CS 1671/2071
Human Language Technologies

Session 1: Course introduction and NLP basics

Michael Miller Yoder

January 8, 2025

School of Computing and Information

Overview: Course introduction and NLP basics

- Introductions
- What are human language technologies?
- Course logistics
- CRC JupyterHub setup

About Michael Miller Yoder

- You can call me "Michael"
- Teaching faculty, Pitt School of Computing and Information
- BA, Computer Science from Goshen College (2013)
- PhD, Language Technologies Institute at Carnegie Mellon University (2021)
- **Research interests:**
  - natural language processing (NLP)
  - computational social science
  - data science
  - ethics and bias in AI

Michael's office hours

- By appointment in person in IS 604B or on Zoom
- Sign up for a slot here
  - Link also posted on course website
- Drop in to ask questions about the course or anything else

Introductions

1. What is your name?
2. What is your major/minor/academic interests?
3. What is a language or dialect other than English that you speak, or some your ancestors spoke?
4. [Optional] Is there anything that makes you interested in language technologies or excited to take this class?

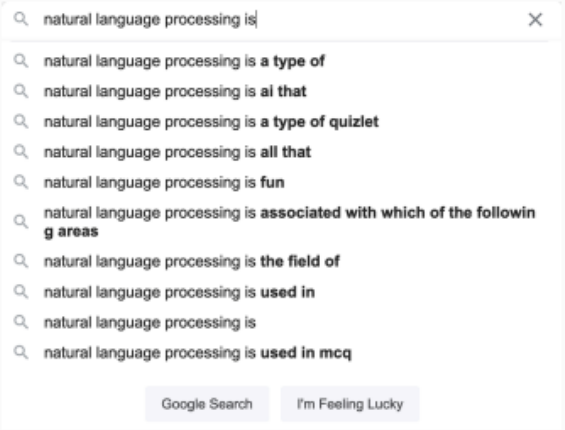
What are human language technologies?

(Human) language technologies

- Also known as
  - **natural language processing (NLP)**
    - "Natural language" = human languages (not programming languages)
  - computational linguistics
- Intersects with
  - artificial intelligence (AI)
  - machine learning (ML)
- Computational **analysis** and **synthesis** of language and speech

NLP is Everywhere

Did you ever wonder how web search engines work...



...or how Google can anticipate what you're searching for?

That's NLP!

Slide credit: David Mortensen

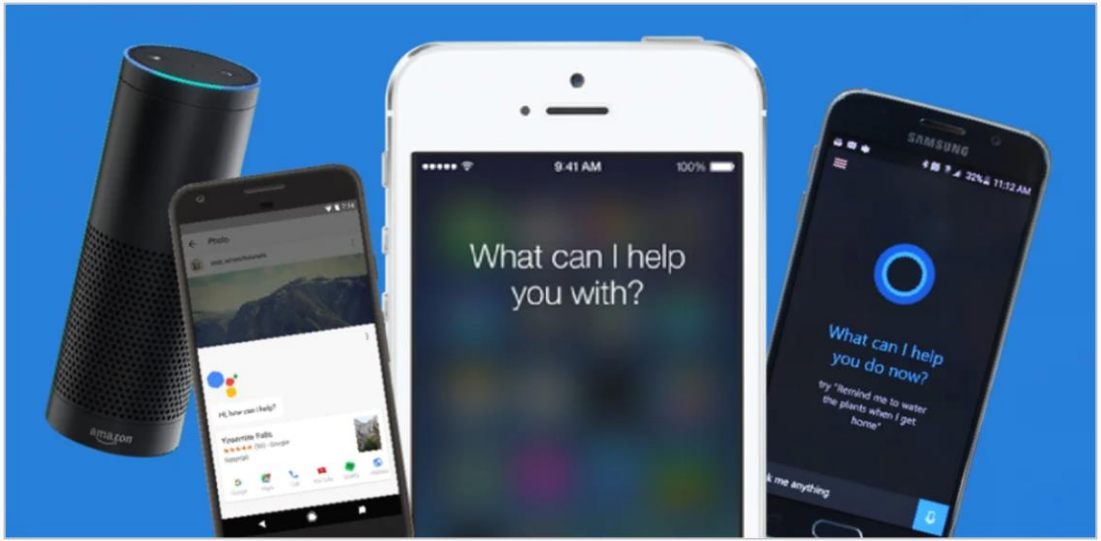
NLP is Everywhere

Did you ever wonder how ChatGPT generates language?

That’s NLP!

NLP is Everywhere

Did you ever wonder how digital assistants work?



That's NLP!

