

Jupyter Notebooks for Coursework

Winter 2020 Pilot Results

UW-IT is exploring three distinct approaches for integrating computational infrastructure and technologies into the curriculum. One of these is Jupyter Notebooks. In December 2019, UW-IT began to work with a few early adopters of Jupyter Notebooks for coursework. This report addresses the results of a pilot study conducted with Dr. David Shean in one course, CEE498/CEWA599 Geospatial Data Analysis, in winter 2020. The pilot had two aims: 1) determine support issues that would need to be addressed for a campus-wide rollout, and 2) identify specific pedagogical challenges and opportunities of integrating Jupyter Notebooks in the classroom.

Technology

Jupyter Notebooks are web-based interactive computational environments that are pre-provisioned with course material. Students connect — each to their own copy of the environment — and develop content as directed, often writing short segments of code. The Jupyter model is successful for a number of reasons. First, containerization allows many students (up to thousands) to develop their work independently. The cloud-based virtual machines host the execution kernels, so students are not installing software on their own machines. Systems such as nbgrader are available to assign and grade coursework. The underlying deployment technology can be used to make modifications and additions as the term progresses.

Pilot Study

Dr. David Shean is an assistant professor in the UW Department of Civil & Environmental Engineering. Prior to the partnership with UW-IT, Shean had used Python for nearly a decade as a researcher, before Jupyter Notebooks became mainstream. The arrival of Jupyter Notebooks in the research world had a significant impact; Shean saw their potential as a teaching tool, highly relevant for investigations in applied academic research.

Shean began using Jupyter notebooks for instruction during UW's Geohackweek, sponsored by the UW eScience Institute, where he observed both the benefits and challenges of shared environments and infrastructure for student learning. Organizers of Geohackweeks started using JupyterHubs three years ago and used Jupyter notebooks during instruction and tutorial sessions. Experience with the hackweek model taught Shean about provisioning resources, scaling, and how best to integrate changes to support student learning. The experience also informed Shean's pedagogical and content knowledge, which translated to the planning and delivery of his courses.

Prior to the start of winter quarter, Shean worked closely with UW-IT's Chris Land to set up his desired infrastructure for the course's JupyterHub. Shean met with Land to discuss requirements, including the authentication mechanism, resource allocation, and the docker image to run on the hub. Because of his previous experience with the eScience Institute and Geohackweeks, Shean was able to provide in detail his desired specifications and packages to be installed. Importantly, UW-IT tested resource allocations to ensure adequate resources for student use, as this had been an issue when Shean taught the course previously. Shean provided Land with previous course material, including entire notebooks, which allowed Land to test run actual course content to challenge the system. In addition, Land was able to update the environments and allocations as necessary during the quarter.

It is important to note that Shean's extensive experience with Jupyter notebooks and JupyterHub makes him an expert user. Shean entered into the UW-IT pilot with a significant vision of what he wanted to accomplish during the course and knowledge of the back-end work necessary to ensure its success. As such, CEE498/CEWA599 was a strong candidate for the initial pilot and presented an excellent opportunity to test UW-IT's capacity to provide support for a potential new service. The large data sets used in the course and the length of student notebooks resulted in the course far exceeding the resource loads originally expected and allocated by UW-IT, and the loads expected for most other courses.

Course Details

Course: CEE498/CEWA599, Geospatial Data Analysis (4 credits)

Time: Fridays, 2:30-5:20

Location: MGH 058

Enrollment: 14

Tools used: Jupyter Notebooks, Jupyter Hub, GitHub, Slack, GoogleDrive, Canvas

TA: None

Student Makeup

The course attracted students from multiple departments, with a variety of professional and research-oriented backgrounds and diverse programming skills. While the 2019 section was mostly populated by advanced PhD students conducting independent research, the 2020 section included an undergraduate senior, professional- and research-degree masters students, and research-track PhD students.

