ECS 119:
Data Processing Pipelines

Lecture 0 - Fall 2024

Welcome! (Am I in the right place?)

- This is a **required course** for data science majors
- Open as a CS non-major elective (100-119)
- Cap: 120, 67 currently registered
- 4 units

Basic Details

- **Instructor:** Caleb Stanford
- **TA:** Muhammad Hassnain
- **CRN:** 49704
- **Units:** 4
- **Lectures:** Monday, Wednesday, Friday 3:10-4pm in Teaching and Learning Complex 3215
- **Discussion section:** Mondays at 9am in Olson Hall 206.
- **Office hours:** See Piazza
- **Final exam:** Wednesday, December 11, 6-8pm.

Syllabus + schedule + all of the above information is on the class repository

About the Instructor

What I do: programming language design for data processing and systems applications (Started at Davis July 2023)

DavisPL Research Group

Website

Class TA

Muhammad Hassnain

Discussion section:
Mondays at 9am in Olson Hall 206.

Office hours: TBD on Wed

Plan for today

1. Course introduction
2. Syllabus and logistics
3. Q+A

Plan for today

1. Course introduction
2. Syllabus and logistics
3. Q+A

What is this class about?

ECS 119: Data Processing Pipelines

Home • Schedules and Classes • ECS 119: Data Processing Pipelines

Subject

ECS 119

Title

Data Processing Pipelines

Status

Active

Units

4.0

Scenario 1

You have compiled a spreadsheet of **1K (1,000) movies** you have watched or want to watch with associated data (name, date, rating on a scale from 1 to 10). You want to put together an app which helps you visualize the data by showing the movies you liked the most and whether the distribution of ratings changed over time.

1. What **software components** does your app need?
2. What **design constraints** must your software cope with in this scenario?

Scenario 2

You are a marketing analyst is studying website traffic data for a mid-sized e-commerce company. You have a dataset of **10M (10,000,000) user sessions**, each together with click-through rates and detailed purchase history. You want to put together an app which models user behavior and identifies trends in purchase history over time.

1. What **software components** does your app need?
2. What **design constraints** must your software cope with in this scenario?

Scenario 3

You are working as a consultant on a project for a large telecommunications company. As part of the project you have a dataset of **100B (100,000,000,000) data points** including call records, billing information, and service usage during the company's recent history. You want to build a predictive model which will identify potential communications bottlenecks and dropped calls in real time.

1. What **software components** does your app need?
2. What **design constraints** must your software cope with in this scenario?

Poll

https://forms.gle/M3hdys1dUSknEN916

https://tinyurl.com/y4trfdxa

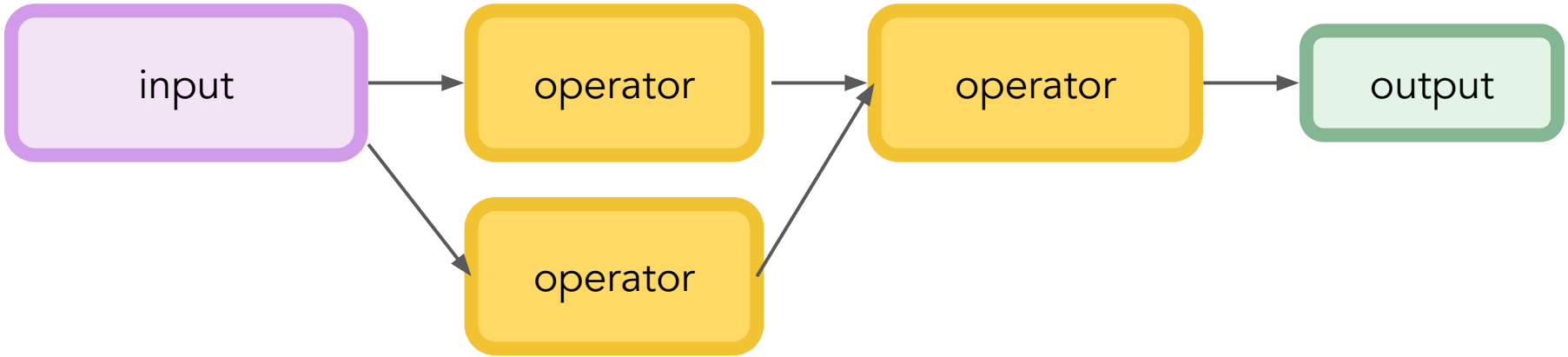
A common theme

Components, design constraints, tools and techniques differ depending on the size of the dataset…