Slide credit: David Mortensen, Xiang Lorraine Li

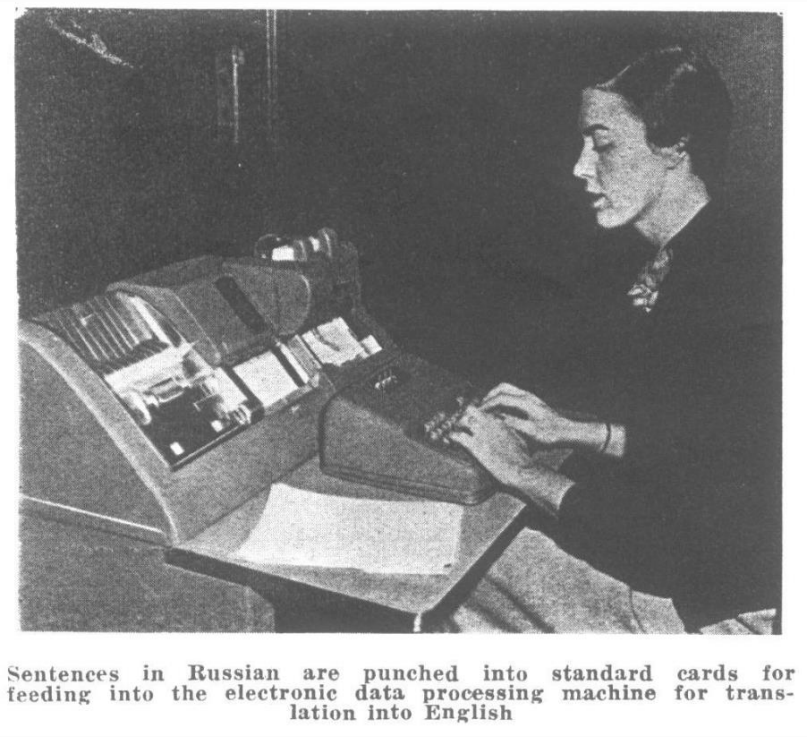
NLP is Everywhere

Did you ever wonder how the government is spying on your every word?

That's also NLP!

Slide credit: David Mortensen

A brief history of NLP



Sentences in Russian are punched into standard cards for feeding into the electronic data processing machine for translation into English

The Georgetown-IBM Experiment.
Credit: John Hutchins

- 1950s: **foundations**
  - Turing Test ("Computing Machinery and Intelligence" paper)
  - Georgetown-IBM Experiment translating Russian to English

A brief history of NLP

Sentences in Russian are punched into standard cards for feeding into the electronic data processing machine for translation into English

The Georgetown-IBM Experiment.
Credit: John Hutchins

- 1950s: **foundations**
  - Turing Test ("Computing Machinery and Intelligence" paper)
  - Georgetown-IBM Experiment translating Russian to English
- 1960s-1980s: **symbolic reasoning**
  - Processing language with hand-built rules

A brief history of NLP

Sentences in Russian are punched into standard cards for feeding into the electronic data processing machine for translation into English

The Georgetown-IBM Experiment.
Credit: John Hutchins

- 1950s: **foundations**
  - Turing Test ("Computing Machinery and Intelligence" paper)
  - Georgetown-IBM Experiment translating Russian to English
- 1960s-1980s: **symbolic reasoning**
  - Processing language with hand-built rules
- 1990s-2010s: **statistical NLP**
  - Learn patterns from large corpora (feature-based machine learning)

A brief history of NLP

Sentences in Russian are punched into standard cards for feeding into the electronic data processing machine for translation into English

The Georgetown-IBM Experiment.
Credit: John Hutchins

- 1950s: **foundations**
  - Turing Test ("Computing Machinery and Intelligence" paper)
  - Georgetown-IBM Experiment translating Russian to English
- 1960s-1980s: **symbolic reasoning**
  - Processing language with hand-built rules
- 1990s-2010s: **statistical NLP**
  - Learn patterns from large corpora (feature-based machine learning)
- 2000s-today: **neural NLP, LLMs**
  - Powerful models of language from "deep" layers of neural networks trained on tons of text

The other NLP 😂

Neuro-linguistic programming (pseudoscience)

Course objectives and overview

Learning objectives

At the end of this course, a student will be able to implement an NLP system to achieve a desired outcome from language data.

