Assignment Instructions

IDS:705 Principles of Machine Learning

Table of contents

Assignment Grading

Assignment grading will consist of two components: content (90%) and presentation (10%). The content grade will be based on the accuracy and completeness (including the depth) of your answers to each question. If a question asks you to explain, hypothesize, or otherwise think critically, be sure to demonstrate your critical engagement with the question. Presentation and collaboration are critical skills for success as a data scientist. To support you in refining those skills, we include the presentation component of the grade. The presentation score will be based on how well you communicate in writing, figure creation, and coding. Clear writing (with appropriate grammar and spelling), organized answers, well-organized/commented code, and properly formatted figures will ensure your success on the presentation component of the assignment.

How to submit assignments

Assignments will be submitted via Gradescope as a PDF and you will need to indicate the location of your work for each question and subquestion. Below is a step-by-step guide to submitting your assignments.

1. **Header**. At the top of the document, include the assignment number, your name, your netid, and a list of anyone you collaborated with on the assignment.
2. **Each answer is prefaced in bold by the question/subquestion number**. Each question is followed by one or more sections typically labeled with section numbers (e.g. 2.3). Make sure you have a cell with that section number prior to your response. You may use multiple cells for each answer that include both markdown and code. For full credit, submitted assignments should be easily navigable with answers for subsections clearly indicated.

3. **Ensure that all cells have been run**. Cells that have not been run will be treated as unanswered questions and assigned zero points.
4. **Create a PDF document of your notebook**. There are a few ways to do this. Please see the guide below.
5. **Your content is legible prior to submission**. Look over your PDF before you submit it. If we cannot read it, or parts are missing, we cannot grade it, and no credit will be given for anything we cannot read or is missing.

    - Code is valid (able to run and producing the correct answer), neat, understandable, and well commented.
    - Math is either clearly written and inserted into the proper part of the document or typeset using LateX equations.
    - All text that is not code should be formatted using markdown. Two references to help include: ref1 and ref2.

6. **Submit your assignment by the deadline on gradescope**. Any assignments received after the deadline will be treated as late. Please see this video for how to submit your assignment on gradescope. The submission link is here: https://www.gradescope.com/
7. **Assign the respective pages to each question**. When you submit your assignment on Gradescope, you will be asked to assign pages of your PDF to each question accordingly. Be sure to leave time to complete this before submission. Please remember, if we cannot find your answer, we cannot grade it.

Example Question and Template

To demonstrate how to organize your responses using a Jupyter notebook please review the following:

- Sample assignment questions
- Sample assignment template and example for submission

Entering mathematical equations

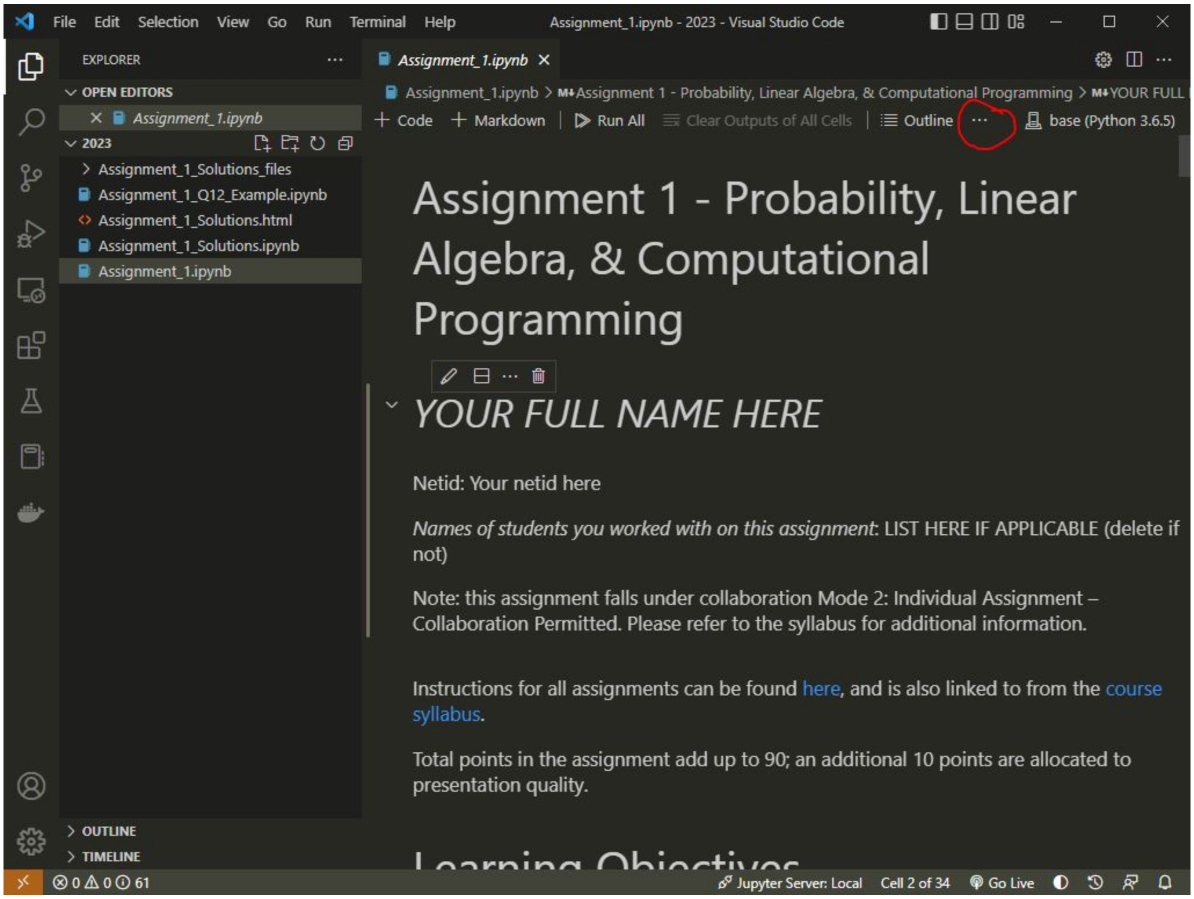
You may either write out equations by hand or using markdown and LaTeX (LaTeX is recommended). Either way, I recommend that you complete the work on paper before typing up the final version if you choose to use LaTeX. If you hand-write your math, please digitize them (scan them in or take a picture) and place them in the proper order for of the document for your final PDF. Either way, show your math including any intermediate steps necessary to understand the logic of your solution. If we are not able to interpret your meaning (or understand your writing), no credit will be given.