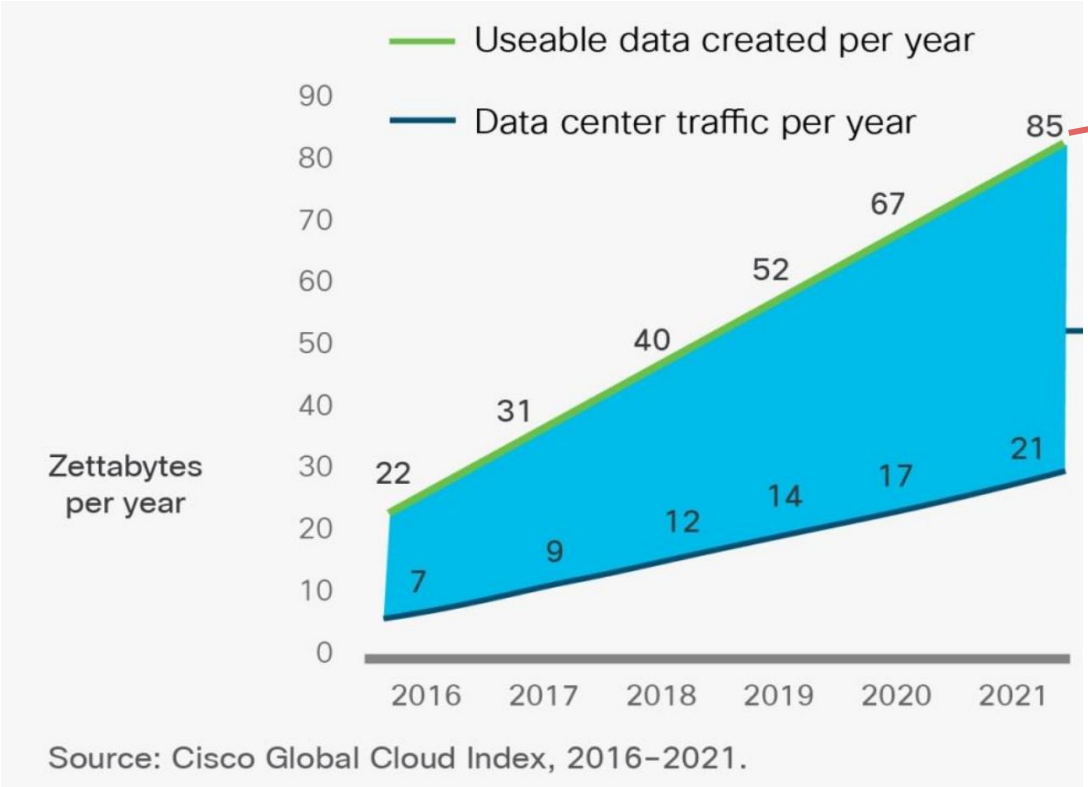
- … 1K datapoints
- ... 10M datapoints
- … 100B datapoints

Additional constraints for large datasets

Parallel computing



Distributed and real-time computing



More data produced than can be feasibly stored

Distributed computing – failures

How did hackers compromise my EC2 instance?

Asked 8 years, 6 months ago Modified 2 years, 8 months ago Viewed 8k times

11

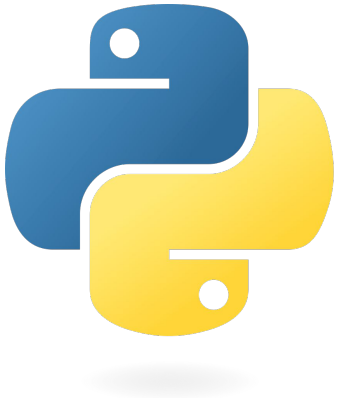
My EC2 instance was hacked recently. It doesn't really matter as I'm just starting my website and there was no sensitive information on my server yet, but I do plan for there to be in the future. I am going to terminate the compromised server and set up another one in order to secure my website.