Learning objectives

When coming across a natural language problem, students will be able to:

- Preprocess text data into a machine-readable numeric format
- Extract features from text that are required for running machine learning models, such as with n-gram or dense vector representations
- Explain what is needed for machine learning-based NLP systems, including supervised training data, algorithms to learn patterns in that data (with possibly pretrained models), and unseen test data for evaluation
- Choose relevant machine learning algorithms to try on different types of NLP task
- Explain the basics of language structure that are relevant to NLP. These include syntax and semantics from linguistics
- Identify potential ethical pitfalls (such as imbalanced training data, model amplification of biases) in an NLP system and ways to address them
- Communicate motivation, key components, and implications of an approach to NLP tasks in writing

Structure of this course

MODULE 1	Prerequisite skills for NLP	text normalization, linear alg., prob., machine learning	
	Approaches	How text is represented	NLP tasks
MODULE 2	statistical machine learning	n-grams	language modeling text classification
MODULE 3	neural networks	static word vectors	text classification
MODULE 4	transformers and LLMs	contextual word vectors	language modeling text classification sequence labeling
MODULE 5	NLP applications and ethics	machine translation, chatbots, information retrieval, bias	

Resources

Textbook (free)

- Dan Jurafsky and James H. Martin, *Speech and Language Processing*, 3rd edition draft, 2024-02-03.
- Available completely free online: https://web.stanford.edu/~jurafsky/slp3/
- Why do the readings?
  - Learn better: get the information from readings and class
  - Spend class time more efficiently: come with questions
  - They help for quizzes

Class sessions

- Cover the most important parts of the course content
- Students are expected to attend each class
- **Attendance will be taken via Top Hat at a number of random class sessions**
- **Bring a laptop or tablet for in-class coding exercises**
- Slides will be provided in advance of each session for note-taking
- There are no current plans for recording classes

Infrastructure

- Website
  - https://michaelmilleryoder.github.io/cs1671
  - Or https://bit.ly/cs1671
  - How do I find the website?
    - Link is in “Syllabus” on Canvas
    - In first Canvas announcement
    - From the ‘Teaching’ page of my website: https://michaelmilleryoder.github.io
  - Up-to-date syllabus and schedule
  - Class slides
  - Homework assignment and project instructions
- Canvas
  - Complete quizzes
  - Submit assignments (homework and project assignments)
  - Receive course announcements
  - Post questions
  - Check your grade

Programming languages and software

- **Python will be the expected programming language used in assignments**
- Python-based data science packages (numpy, pandas, jupyter, scikit-learn, pytorch) will be used and encouraged in both assignments and the project
- If you have zero familiarity with Python (no shame):
  - Check out the **Tutorials on Python and data science** section of the course website under **Learning resources**
- You can use whatever you want for the project

Assessments

Assessment overview

Assessment	Points	Percentage of grade
Homework assignments (3) total	270	27%
Project	230	23%
Quizzes (5) total	200	20%
Midterm exam	200	20%
Participation	100	10%

Homework assignments

- 3 total
- 9% of total course grade each
- Hands-on coding assignments in Python
- Due 2-3 weeks after they are released
- Descriptions will be on the course website
- Submitted through Canvas

Project

NLP is inherently hands-on. The course project will demonstrate an ability to build a system that takes in language data and automatically produces some sort of output.

- Example tasks: machine translation, question answering, dialogue system (chatbot), aspect-based sentiment analysis, named entity recognition
- You can also come up with your own idea for an NLP system you want to build!
- I will provide more information on the example projects later

Project methods

Projects will usually involve building and evaluating multiple systems on a task:

1. Classical approaches of n-gram text representation and feature-based machine learning models
2. (Optionally) Neural networks with static word embeddings
3. Contemporary LLM-based approaches

Project idea and group formation process

1. Look through example project ideas and possibly come up with your own
2. Submit ideas you would be interested in working on (either examples or your own)
3. Rank all ideas generated by the class by how much you'd like to work on them
4. Teaching staff will group students into groups of 2-4 by which ideas they are interested in
    - There will be peer review of teammates

Project components

Component	Points	Percentage of course grade	Due
Idea submission form	5	0.5%	01-30
Idea ranking form	5	0.5%	02-06
Proposal	60	6%	02-28
Proposal presentation	*None*	*None*	03-10
Progress report	40	4%	03-20
Final presentation	*None*	*None*	*TBD*
Final report	120	12%	04-24

Quizzes

- On Canvas
- Checks for comprehension of the main important ideas in preceding weeks
- Designed to motivate you to do the reading and come to class
- Auto-graded (generally multiple choice or short answer)

First will be due Jan 30

Midterm exam

- In-person exam on Apr 2, so more than 75% of the way through the course
- Will cover all content covered in Modules 1-4 (before then)
- 20% of your final grade
- There will be a exam session in the class session before that where you can ask questions and we can go through stuff on the board
- I will allow some form of notes (probably 1 page), otherwise will be likely be closed-book