Rendering a PDF file

Option 1: Render from VS Code (without additional extensions)

We recommend using VS Code for this course and for use in developing your Jupyter notebooks. Rendering pdfs from Jupyter Notebooks in VS Code has a couple of extra steps, though. To render a pdf without any additional extensions, you do the following:

1. Open your notebook in VS code and hit the ellipsis (the three horizontal dots) just to the right of the tools for your notebook:


File
Edit
Selection
View
Go
Run
Terminal
Help
Assignment_1.ipynb - 2023 - Visual Studio Code
EXPLORER
OPEN EDITORS
Assignment_1.ipynb
2023
Assignment_1_Solutions_files
Assignment_1_Q12_Example.ipynb
Assignment_1_Solutions.html
Assignment_1_Solutions.ipynb
Assignment_1.ipynb
OUTLINE
TIMELINE
Assignment_1.ipynb > M↓Assignment 1 - Probability, Linear Algebra, & Computational Programming > M↓YOUR FULL
+ Code
+ Markdown
Run All
Clear Outputs of All Cells
Outline
···
base (Python 3.6.5)
⊗ 0 ⚠ 0 ⓘ 61
Jupyter Server: Local
Cell 2 of 34
Go Live

Assignment 1 - Probability, Linear Algebra, & Computational Programming

YOUR FULL NAME HERE

Netid: Your netid here

Names of students you worked with on this assignment: LIST HERE IF APPLICABLE (delete if not)

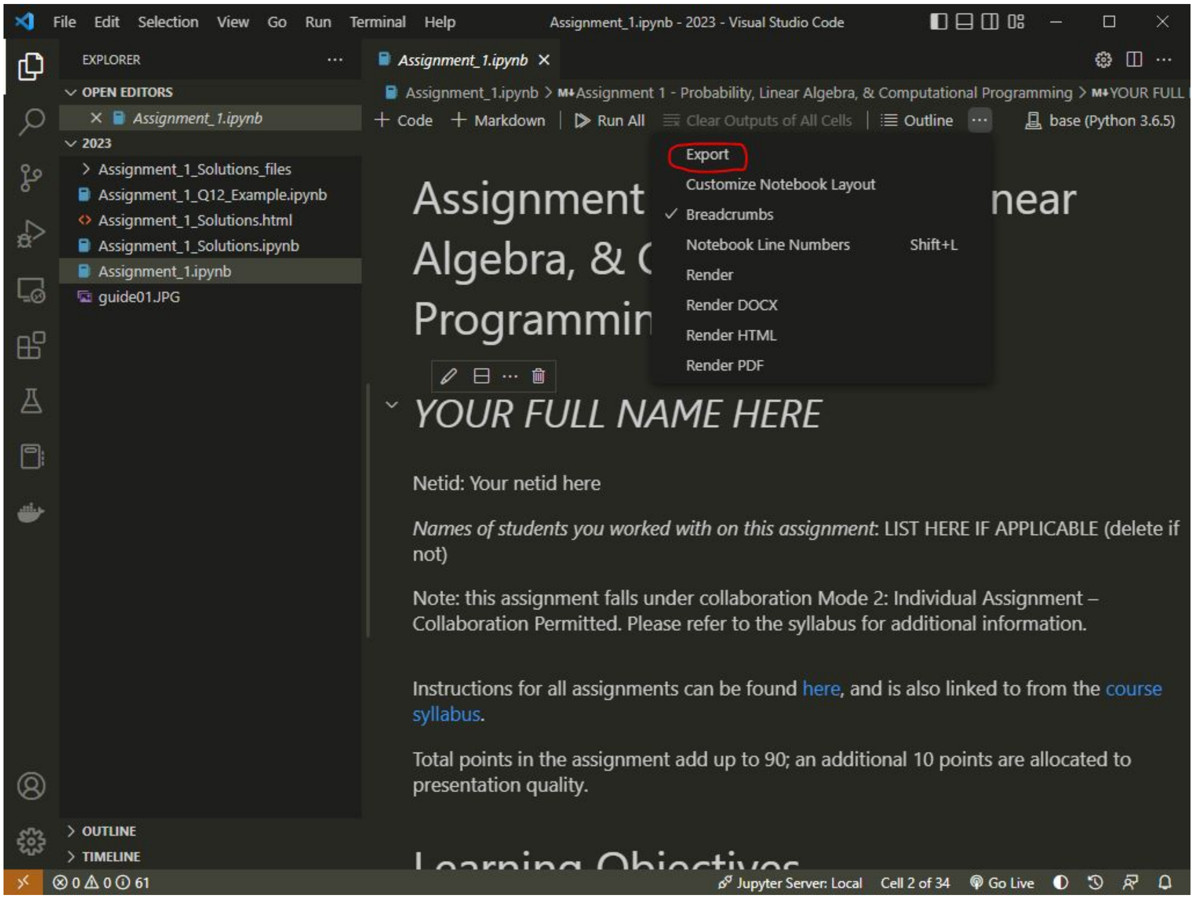
Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus for additional information.