About half of enrolled students identified themselves as novice or advanced beginner users of Jupyter notebooks. Over half of students identified themselves as novice or advanced beginner users of Python, but many students had prior experience in MATLAB and ArcGIS. Required prerequisites/experience in the course included an introductory programming, CS, or scientific computing course, an introductory GIS course, and basic working knowledge of Python. Shean communicated that students without prior knowledge would have to invest additional time to maintain pace with the content and directed students to resources on-campus, such as

eScience Python bootcamps. In addition, Shean encouraged students to work together and to utilize UW campus-wide resources (e.g., eScience consulting) and online documentation to fill any gaps in their Python knowledge.

Class Structure

At the start of the quarter, Shean began each three-hour class period with 15- to 60-minute presentations on course content, followed by an open, collaborative learning environment in which the class solved problems as a group (including the instructor) or in pairs. Shean distributed weekly lab assignments for a total of ten assignments over the quarter. Acting on student feedback gathered through mid-term evaluations, Shean shifted to using more direct instruction in the second half of the course; students had communicated that the content was a lot to absorb and that they would like more traditional instruction. In addition to the weekly lab assignments and required class participation, students were expected to complete a final project in the course, with components of the project due throughout the quarter. Students worked individually or in small groups for their final projects, applying what they learned to an independent research investigation.

Instructor Preparation

With weekly lab assignments in the notebooks, Shean's preparation for the course was considerable. To create each lab, Shean created a notebook with code that produced a specific output. He then deleted select code, distributed the notebooks, and had students complete the code to achieve the specified output. Although Shean taught the course once before using Jupyter notebooks and was able to transfer a bulk of previously developed content, he reported still needing to revise and adapt content each week to serve the diverse needs of his students. Shean anticipates subsequent iterations of the class will require less upfront preparation.

Discussion

Slack, a team communication platform, was used extensively in CEE498/CEWA599. Shean created a dedicated workspace for the course, and organized channels for each lab assignment, IT help questions, and general discussion. Chris Land also joined Slack at the start of the quarter, first in an admin channel, and then in a user-facing IT help channel. Shean, students, and Land used Slack to help manage the course and to address issues that ranged from coding questions to infrastructure troubleshooting.

Feedback/Grading

The winter 2020 CEE498/CEWA599 session had 14 students and no teaching assistant. Shean graded and provided individual feedback on each of the weekly lab assignments as well as on students' final projects. Shean used GitHub to post all course material and assignments. For weekly labs, students submitted their updated notebooks through Github, which allowed Shean to use pull requests to interactively comment on students' notebooks.

While Sean considered a variety of automated graders prior to the start of the course, these required some overhead to prepare, and at the time he was primarily focused on preparing content. For the winter 2020 course, Sean had answer keys for the assignments, and this, plus the small number of students, made individual feedback on the assignments feasible.

Evaluation

Student feedback on the course content and use of Jupyter notebooks for learning was overwhelmingly positive. Sean was pleased with the breadth and depth of student learning and how his adaptations of the course effectively met the diverse needs of the group. As discussed previously, he had to expand the course content and his own teaching abilities to accommodate all needs. He felt that adaptation was effective.

Overall, students favored the notebooks and saw value in its integration into the course. In a UW-IT administered post-course student survey (n=11/14), students expanded on their support for Jupyter notebooks and JupyterHub:

This took away the need to deal with package issues so we could focus on the concepts and the tools.

Having a consistent environment to learn data analysis with Python left more time to learn the subject matter rather than troubleshooting issues of consistency between my workspace and those of the professor and other students.

The Jupyter Lab interface allowed everyone to easily use Jupyter notebooks without taking a lot of time to download/make the system work on personal laptops. The Jupyter notebooks developed for the course (homework exercises) were easy to use and learn from effectively.

The Jupyter notebooks were a great place to collect both lecture material and homework within one framework--it was very helpful to reference material within the place where I needed it to work on code.