Participation grade

- Class interactions (activities, discussions) are better with more people in class
- Incentives to come to class and engage
- 10% participation grade
  - 6%: attendance on a random subset of class sessions, taken via Top Hat
  - 4%: engagement
    - Have you ever asked a question in class, afterward or over email?
    - Do you participate in in-class activities?
    - If yes to either, you will be fine

Policies

Grading scale

Range	Letter grade
92.5 – 100%	A
90.0 – <92.5%	A-
87.5 – <90.0%	B+
82.5 – <87.5%	B
80.0 – <82.5%	B-
77.5 – <80.0%	C+
72.5 – <77.5%	C
70.0 – <72.5%	C-
67.5 – <70.0%	D+
62.5 – <67.5%	D
60.0 – <62.5%	D-
< 60%	F

Late work

- Students are granted 5 total late days across all homework assignments without penalty.
- After those five late days, you will be penalized **10% for each day that your submission is late** except in extreme unforeseen circumstances.
- Group project work will be penalized 10% for each day late. No late work will be accepted for the final project report.

Homework resubmissions

- If you are unsatisfied with your grade on an assignment and wish to resubmit work, talk with me
- If you completely miss parts of an assignment are missing (sections of the rubric are 0), a resubmission may be possible.
- Updated or added text in resubmitted reports must be highlighted in yellow.
- Resubmissions are subject to an automatic 10% deduction. Only 1 resubmission per homework assignment will be accepted.

Academic integrity

- Students in this course will be expected to comply with the University of Pittsburgh's Policy on Academic Integrity. Any student suspected of violating this obligation for any reason during the semester will be required to participate in the procedural process, initiated at the instructor level, as outlined in the University Guidelines on Academic Integrity
- Discussing tools, concepts, and formalisms is acceptable collaboration
- Sharing code is prohibited (except when working together on the group project)

Generative AI policy

- You are allowed to use generative AI (ChatGPT, DALL-E, GitHub Copilot, etc) in some circumstances
    - Exposes you to the current capabilities and limitations of such systems
- Allowed use:
    - **Use as an aid, not for a finished product.** Generating ideas, study guides, bibliographies, or for grammar (watch for hallucinations, though) is ok. Drafting entire homework assignments or project reports, even if you revise the draft, is not ok.
    - **Cite its use.** Citing the generative AI's tool contribution to your work is required. See the APA guidelines on how to cite ChatGPT.
    - **You are responsible for the work you turn in.** LLMs and other generative AI systems can and do generate biased, socially problematic language and assert unfounded claims.
- When in doubt, ask instructor if specific uses are ok. There will be no retaliation for asking.

Disability rights

Many people have disabilities. **We view disabilities as deficits not in disabled people but in the institutions and societies that are structured to disadvantage disabled people.**

If you have a disability (visible or invisible), please let us know as soon as possible (you don't need to tell us the nature of the disability). You are encouraged to work with Disability Resources and Services (DRS), 140 William Pitt Union, (412) 648-7890, drsrecep@pitt.edu, (412) 228-5347 for P3 ASL users, as early as possible in the term. DRS will work with you to determine reasonable accommodations for this course. This might include lecture materials that are usable by people with visual disabilities, sign language interpretation, captioning, flexible due dates, etc.

Adapted from policies by David Mortensen and Lori Levin

Maintaining scholarly discourse

In this course we will be discussing some complex issues. It is essential that we **approach this endeavor with our minds open** to evidence that may conflict with our presuppositions. Moreover, **it is vital that we treat each other's opinions and comments with courtesy even when they diverge and conflict with our own**. We must avoid personal attacks and the use of ad hominem arguments to invalidate each other's positions. Instead, we must develop a culture of civil argumentation, wherein all positions have the right to be defended and argued against in intellectually reasoned ways. It is this standard that everyone must accept in order to stay in this class; a standard that applies to all inquiry in the university, but whose observance is especially important in a course whose subject matter is so emotionally charged.