Instructions for all assignments can be found here, and is also linked to from the course syllabus.

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

Learning Objectives

2. Click “Export”:


File Edit Selection View Go Run Terminal Help
Assignment_1.ipynb - 2023 - Visual Studio Code
EXPLORER
OPEN EDITORS
Assignment_1.ipynb
2023
Assignment_1_Solutions_files
Assignment_1_Q12_Example.ipynb
Assignment_1_Solutions.html
Assignment_1_Solutions.ipynb
Assignment_1.ipynb
guide01.JPG
OUTLINE
TIMELINE
Assignment_1.ipynb
Assignment_1.ipynb > M↓Assignment 1 - Probability, Linear Algebra, & Computational Programming > M↓YOUR FULL
+ Code + Markdown Run All Clear Outputs of All Cells Outline
base (Python 3.6.5)
Export
Customize Notebook Layout
✓ Breadcrumbs
Notebook Line Numbers Shift+L
Render
Render DOCX
Render HTML
Render PDF
Assignment near
Algebra, & C
Programmin
Learning Objectives
⊗ 0 ⚠ 0 ⓘ 61
Jupyter Server: Local
Cell 2 of 34
Go Live

YOUR FULL NAME HERE

Netid: Your netid here

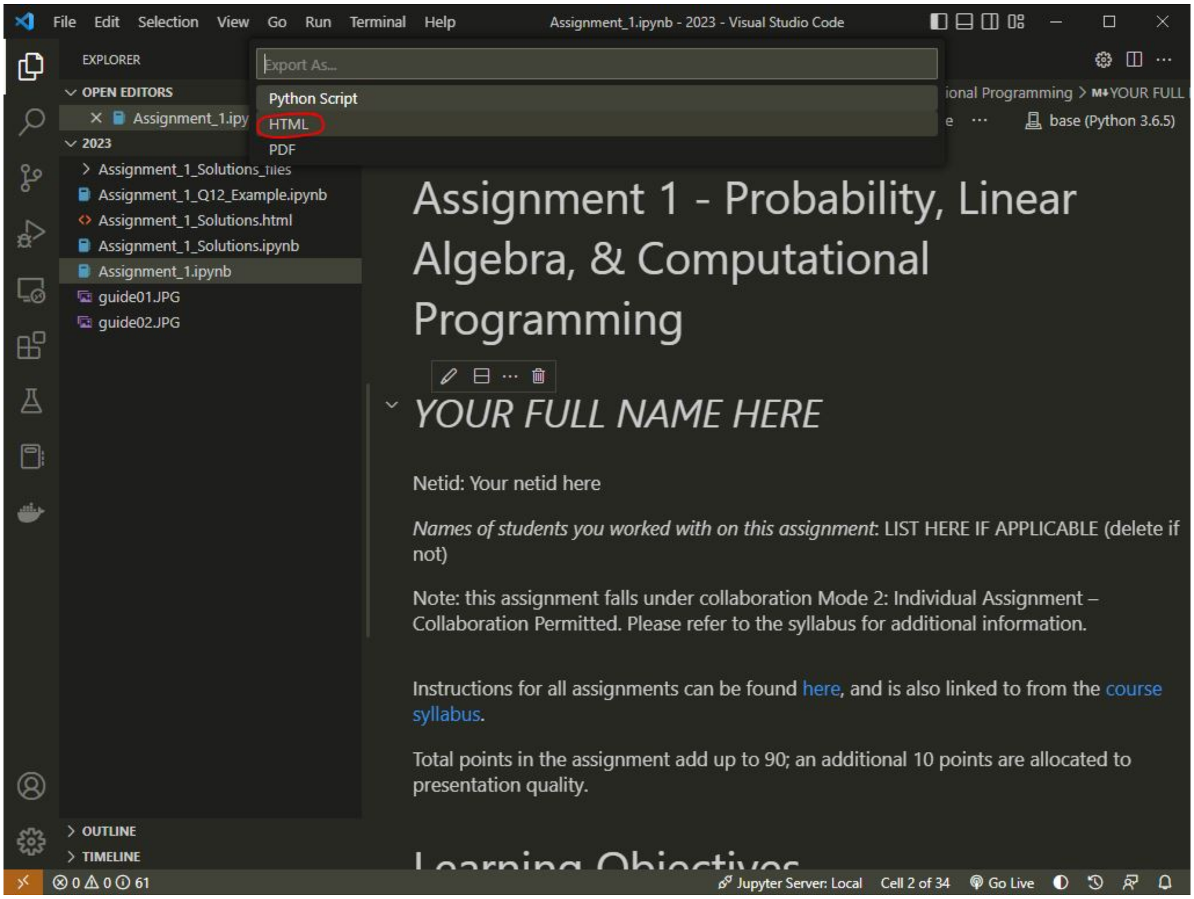
Names of students you worked with on this assignment: LIST HERE IF APPLICABLE (delete if not)

Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus for additional information.

