook code cells. Please see the `ps0_jupyter.ipynb` file in the .zip for a sample. When submitting in this way, please make sure you clean up your Jupyter notebook before converting to PDF to only include outputs of cells that are relevant to the assignment.

Submission

For your submission for this problem, simply submit lines 15-28 of `code.py` in this .zip (the function named `increment`) using either of these methods.

Midterm (0 Points)

As will be stated on the first lecture in class, 35% of your grade is the in-class midterm during the usual lecture slot on **Tuesday, March 10, 2026 2:45pm - 4:45pm ET.** Please make sure you are available in-person on this day!

Due to resource constraints, we will not be able to allow make-up midterms for those who cannot attend lecture on that day, so missing the midterm will result in a zero for the midterm. If you must miss the midterm due to an unexpected emergency or extenuating circumstance (and do not want to continue the course with a grade of zero on the midterm), you should bring this up with the instructors as soon as possible so we can possibly arrange assigning you a grade of "Incomplete."

For your submission for this problem, just write: "I acknowledge the midterm policy for this class and take responsibility for my attendance on the midterm date."

Prerequisite Questionnaire (0 Points)

In the `prereqs.pdf` document, there is a questionnaire that you can fill out to gauge for yourself your readiness for the class. We have just included it so you can self-diagnose any spots you might want to brush up on as the class begins. **No need to turn this questionnaire in!**

The Prerequisites section of the course website has some references to help you review some of the mathematical prerequisites for this course. If you have any questions about other ways you can brush up, feel free to post on Ed!

Basic understanding of "black box" machine learning. A prerequisite for this class is DS-GA 1001 (Introduction to Data Science). Because of this, we will assume that students have already been exposed to the basic pipeline and workflow of machine learning at a "black box" level (without paying attention to the inner workings of any particular algorithm or method). This includes concepts such as:

- Training data, test data, train-test splits.
- Features and labels; the difference between unlabeled data and labeled data.
- The phenomenon of overfitting.

The first lecture will hit the ground running assuming that students have already been exposed to these concepts, so we will provide some resources in the first week so you can brush up on these basic prerequisite ideas if they are unfamiliar to you.

For your submission for this problem, just write: "I have looked at the prerequisite questionnaire."