JupyterHub setup

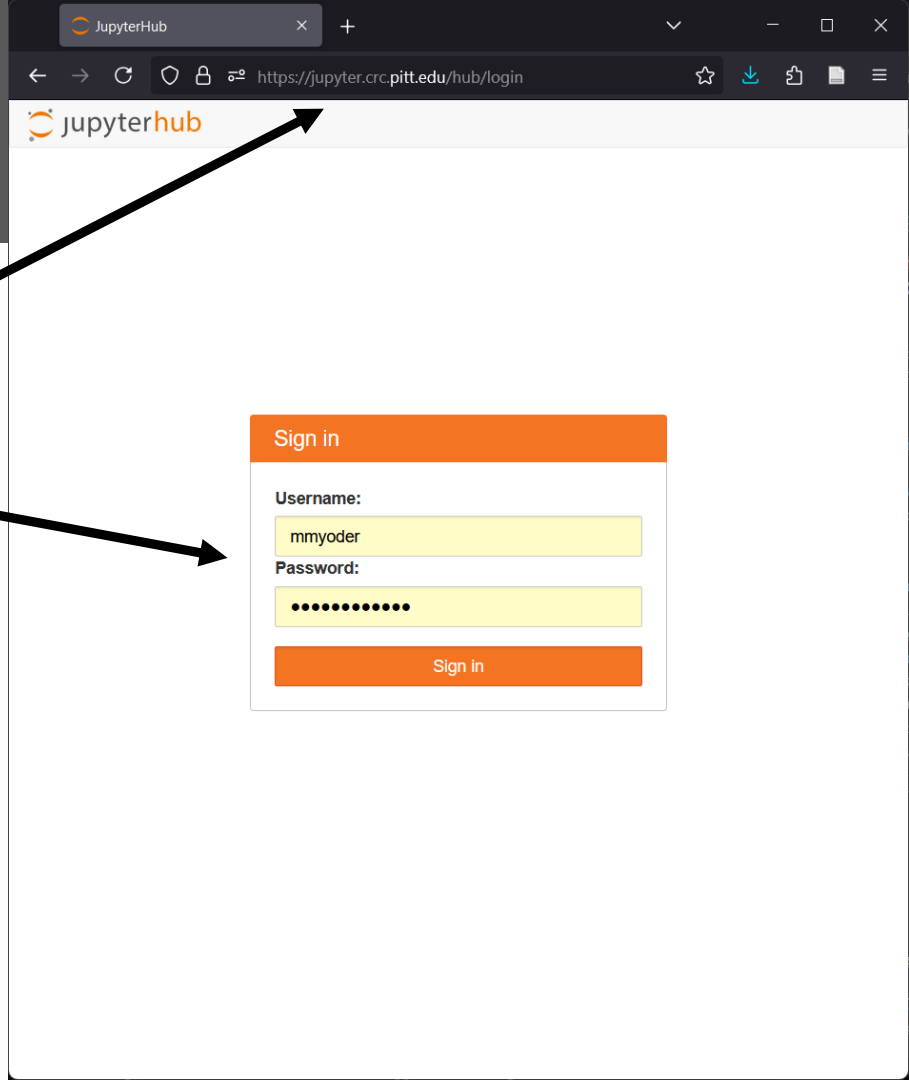
CRC and JupyterHub

- CRC (Center for Research Computing) is a Pitt center providing computing services on various clusters
- They maintain a JupyterHub where people can run Jupyter Notebooks on their servers
- What we will be using the CRC for:
  - Working through code examples in class
  - Writing code to submit as part of homework assignments
  - Running code and storing data for your projects (if you want to)

Logging in to your CRC JupyterHub account

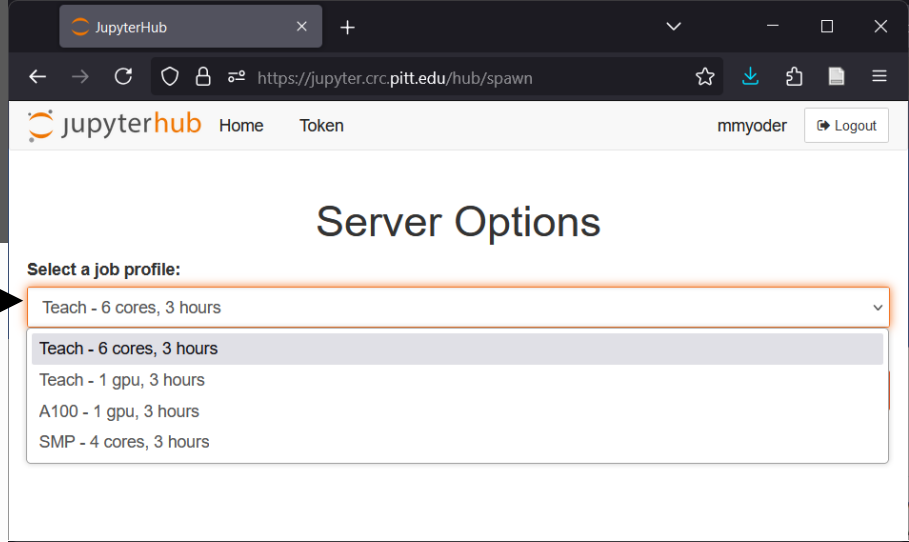
1. Go to **jupyter.crc.pitt.edu** in a web browser
2. Log in with your Pitt credentials