Instructions for all assignments can be found here, and is also linked to from the course syllabus.

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

3. Select “HTML” (exporting to pdf won’t work without the installation of additional tools):



Assignment 1 - Probability, Linear Algebra, & Computational Programming

YOUR FULL NAME HERE

Netid: Your netid here

Names of students you worked with on this assignment: LIST HERE IF APPLICABLE (delete if not)

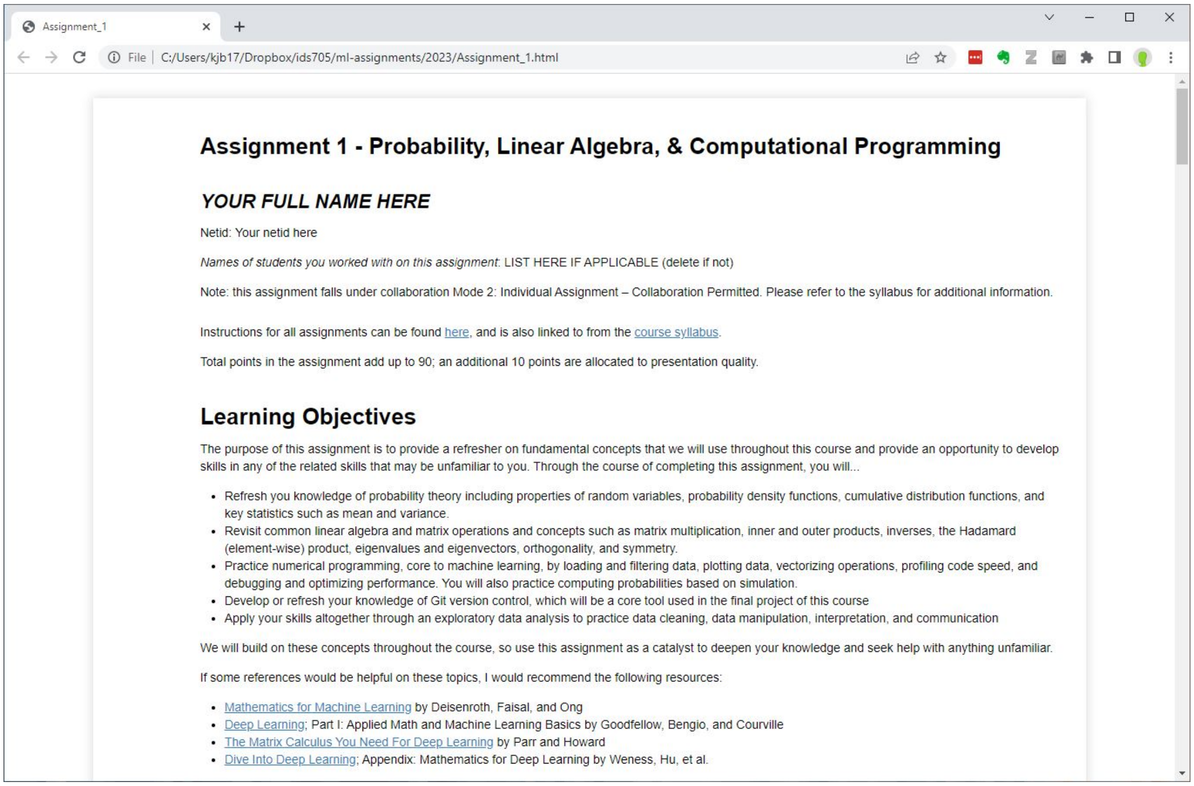
Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus for additional information.

Instructions for all assignments can be found here, and is also linked to from the course syllabus.

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

Learning Objectives

4. Open the HTML file you just saved in Chrome:


Assignment_1
File
C:/Users/kjb17/Dropbox/ids705/ml-assignments/2023/Assignment_1.html

Assignment 1 - Probability, Linear Algebra, & Computational Programming

YOUR FULL NAME HERE

Netid: Your netid here

Names of students you worked with on this assignment: LIST HERE IF APPLICABLE (delete if not)

Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus for additional information.

Instructions for all assignments can be found here, and is also linked to from the course syllabus.

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

Learning Objectives

The purpose of this assignment is to provide a refresher on fundamental concepts that we will use throughout this course and provide an opportunity to develop skills in any of the related skills that may be unfamiliar to you. Through the course of completing this assignment, you will...

