CDS DS 110: Introduction to Data Science with Python

Instructor Name: Dr. Kevin Gold

Office Location: CCDS 1406

Contact Information: klgold@bu.edu

Drop-in Office Hours: MW 3-4:30pm

Course Dates: 1/18-5/10

Course Time & Loc: MWF 12:20-1:10pm, CGS505
and MWF 1:25-2:15PM, CGS505

Course Credits: 4

TAs:
Harsh Sharma: hsharma@bu.edu
Asma Furniturewala: asmaf@bu.edu
Harsh Khatri: harsh242@bu.edu
Shubh Gupta: shubhg7@bu.edu

Graders and CAs:
Jason Chen (Office Hours): jachen7@bu.edu
Shreyas Sudarsan Venkatadrhi: shreyas9@bu.edu
Pranjal Ekhande: pekhande@bu.edu
Keval Patel: keeyval@bu.edu
Xiang Fu: xfu@bu.edu
Urja Damodhar: urjadd@bu.edu

Course Description:

DS 110 is the first in a two-course sequence (including DS 210) that builds students' competence in computing techniques central to data science.

In DS 110, students will use Python to explore many of the fundamental computer science concepts and processes used in data science. This begins with core computer science topics such as data structures, development of functions and recursion, and leads to topics including object oriented programming and data visualization.

In terms of Python tools for data science, students will learn how to work with numpy, pandas, and matplotlib to analyze real-world data. (See the more comprehensive list of topics below in this syllabus.)

The sequence of DS 110 and 210 works in concert with the 120-121-122 series: as students develop their expertise in the mathematical foundations covered in 120-121-122, they build their abilities to implement those tools and manage actual datasets in the 110-120 sequence.

Hub Learning Outcomes

Quantitative Reasoning I (QR1) Learning Outcomes:

Students will be introduced to the Python programming language. Students will learn how to use pure Python to write basic functional and object-oriented programs. Students will also learn about core data science libraries such as numpy and pandas and, by the end of the course, students will be able to use these tools to produce descriptive analyses and visualizations of different datasets.

In parallel to learning Python for data science, students will develop and demonstrate their understanding of core concepts in computer science, and their ability to use them to manipulate different types of data. Students will learn about basic CS topics such as algorithms, big-O notation and algorithm runtime, object oriented programming methods, and parallel computing.

Students will master the fundamentals of programming in Python for data science, including learning how to use common libraries such numpy, pandas, and matplotlib. Using these tools, students will build basic models of the data to uncover and interpret interesting patterns in real-world data.

Students will learn to communicate the findings of their analyses both verbally and as part of a presentation using summary statistics, tabulations, and visualizations.

Teamwork/Collaboration Learning Outcomes:

Students will form teams of 3-4 to complete a half-semester-long project. Students will be provided with feedback and guidance by the course instructor.

Instructional Format, Course Pedagogy, and Approach to Learning

The course will be primarily lecture-based, but with additional exercises for the audience designed to engage and check understanding. Homework will play a key role in moving from hypothetical concepts to applied problems. The discussion section will serve as additional review. The final project will be a chance to let the students guide their own learning toward a personally meaningful goal.

Because of the prevalence of LLMs and other online resources that may mask gaps in a student's knowledge, understanding, and skills, two in-class midterms will be administered in which other resources are not allowed.

Books and Other Course Materials

Recommended Text: Deitel & Deitel, *Intro to Python for Computer Science and Data Science*, ISBN-13 978-0135404676

Recommended readings by day are in the schedule at the end of this syllabus. This book can be helpful in providing an alternate take on the material, as well as in providing practice problems to help students prepare for the midterms. But, it isn't required.

Recommended Hardware: You need a basic laptop that has a web browser for this course. If you don't own one, you can borrow one from BU. The heavy lifting of running programs will happen in the cloud, with very little done locally on your machine unless you prefer to work that way (see Anaconda below).

Courseware

We'll be using Blackboard for distributing assignments, collecting assignments, grading, and the distribution of course notes: https://learn.bu.edu

We will mostly be using Google Colab for assignments and course notes: colab.research.google.com/

Students who wish to work offline could download and install Anaconda: https://www.anaconda.com/

We'll use Piazza for Q&A: https://piazza.com/bu/spring2024/ds110/

Assignments and Grading

Course assignments consist of weekly individual programming assignments, and a half-semester long group project.

The individual assignments are meant to teach programming skills and other concepts. Help is available in the form of TA office hours as well as a "bot" that uses GPT-4 to answer questions. Students should try to do the work themselves when they can, because they will be tested on their abilities in the in-class exams, but these resources can easily get students unstuck if they aren't making progress.

In general, assignments are about all the topics that were covered in lecture since the last assignment - see the day-by-day topic list at the end of the syllabus for more details. All the assignments are weighted equally, even if maximum point values differ.

The team project is student-driven and consists of two parts: in the first part students work in teams of 3-4 to form an interesting data-science question that can be answered through publicly available data, and descriptive analysis using Python. The students submit their proposal in the middle of the semester. The students incorporate any feedback from the instructor, and upon approval of their proposal begin work on their project. As part of their assessment, students will produce peer evaluations of the others in their group, as well as reflections on what they could do better. During the final week of the semester, the students present their projects in "lightning talks" and submit papers.

There will be two in-class exams during the semester - one at roughly the halfway point that emphasizes having learned to program, and another near the end that is cumulative but which

emphasizes topics covered since the last exam. These will be mostly multiple choice, but each will also involve coding a small function using paper and pencil. These exams are meant to encourage learning all the material rather than just what is emphasized on the homework, and also to discourage using generative AI in any way that impedes the learning process. Students should therefore practice on their own until they can produce basic programs quickly and without external aids.