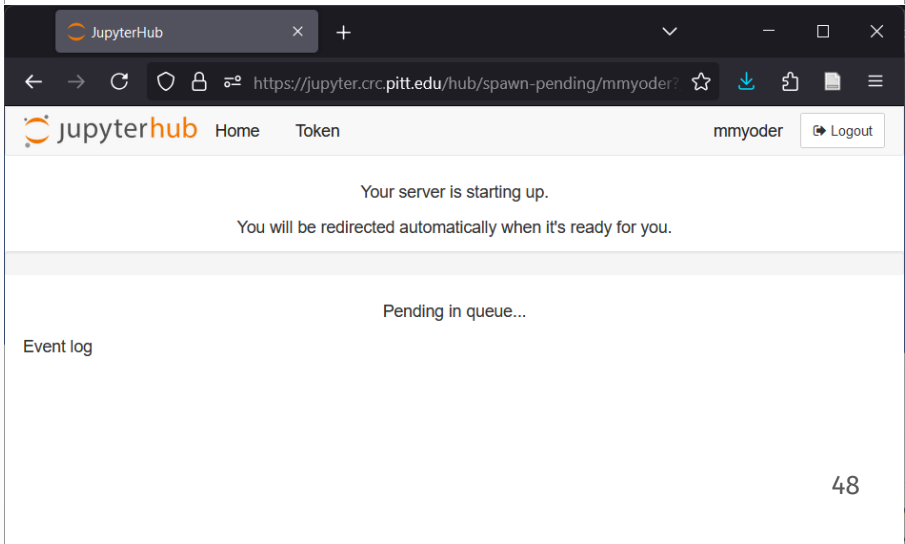
Note that if you are off-campus, you have to log in to the Pitt VPN first through the GlobalProtect app. Instructions: https://services.pitt.edu/TDClient/33/Portal/KB/ArticleDet?ID=293



Starting a Jupyter Notebook on the CRC JupyterHub

1. Select **Teach – 6 cores, 3 hours**
   We might use the GPU options later on in the course
2. Click the **Start** button
3. Wait for the server to start up