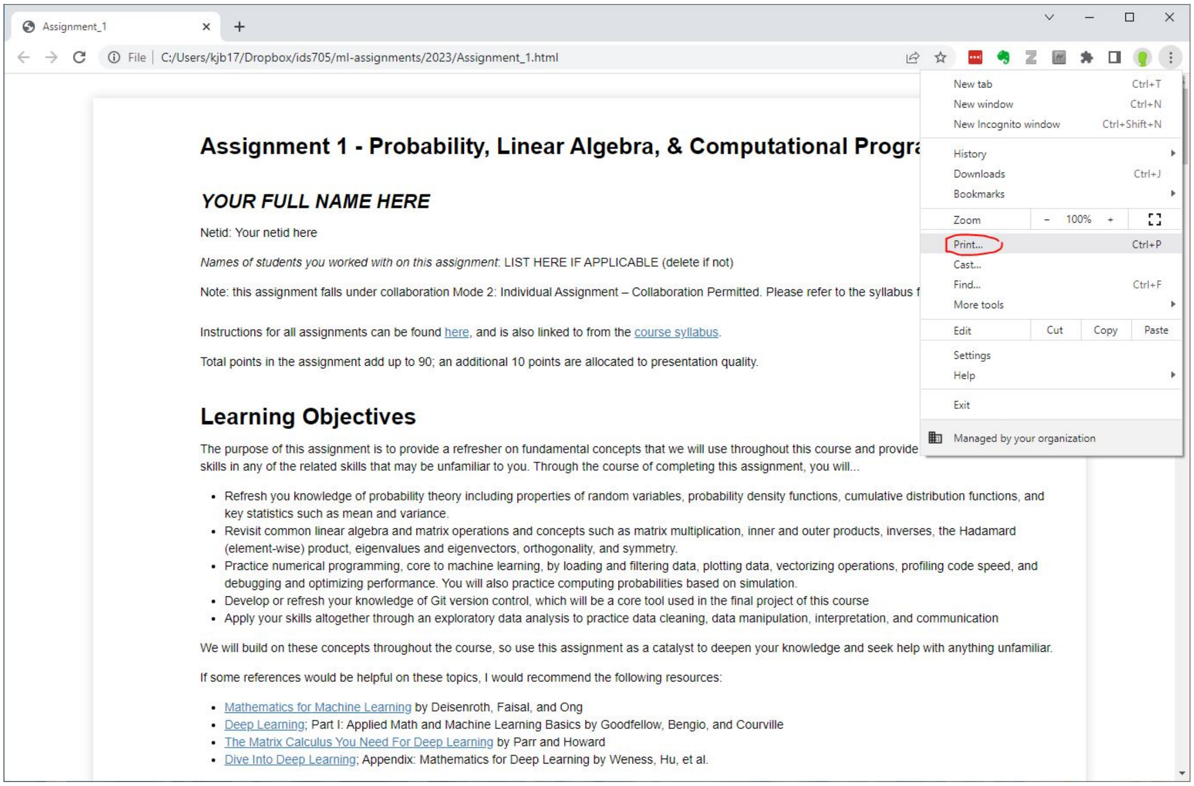
- Refresh you knowledge of probability theory including properties of random variables, probability density functions, cumulative distribution functions, and key statistics such as mean and variance.
- Revisit common linear algebra and matrix operations and concepts such as matrix multiplication, inner and outer products, inverses, the Hadamard (element-wise) product, eigenvalues and eigenvectors, orthogonality, and symmetry.
- Practice numerical programming, core to machine learning, by loading and filtering data, plotting data, vectorizing operations, profiling code speed, and debugging and optimizing performance. You will also practice computing probabilities based on simulation.
- Develop or refresh your knowledge of Git version control, which will be a core tool used in the final project of this course
- Apply your skills altogether through an exploratory data analysis to practice data cleaning, data manipulation, interpretation, and communication

We will build on these concepts throughout the course, so use this assignment as a catalyst to deepen your knowledge and seek help with anything unfamiliar.

If some references would be helpful on these topics, I would recommend the following resources:

- Mathematics for Machine Learning by Deisenroth, Faisal, and Ong
- Deep Learning; Part I: Applied Math and Machine Learning Basics by Goodfellow, Bengio, and Courville
- The Matrix Calculus You Need For Deep Learning by Parr and Howard
- Dive Into Deep Learning; Appendix: Mathematics for Deep Learning by Weness, Hu, et al.

5. Click the options button (three vertical dots) and click “print”:



Assignment 1 - Probability, Linear Algebra, & Computational Progra

YOUR FULL NAME HERE

Netid: Your netid here

Names of students you worked with on this assignment: LIST HERE IF APPLICABLE (delete if not)

Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus f

Instructions for all assignments can be found here, and is also linked to from the course syllabus.

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

Learning Objectives

The purpose of this assignment is to provide a refresher on fundamental concepts that we will use throughout this course and provide skills in any of the related skills that may be unfamiliar to you. Through the course of completing this assignment, you will...