Grading:

- Midterm exam 1 (15%)
- Midterm exam 2 (20%)
- Individual assignments (30%, lowest dropped)
- Group project (30%)
- Recitation participation (5%)

Resources/Support/How to Succeed in This Course:

1. To succeed in this course students should attend all lectures, come to discussion sections prepared with questions, complete all assignments on time, and discuss problems and material with fellow classmates.
2. Students are welcomed and encouraged to visit office hours to ask questions about the material covered in class, the homework, or any relevant topics.
3. The Education Resource Center offers free individual and group tutoring.
4. Accommodations for Students with Documented Disabilities: If you are a student with a disability or believe you might have a disability that requires accommodations, please contact the Office for Disability Services (ODS) at (617) 353-3658 or access@bu.edu to coordinate any reasonable accommodation requests. ODS is located at 25 Buick Street on the 3rd floor. Please give me notice of any special needs at least 2 weeks before an exam that requires special accommodations, so that we can schedule space if necessary.

Community of Learning: Class and University Policies

1. **Courtesy expectations.** Students are responsible for supporting a courteous learning environment. Please show respect for other students' questions, and maintain an attentive attitude in class. Please participate in the in-class exercises and activities. Using screens for activities other than following lecture is discouraged.

2. **Attendance & Absences.** Attending lecture is optional but highly encouraged. Attending discussion sections carries some participation credit. However, it can be waived with a reasonable request for accommodation. We will in particular waive for religious holidays; see the University Policy on Religious Observance.

3. **Assignment Completion & Late Work**. Assignments will be submitted as PDF and .ipynb files on Blackboard. Students have 5 late days which can be used over the course of the semester, no more than 2 per assignment, which allow work to be turned in late with no penalty. Using 1 late day extends the deadline 24 hours for that student;

late days can't be used in fractions. Late days can't be used on final project deliverables.

4. **Academic Conduct Statement**

   Students are expected to abide by the guidelines and rules of the Academic Code of Conduct. *https://www.bu.edu/academics/policies/academic-conduct-code/*

   In particular, **if code is submitted that is heavily based on code found online or code produced by an AI, the source must be cited in the comments or a text box.** This is true for all submitted code (e.g., including the project). **Failure to cite a source may be considered plagiarism and reported to CDS for a hearing.**

   In particular, the CDS "GAIA policy" requires that student submit transcripts of their interactions with generative AI whenever they use it for assignments. https://www.bu.edu/cds-faculty/culture-community/gaia-policy/

   You can also consult the CS department's helpful webpage about code plagiarism if you have any doubts about what is considered ethical: https://www.bu.edu/cs/undergraduate/undergraduate-life/academic-integrity/

Outline of Class Meetings: Topics by Day, Optional Readings, Assignments Due

Normal homework goes out and is due on *Mondays* unless otherwise noted.

Week of 1/15
F Introduction

Week of 1/22
M Intro to Jupyter notebooks; Expressions: arithmetic, comparison, boolean, strings; print and other built-in functions; comments (Deitel Ch 2)
W Variables; types; assignment; if/elif/else; lists; tuples (Deitel Ch 3.1-3.6, 5.1-5.4) *HW1 OUT*
F Introducing modules: numpy & matplotlib - plotting, basic analysis (Deitel Ch 7)

Week of 1/29
M foreach loops, while loops; range (Deitel Ch 3.7-3.13)
W More on loops *HW1 DUE, HW2 OUT*
F Defining a function; typical commenting; refactoring; writing a test (Deitel Ch 4)

Week of 2/5
M More on functions
W Hash tables, sets, and dictionaries; pass by reference (Deitel Ch 6), *HW2 DUE, HW3 OUT*
F Coding bigger programs

Week of 2/12
M Reading data into Pandas; escape characters; easy analysis in Pandas (Deitel Ch 7)
W More Pandas examples *HW3 DUE, HW4 OUT*
F Strings, regexes, and data cleaning (Deitel Ch 8)

Week of 2/19
M No class
W (M schedule) Stats overview: chi-square and t-test *HW4 DUE*
F Midterm review

Week of 2/26
M Midterm 1
W Files and Exceptions (Deitel Ch 9) *HW5 OUT*
F Object-oriented programming: creating our own objects (Deitel Ch 10)

Week of 3/4
M Common methods to override; other details of OO Python
W Recursion *HW5 DUE, HW6 OUT*
F Data structure implementation: trees, linked lists, dynamic arrays *PROPOSAL OUT*

Week of 3/11: SPRING BREAK

Week of 3/18
M scikit-learn and k-nearest neighbors (Deitel Ch 15)
W Decision Trees *HW6 DUE*
F Decision Forests ***HW7 OUT***

Week of 3/25
M Regression
W Using ML with Natural Language, *PROPOSAL DUE,*
F Scraping the web with BeautifulSoup ***HW7 DUE, HW8 OUT,*** *OTHER PROJECT DOCS OUT*

Week of 4/1
M Advanced Pandas
W Basic SQL/SQLite
F Midterm 2 review ***HW8 DUE***

Week of 4/8
M Midterm 2
W Complexity and big-O (Deitel Ch 11)
F Analysis of algorithms: binary search, sorts

Week of 4/15
M No class - Patriots' Day
W Graphs, centrality, and BFS *PEER ASSESS IN, HW9 OUT*
F Other ways to speed up code: parallelism, vectorizing, compiler

Week of 4/22
M Visualization
W Making messages stick, *HW9 DUE*
F Lightning Talks, Day 1

Week of 4/29
M Lightning Talks, Day 2
W Lightning Talks, Day 3 **Final Projects Due**

~Last updated 1/12/24~