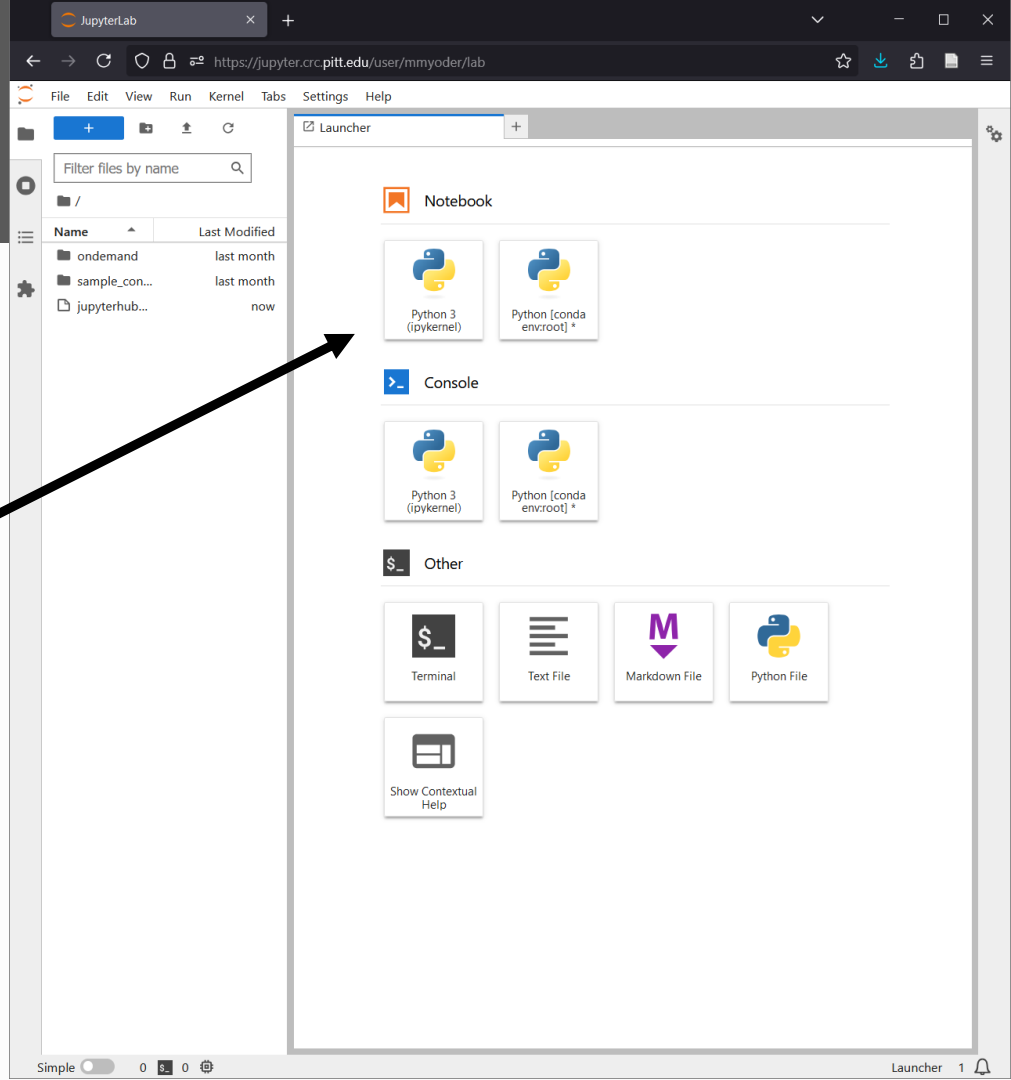




Welcome to your JupyterLab

Files are here

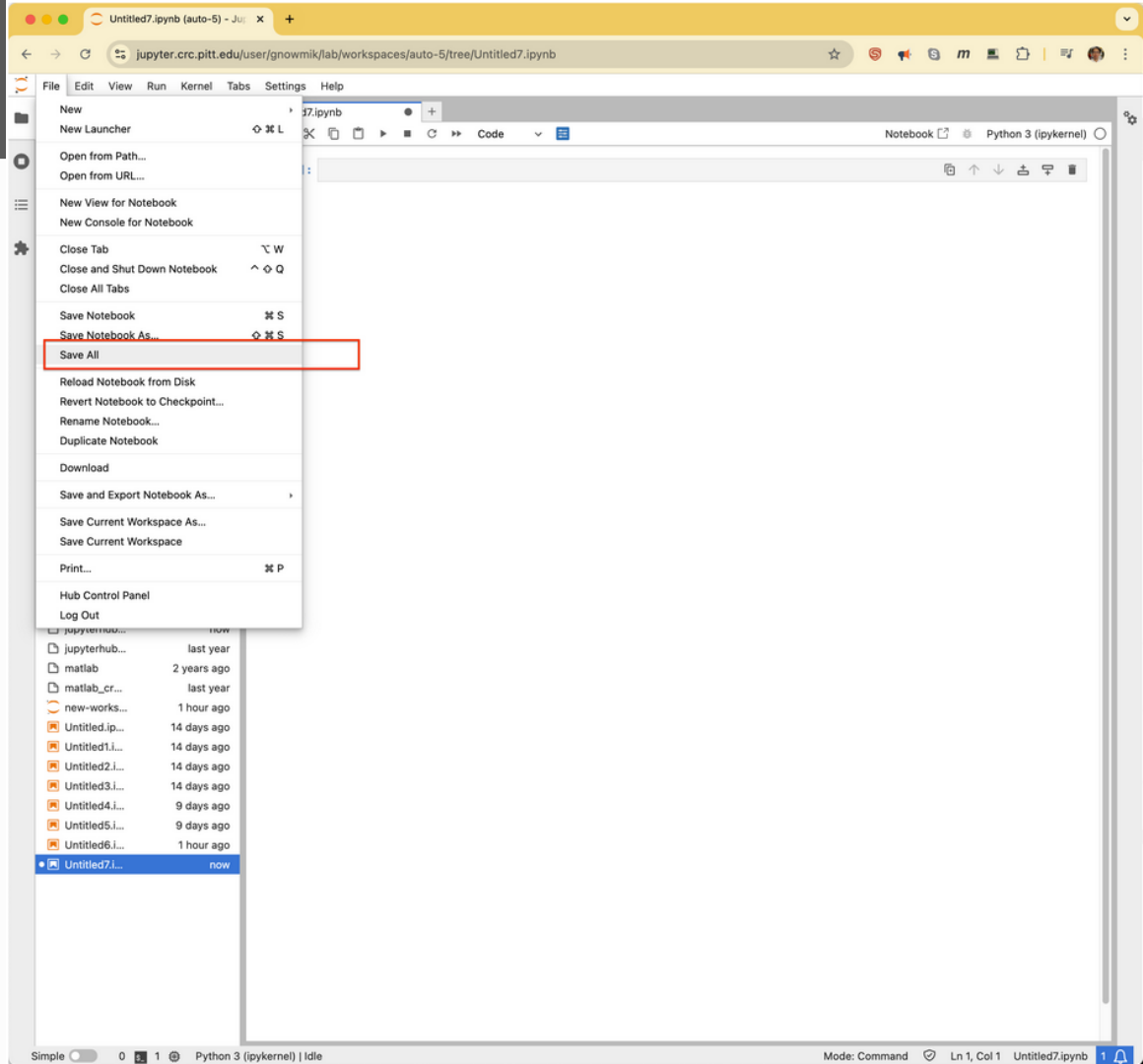
Launch a new Jupyter Notebook by clicking Python 3 (ipykernel) under Notebook