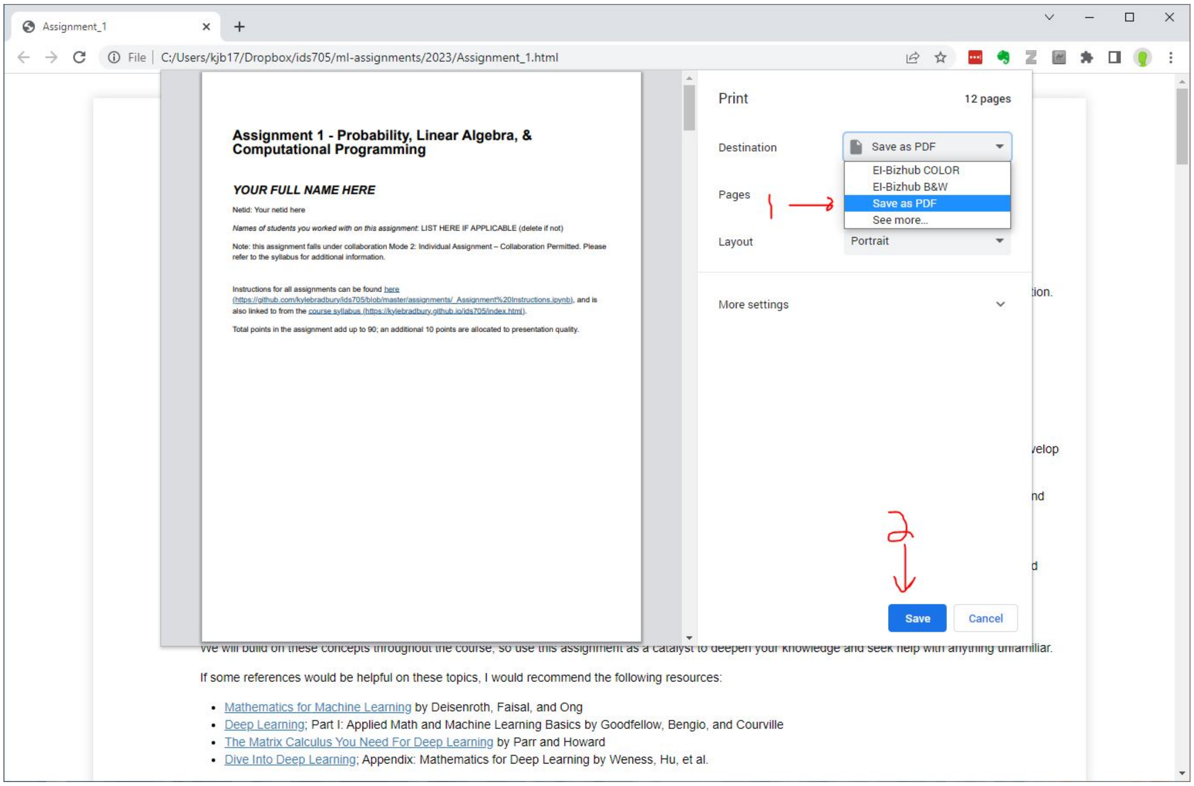
- Refresh you knowledge of probability theory including properties of random variables, probability density functions, cumulative distribution functions, and key statistics such as mean and variance.
- Revisit common linear algebra and matrix operations and concepts such as matrix multiplication, inner and outer products, inverses, the Hadamard (element-wise) product, eigenvalues and eigenvectors, orthogonality, and symmetry.
- Practice numerical programming, core to machine learning, by loading and filtering data, plotting data, vectorizing operations, profiling code speed, and debugging and optimizing performance. You will also practice computing probabilities based on simulation.
- Develop or refresh your knowledge of Git version control, which will be a core tool used in the final project of this course
- Apply your skills altogether through an exploratory data analysis to practice data cleaning, data manipulation, interpretation, and communication

We will build on these concepts throughout the course, so use this assignment as a catalyst to deepen your knowledge and seek help with anything unfamiliar.

If some references would be helpful on these topics, I would recommend the following resources:

- Mathematics for Machine Learning by Deisenroth, Faisal, and Ong
- Deep Learning; Part I: Applied Math and Machine Learning Basics by Goodfellow, Bengio, and Courville
- The Matrix Calculus You Need For Deep Learning by Parr and Howard
- Dive Into Deep Learning; Appendix: Mathematics for Deep Learning by Weness, Hu, et al.

6. Select the destination as “Save as PDF” and click “Save”:



Assignment 1 - Probability, Linear Algebra, & Computational Programming

YOUR FULL NAME HERE

Netid: Your netid here

Names of students you worked with on this assignment: LIST HERE IF APPLICABLE (delete if not)

Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus for additional information.

Instructions for all assignments can be found here (https://github.com/kylebradbury/ids705/blob/master/assignments/_Assignment%20Instructions.ipynb), and is also linked to from the course syllabus (https://kylebradbury.github.io/ids705/index.html).

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

We will build on these concepts throughout the course, so use this assignment as a catalyst to deepen your knowledge and seek help with anything unfamiliar.

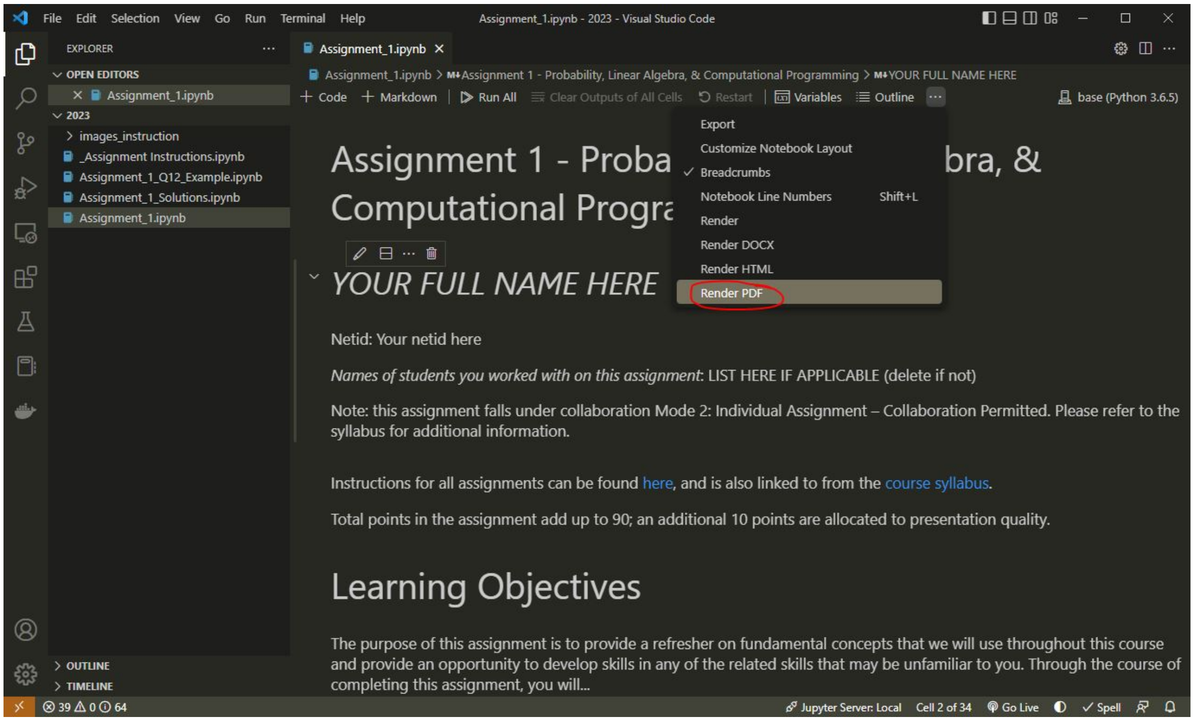
If some references would be helpful on these topics, I would recommend the following resources:

- Mathematics for Machine Learning by Deisenroth, Faisal, and Ong
- Deep Learning; Part I: Applied Math and Machine Learning Basics by Goodfellow, Bengio, and Courville
- The Matrix Calculus You Need For Deep Learning by Parr and Howard
- Dive Into Deep Learning; Appendix: Mathematics for Deep Learning by Weness, Hu, et al.

7. Voila! You should have a pdf

Option 2: Render from VS Code with Quarto and LaTeX

You can save a lot of the above steps if you're willing to install 2 things: 1. Install Quarto, which is an open-source scientific and technical publishing system (great for making websites and blogs for your professional portfolio). 2. Install the Quarto VS Code extension 3. Install a version of tex for your operating system. 4. Once you do this, you can directly export your documents to pdf:


File Edit Selection View Go Run Terminal Help
Assignment_1.ipynb - 2023 - Visual Studio Code
EXPLORER
OPEN EDITORS
Assignment_1.ipynb
2023
images_instruction
_Assignment Instructions.ipynb
Assignment_1_Q12_Example.ipynb
Assignment_1_Solutions.ipynb
Assignment_1.ipynb
OUTLINE
TIMELINE
Assignment_1.ipynb
Assignment_1.ipynb > M↓Assignment 1 - Probability, Linear Algebra, & Computational Programming > M↓YOUR FULL NAME HERE
+ Code
+ Markdown
Run All
Clear Outputs of All Cells
Restart
Variables
Outline
base (Python 3.6.5)
Export
Customize Notebook Layout
✓ Breadcrumbs
Notebook Line Numbers Shift+L
Render
Render DOCX
Render HTML
Render PDF
⊗ 39 ⚠ 0 ⓘ 64
Jupyter Server: Local
Cell 2 of 34
Go Live
✓ Spell

Assignment 1 - Proba bra, &
Computational Progra

YOUR FULL NAME HERE

Netid: Your netid here

Names of students you worked with on this assignment: LIST HERE IF APPLICABLE (delete if not)

Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus for additional information.

Instructions for all assignments can be found here, and is also linked to from the course syllabus.

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

Learning Objectives

The purpose of this assignment is to provide a refresher on fundamental concepts that we will use throughout this course and provide an opportunity to develop skills in any of the related skills that may be unfamiliar to you. Through the course of completing this assignment, you will...

Option 3: Render from Jupyter Notebooks

Open your notebook in a Jupyter Notebook in Google Chrome. Go to File->Print Preview, then after verifying the document looks correct, click "print" and for your printer choose "Save as PDF."

Whichever method you choose - always check your pdf before submitting to make sure everything is rendered and nothing is cut off!

Figure Guidelines

Here is an example of a well-prepared figure that checks all of the requirements for good figures. Please check out this Coursera course for skill development and Python plotting best practices.

Figure checklist

1. All plots should have a purpose - either being directly requested or should make a clear point.
2. **All** plots should have axes labels and legible fonts (large enough to read).

3. Legends are used when there are multiple series plotted on a single plot.
4. Markers on plots are properly sized (e.g. not be too big nor too small) and/or have the appropriate level of transparency to be able to clearly read the data without obscuring other data.
5. If there are multiple colors used on a plot, EVERY color can be distinguished from the rest.
6. Figures are crisp and clear - there are no blurry figures.
7. Plots should be explained clearly in text or have a caption explaining them

To demonstrate we'll start by loading some data to plot (the example here is from the lecture on the bias-variance tradeoff):

```
import numpy as np
import matplotlib.pyplot as plt
import pickle

# Load the data for plotting
datafilename = './plot_example_data/data.pkl'
infile = open(datafilename,'rb')
loaded_data = pickle.load(infile)
infile.close()

# Store the loaded data in convenient variable names
kvalues = loaded_data['kvalues']
error_training = loaded_data['error_training']
error_testing = loaded_data['error_testing']
error_bayesclf = loaded_data['error_bayesclf']

# Choose colors that will be distinguishable from one another
color0 = '#121619' # Dark grey
color1 = '#00B050' # Green
color2 = '#7c7c7c' # Light grey
```