My issue is that I want to make sure this never happens again, and the only way I can convince

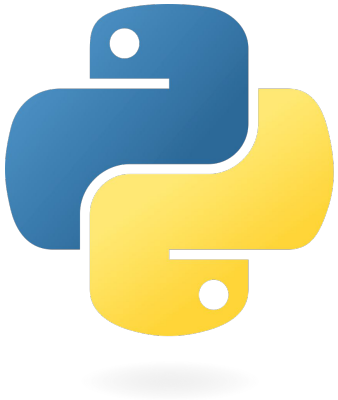
What this course is about

Principles and practice of working with **large datasets…**

… working with **real tools** used in industry



… working with **real tools** used in industry



✨ Hands-on Approach ✨

⚠️ New course disclaimer

Some things will be **experimental!**

Q: What are the prerequisites?

ECS 116 or 165A

Ability to program FizzBuzz

Q: Is this an AI/ML course?

Short answer: No

Long answer: No, but the tools and software used in this class are of fundamental importance for building, training, and deploying large ML models in industry

Q: Is this course right for me as an elective?

Short answer: Yes in some cases

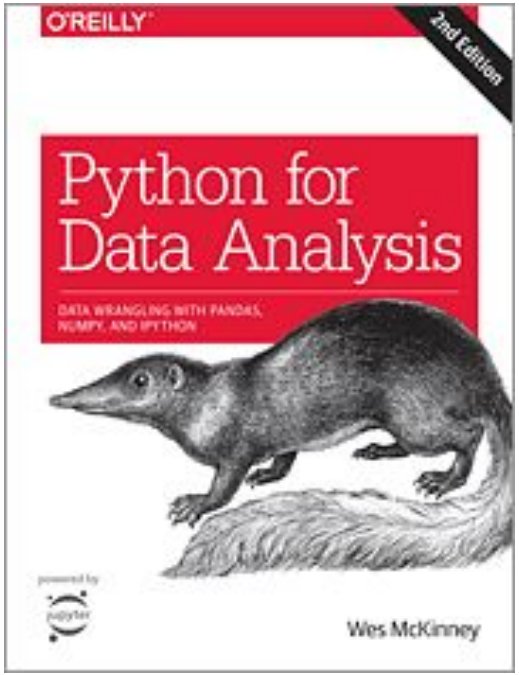
Disclaimer: Non-major CS class – does not count as an upper-division elective*

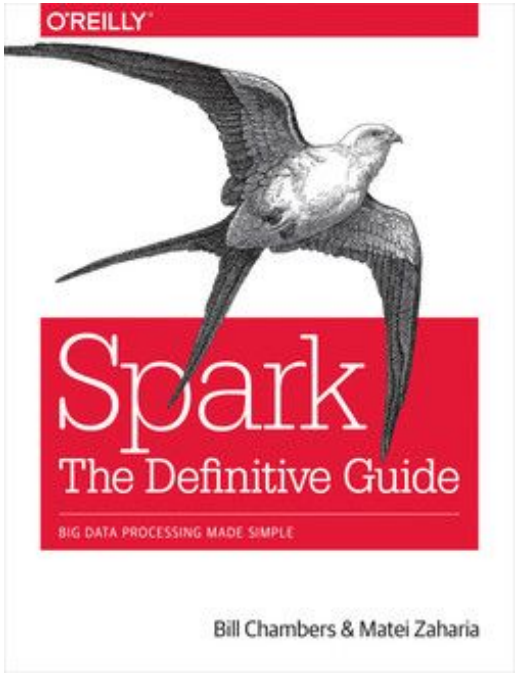
- * I am hoping to get this changed, however, this probably won't happen for this quarter

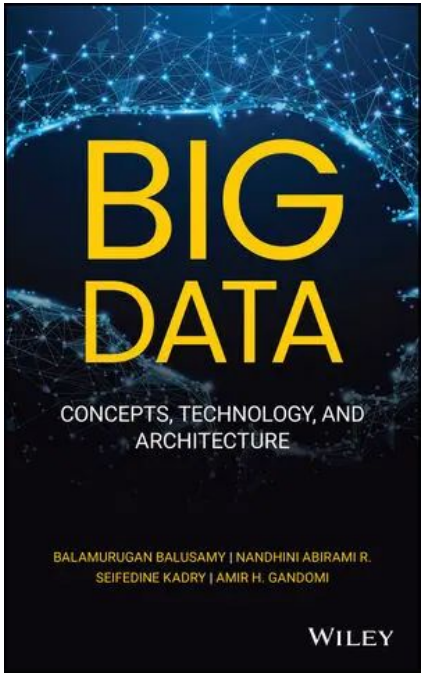
Long answer: The material will be relevant to you if:

- You want to know about the fundamental principles & practice behind large-scale data processing tasks
- You want to gain hands-on experience with the tools used in industry

Textbooks (Optional)







Learning objectives

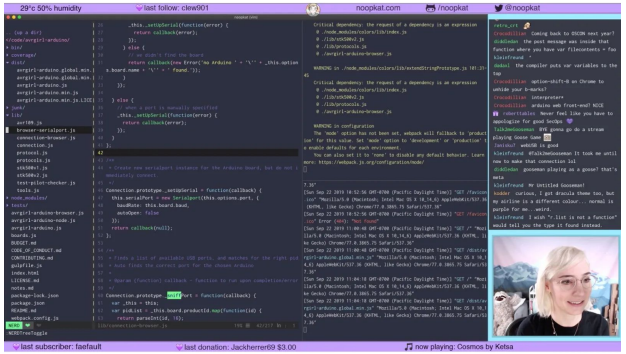
See Syllabus

Plan for today

1. Course introduction
2. Syllabus and logistics
3. Q+A

Lectures

- **I generally lecture with live coding (not slides)**



- Programming can only be learned by **doing!**

Attendance: Lectures and discussion section are encouraged, but not required

- (In-class polls can be made up at any time)

To follow along…

https://github.com/DavisPL-Teaching/119

```
> git clone git@github.com:DavisPL-Teaching/119.git
> git pull
```

Piazza

Please join the Piazza

Grade breakdown

- **Participation (10%):** via in-class quizzes
- **Homeworks (50%):** 3 assignments, each assignment including a project component, plus homework 0
- **Midterm (10%):** covering the first half and main concepts of the course
- **Final Exam (30%):** covering all topics covered in the course.

Minimum grade cutoffs: 93%=guaranteed A, 90%=guaranteed A-, etc.

Final exam will be curved

Attendance and participation (10%)

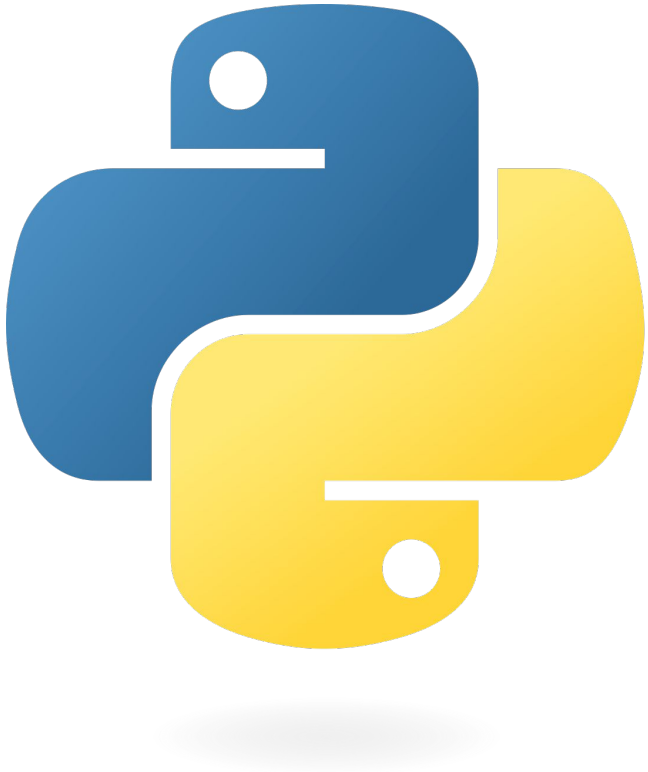
Fill out the in-class polls (participation points only)

If you are sick: Starting from Wednesday, you may join the class remotely via Zoom (the quality may not be as good)

If you miss class: Lectures are recorded. You can make up the in-class polls at any time

Homeworks (50%)