Jupyter Notebook basics

- Each block is called a “cell”
  - Has input and possibly output
  - Input can be Python code, Markdown or shell commands (after `!`)
- Modes
  - Command mode
    - Move, select, manipulate cells
    - Get into command mode by clicking anywhere outside of a cell
  - Edit mode
    - Edit content of a particular cell
- Running cells
  - Click “Run” button or do Ctrl+Enter (on Windows or Linux, Cmd+Enter on Mac) to run code or render Markdown
  - Any result will be shown in the output of the cell

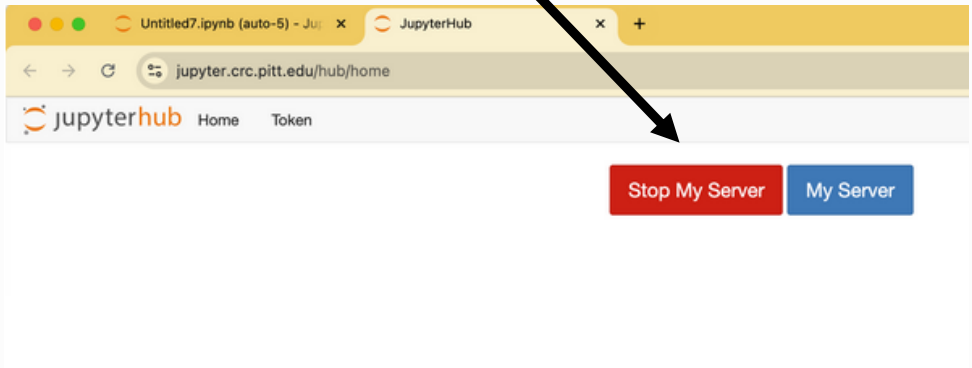
Saving your work

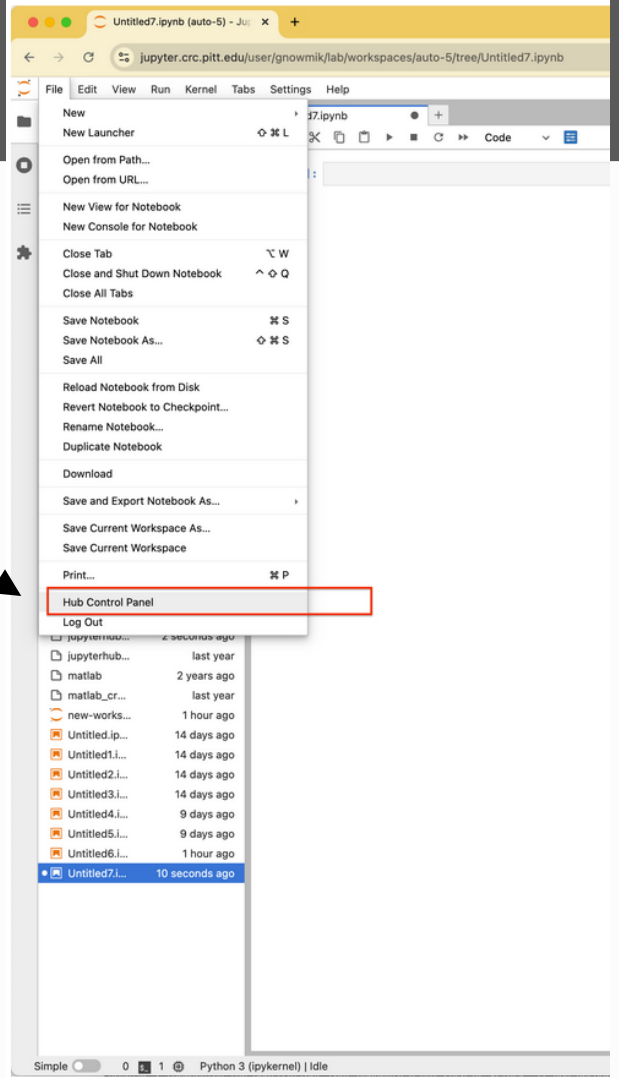


Ending your session

Be sure to save your work before ending the session

1. Select **File** > **Hub Control Panel**
2. Click **Stop My Server**





Questions?