Then, we create the plot:

```
%config InlineBackend.figure_format = 'retina' # Make clear on high-res screens

# Create the plot
fig, ax = plt.subplots(figsize=(7,5), dpi= 100) # Adjust the figure size and dots per inch to
ax.semilogx(kvalues,error_training,
            color=color0,
            label='Training (in-sample)')
```

```
ax.semilogx(kvalues,error_testing,
            color=color1,
            label='Test (out-of-sample)')
ax.semilogx(kvalues,error_bayesclf,'--',
            color=color2,
            label='Bayes (optimal)')
ax.legend()
ax.grid('on')
ax.set_xlabel('k nearest neighbors') # Always use X and Y labels
ax.set_ylabel('Binary Classification Error Rate')
ax.set_xlim([1,200]) # Ensure the axis is the right size for the plot data
ax.set_ylim([0,0.5])
fig.tight_layout() # Use this to maximize the use of space in the figure
plt.show()
```

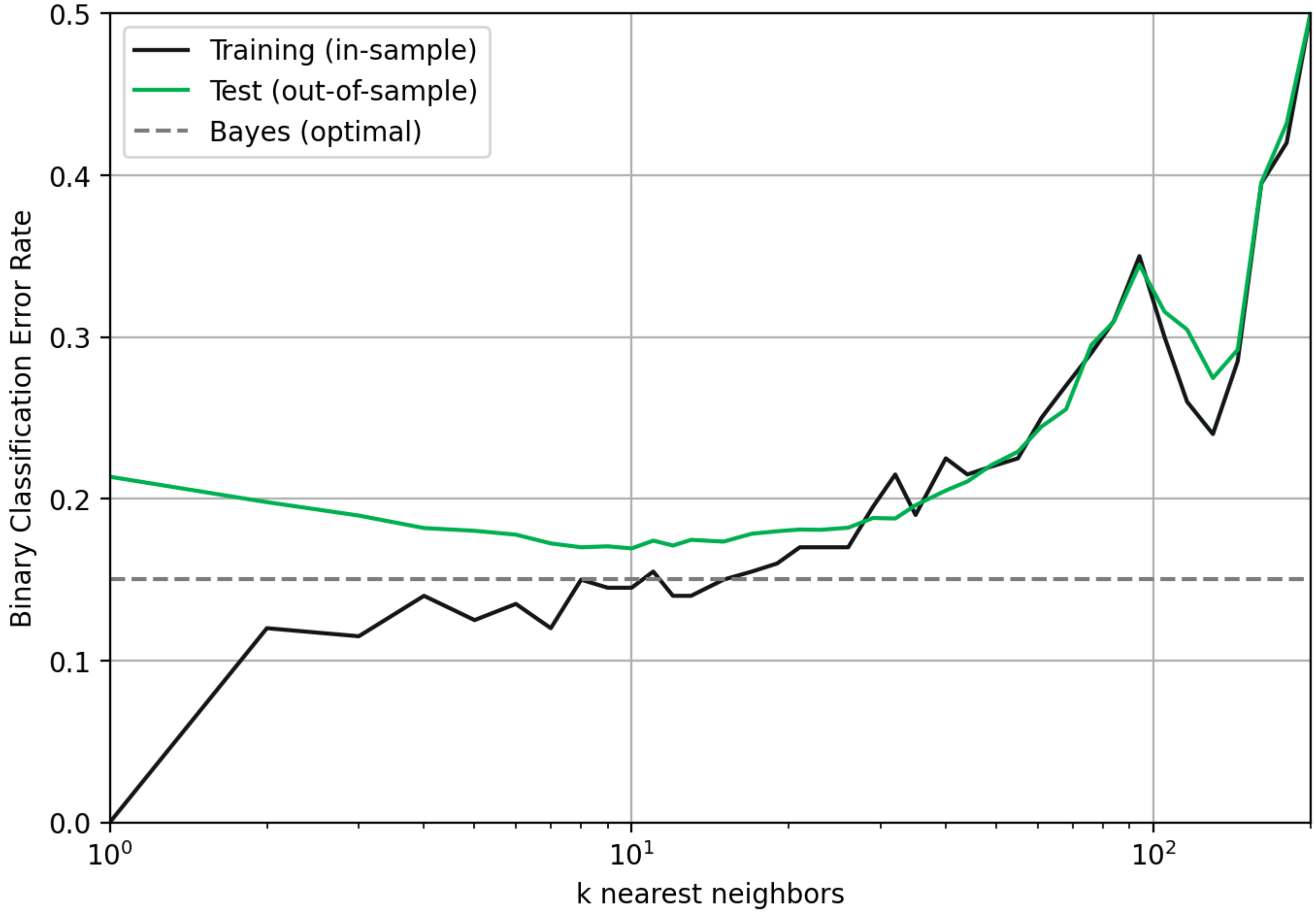


Figure 1. Ideally, each figure should have a caption that explains the figure. Here, the test data (green) approximates the generalization error rate, the lower-bound of which is the Bayes error rate (grey dotted line). The value of k represents the flexibility of the model with lower values of k representing higher model flexibility and higher values of k, lower flexibility. The training error (black) is not considered out of sample, so in cases of high model flexiblity (and in this case high overfit), the training error rate can reach zero.

Note the clear x and y axis labels and legends - there are no acronyms or shorthands used, just the full description of what is being plotted. Also note how easy it is to compare the color of lines and identify the baseline (Bayes) comparison line. Make your plots easy to read and understand. A figure caption is generally helpful as well to indicate what we are seeing in each plot.

Sources

Some questions on the assignments are adapted from sources including:
1. James et al., *An Introduction to Statistical Learning*
2. Abu-Mostafa, Yaser, *Learning from Data*
3. Weinberger, Kilian, Machine Learning CS4780, Cornell University