Jupyter notebook was completely new to me but I got lots of benefits from it by accessing code online and saving my work there. I loved that instant result of a code [output] and it worked for me to learn better and quicker.

Students especially appreciated the ease of use provided by the JupyterHub environment. A UW-IT-produced document for trouble-shooting common issues that arose during the course may have saved some time for Sean and Land who responded to these issues.

Some students noted that resource limitations occasionally hindered their learning with Jupyter notebooks and JupyterHub. However, CEE498/CEWA599 in particular is a resource-intensive course, and UW-IT was able to effectively manage the students' specific needs.

It would have been nice to have more memory than 8GB allocated for each notebook. 16GB of memory seems to be the minimum nowadays for data analysis processing to work efficiently.

UW-IT Support

For winter 2020, UW-IT's cloud engineering team was instrumental in ensuring the Jupyter notebook and JupyterHub pilot was a success. Land worked closely with Sean to understand his cloud infrastructure needs and created a custom image tailored to geospatial data analyses in time for the start of the quarter. The JupyterHub ran smoothly throughout the course of the pilot and the UW-IT cloud engineering team was able to effectively troubleshoot student issues. Land monitored metrics on resource use and allocated extra resources to the cluster when needed. Sean's experience in managing JupyterHubs in the past also helped maintain stability over the quarter. Sean called Land's support "critical" for avoiding long delays in class and noted that it was extremely important to have someone present as an infrastructure authority who can look into issues during the course and help manage resources.

Future Considerations

As Jupyter notebooks and JupyterHubs roll out on a larger scale, it is likely that most UW faculty will not have the capacity to manage cloud computing issues. Experienced students, familiar with Jupyter notebooks and JupyterHubs and who are capable of managing back end development and technical problems, may be an important source of support for faculty. This type of support might be included in an existing teaching assistant's job duties. With limited permissions to the cloud computing administrator interface, teaching assistants could troubleshoot common issues with students in real time and take some of the support burden off UW-IT staff.

Lessons Learned/Considerations for Instructors

Sean provided a series of recommendations for instructors looking to develop courses that use Jupyter notebooks and JupyterHub in their courses.

- During course development, keep it simple and list out your course objectives. Explore the necessary tools and datasets for the course and identify which sections of data can allow students to effectively learn a specific tool and/or specific content.
- If the course expects an existing foundational knowledge of Python, cover basic coding concepts at the start of the course, but quickly build upon that and shift toward content instruction.
- Anticipate student needs for support and resources while they ramp up the necessary Python skills to succeed. Many students will come to classes with existing backgrounds in other coding languages (e.g., GIS, MATLAB, R, etc.). Students will require time to

transition to Python and newcomers will need to learn how to efficiently manage their notebooks (e.g., key strokes, code and formulae formatting).

- Consider how you will structure Jupyter notebooks for assignments in order to maximize their potential as a teaching and learning tool. Two options: (a) instructor-generated notebooks with partial code that students fill out as part of assignments, or (b) notebooks that have all the code that students can run and explore, with open-ended problems at the end.
- Consider when you will/will be able to give students access to readings, labs, and solutions. Given the dynamic learning environment of Jupyter, providing students access to labs and solutions some days prior to class periods can help maximize your time spent together. When students are able to preview labs before class time, they can ask you or their peers and more targeted and intentional questions.
- With the creation of rubrics and answer keys, teaching assistants can provide feedback on assignments similar in quality to an instructor with some instructor oversight. Larger courses using Jupyter notebooks may require autograding tools, such as nbgrader, that facilitate creating and grading assignments in the notebook.

Next Steps

UW-IT is continuing to pilot the use of Jupyter notebooks and JupyterHub in UW courses. Six new instructors joined the spring 2020 pilot and a report on their experience is forthcoming. Any faculty member interested in using Jupyter notebooks in a course is encouraged to contact UW-IT's Academic Experience Design & Delivery via help@uw.edu.