CSCI E-29: Advanced Python for Data Science

What lies beyond the Jupyter notebook? How can we elevate code from concept to production? What happens when scikit-learn isn't enough? Will that last script die as a one-off or perform just as well for the next 10,000 inputs? The last decade has seen an amazing commoditization of cloud computing and scientific development tools that make it a truly glorious time to be a data scientist, yet the increasing ease-of-use can paradoxically hinder the development of more sophisticated tools if the scientist relies too heavily on magics and never opens the hood to explore how things really work. In this course, we explore the next level of fundamentals that make a difference for truly impactful data science teams in real organizations using complex data. Key topics include formal collaboration techniques, testing, continuous integration and deployment, repeatable and intuitive workflows with directed graphs, recurring themes in practical algorithms, meta-programming and glue, performance optimization, and an emphasis on practical integration with tools in the broader data science ecosystem such as GitHub, Docker, Amazon Web Services, and Hadoop.

This will be a hands-on, technically challenging course targeting data scientists with intermediate to advanced programming skills. Students should be comfortable working in a command line environment, debugging code, working with existing python packages, and be able to read and and learn independently from external source code and documentation.

Key Info

Lectures See Canvas for live times. You may (and are encouraged) to attend in person.

Sections An additional 1 hour weekly recitation section will be required as part of the curriculum. Several will be offered to account for scheduling flexibility.

Recordings Lecture and sections (at least 1 per week) will be recorded and available for browsing offline. Due to the online and global nature of this course, live participation is not strictly required; however, it is highly recommended, especially for sections.

Computing Requirements Students will need access to a computer capable of running Python, Docker, an IDE such as PyCharm, and various scientific software packages. Internet connectivity will also be required. Mac or Linux is highly recommended. Students who only have access to a Windows computer should ensure they can set up Docker.

Instructor Scott Gorlin, PhD: Director of Applied Science at Solaria Labs, a Liberty Mutual endeavor. scottgorlin@fas.harvard.edu

Office hours By request, or as posted to Canvas.

Teaching fellows As posted to Canvas.

Prerequisites CSCI E-7, CSCI E-50, or equivalent. Students should be operationally fluent in Python, including the use and design of functions and classes, and comfortable using standard numerical

libraries such as NumPy, SciPy, and Pandas. Additionally, familiarity with basic concepts in algorithm design (for example, time and memory complexity), machine learning (classification, regression, and clustering), and statistics is useful. The course will make heavy use of git, docker, and the command line.

Textbooks No textbooks are required for this course. Supplemental online resources will be distributed as necessary throughout the semester.

Skills check A Pset 0 will be distributed over the summer and will be due the first week of class. Students will be expected to complete it easily with minimal preparation. Check canvas for the assignment.

Course Structure and Objectives

The course will focus on a number of subjects highly relevant to the modern data scientist or engineer. Broadly, the semester will be divided into the following sections:

1. Workflows: formal tools and methods for capturing work output in a repeatable, testable, and collaborative manner.
2. Skeletons: establishing the backbone of your project. We'll examine frameworks and techniques that establish a common structure across projects or enable quickly tapping into more powerful and scalable functionality. No project should reinvent the wheel, and a large determiner of success is choosing the right framework to build upon.
3. Data: how to store and think about data from a serialization and container perspective. Is your data columnar, slowly evolving, or unstructured? Should it be partitioned or sorted in a key-value store? How can it be appropriately cached or memoized? Should you optimize for read, write, or both?
4. Algorithms: every project is different, but some core techniques prove useful time and time again. From code optimization and compilation options to the magic of pseudorandom hashes, we'll explore some key fundamentals that enable wholly new approaches to old problems.

Assessment

Grades for the course will be determined in part from the following activities:

- 2 exams (Midterm and Final)
- Weekly graded assignments, including an initial Pset 0 to assess course preparedness. Assignments will include code and data submission and may include an automatic grading component.
- 1 independent study project (graduate students only)

- Participation in lecture, sections, Piazza, and otherwise

Accessibility

The Extension School is committed to providing an accessible academic community. The Accessibility Office offers a variety of accommodations and services to students with documented disabilities. Please visit www.extension.harvard.edu/resources-policies/resources/disability-services-accessibility for more information.

Academic Integrity

You are responsible for understanding Harvard Extension School policies on academic integrity (www.extension.harvard.edu/resources-policies/student-conduct/academic-integrity) and how to use sources responsibly. Not knowing the rules, misunderstanding the rules, running out of time, submitting the wrong draft, or being overwhelmed with multiple demands are not acceptable excuses. There are no excuses for failure to uphold academic integrity. To support your learning about academic citation rules, please visit the Harvard Extension School Tips to Avoid Plagiarism (www.extension.harvard.edu/resources-policies/resources/tips-avoid-plagiarism), where you'll find links to the Harvard Guide to Using Sources and two free online 15-minute tutorials to test your knowledge of academic citation policy. The tutorials are anonymous open-learning tools.

Homework Assignments

All homework assignments, quizzes and exams must be your independent work. We encourage discussion of concepts, solving issues, and asking general questions on our course forum. We take academic integrity very seriously. Programmers learn how to overcome problems via popular internet resources frequently. Please ensure that you include citations in your submission if you used any external resource.

Late Policy

We realize that sometimes due to unforeseen circumstances you may be unable to meet a homework deadline. Therefore, we are giving you a credit of 5 extra days that you may use throughout the duration of the course. You must inform your TF if you are planning to use credit for your homework. You can apply a maximum of 3 credits for any homework. 10% of your assignment grade will be subtracted for any additional late day. Homework will not be accepted after solutions are posted on Canvas. Extra days credit cannot be used for PSET 0, the last PSET, or the final graduate project.

Class Participation

Class participation is evaluated as activity during lectures, recitation sections, and online via Piazza. Students that are contributors and are helpful to their colleagues will be noticed and rewarded. Piazza is an excellent resource for collaborative learning.

Independent Project

Graduate students will be required to complete a small independent project which contributes to the open source community, advances a project of their choosing, or otherwise adds to the collective learning experience of the class. The topic and scope of the project should be established early in the semester, and must include a code deliverable and presentation to the class.

Grading

	Undergraduate	Graduate
Homework	65%	60%
Exams (2)	25%	20%
Class Participation	10%	10%
Graduate Project	Not Required	10%

Accessibility

The Extension School is committed to providing an accessible academic community. The Accessibility Office offers a variety of accommodations and services to students with documented disabilities. More Information

Important Notes for Registered Students

We will be using a course management web service called Canvas for all course communication. Please ensure that you get a Harvard e-mail account and access to Canvas. More importantly, it is critical that you check the e-mail registered with Canvas, monitor course announcements and participate in discussions on Piazza (our forum).

Detailed Syllabus

Please use the course calendar for up to date information.

Exact topic schedule is subject to change!

Week	Date	Section	Topics	Event/Notes
1	9/9	Workflows	Introduction	
2	9/16		Continuous Data Science	
3	9/23		Environments and Packaging	
4	9/30	Skeletons	Iterations and Graphs	
5	10/7		Classes, Composition, Salted Graphs	
6	10/14		Advanced Luigi, Dask	Columbus Day - Prerecorded
7	10/21	—	—	Midterm
8	10/28	Data	Dask and Parquet	
9	11/4		DB's and Webapps	
10	11/11		Webdata, Metaprogramming, and API's	
11	11/18	Algorithms	Factories and Optimization	
12	11/25		Memoization and Sketches	
13	12/2		Vizualization	
14	12/9	Showcase	Science Fair	Student presentations
15	12/16	—	—	Final