We will be working in Python for most of the course

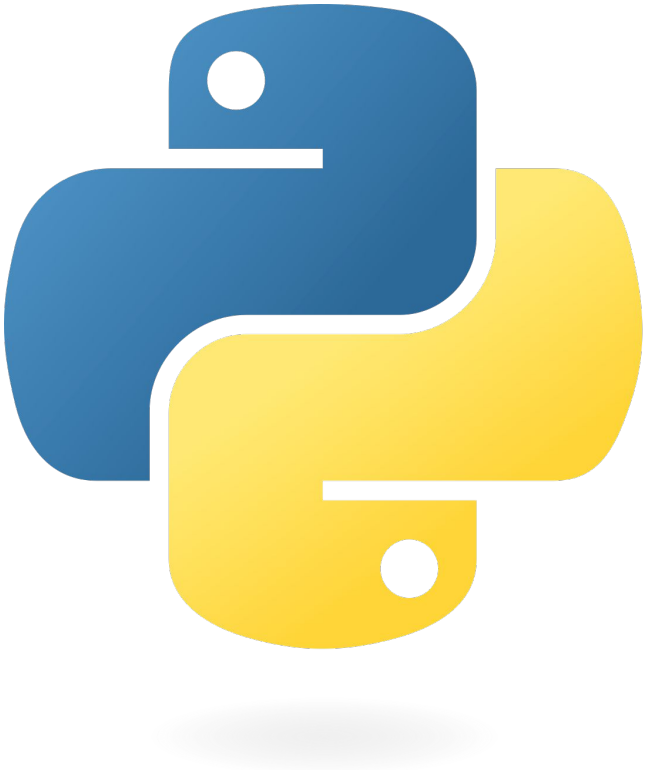


Homeworks (50%)

We will be working in Python for most of the course

… plus, completing a **project in 3 parts**

I.e., 3 homeworks, each with a project component



Homework 0: installation help

(submission link posted soon, probably Friday)

Homework Grading

Most important: please run your code

- Software engineering is about running software!
- Experienced engineers will run and test their code frequently
- I do not like taking points off for code that does not run ☹️

Rough outline of homeworks (and modules)

Homework/Project 1: Basic data processing pipelines on a **single machine**

Homework/Project 2: Gathering larger data sources (including web scraping), building **parallel** data pipelines and optimizing performance

Homework/Project 3: Building advanced pipelines with **real-time, distributed, and cloud** components

Exams (40%)

- There will be one midterm and one final exam
- Why exams? (more on this in a bit)

Platforms

Class discussion, Q+A, and announcements:

- ECS 119 on Piazza
- Don’t email me, post to Piazza!
- Make your post public and (if you prefer) anonymous

Homeworks: GitHub Classroom

Exams: Gradescope

AI Policy

- Allowed and encouraged! (But not required)
- Advice from Jason Lowe-Power
- Midterm and final exam will be in-class and closed-book

AI is a powerful tool! Please use it to help you (and not the other way around)

Collaboration Policy

- Allowed and encouraged! (But not required)
- Everyone should submit their own solution
- Please list your collaborators at the top of your homework

Schedule

Schedule

Communication reminders

TA: **Hassnain**

Office hours: See Piazza

Please use Piazza for questions (not email)

Respect and discrimination

✨ Please be nice! ✨

Include everyone in group discussions

Reach out to me in case of any problems

Plan for today

1. Course introduction
2. Syllabus and logistics
3. Q+A

Questions for me?

Reminder

Please join the Piazza

Rough topic list…

- Basics of data processing
- Software engineering tools
- Input data sources
- Parallelism
- Distributed computing
- Real-time and streaming data processing
- Cloud computing tools

Learning objectives

- Use Python and other scripting tools to manage and manipulate data on a single machine.
- Understand the components, techinques, tools, and performance metrics of setting up data processing jobs in Python.
- Understand the concepts of parallelism, types of parallelism, and parallelization mechanisms, via tools like MapReduce, Hadoop and Spark.
- Understand how software engineering tools and configuration are integrated into a data project, via tools like Git and the shell and other orchestration.
- Understand the concepts of distributed computing and distributed data processing, including distributed consistency requirements, and how it manifests in real-world applications.
- Understand advanced topics including programming over real-time and streaming data sources and using cloud platforms such as AWS, Azure, and Google Cloud.