Course Overview

INST447 Data Source and manipulation

Wei Ai
aiwei@umd.edu

So… What is Data Science?

Simply speaking, it is the Study of Data.

So… What is Data Science?

Simply speaking, it is the Study over the Lifecycle of Data.

It involves a Pipeline of tools.

How data is generated, harvested, processed, stored, consumed,
and transformed into knowledge

Data Science Pipeline

- Forming a research question
- Obtaining data
- Data ingestion
- Data cleaning
- Data wrangling
- Data Analysis and modeling
- Communicating results
- …

A Bigger Image of the Data Science Pipeline — Curriculum of a Master of Applied Data Science Program

Formulating problems

- Introduction to Applied Data Science
- Contextual Inquiry
- Data Science Ethics

Collecting and processing data

- SQL & Databases
- SQL Architectures & Technologies
- Big Data: Efficient Data Processing
- Big Data: Scalable Data Processing
- Data Manipulation
- Experiment Design and Analysis

Analyzing and modeling data

- Math Methods for Data Science
- Visual Exploration of Data
- Data Mining I
- Data Mining II
- Supervised Learning
- Unsupervised Learning
- Deep Learning
- Machine Learning Pipelines
- Causal Inference
- Natural Language Processing
- Network Analysis

Presenting and integrating results into action

- Information Visualization I
- Presenting Uncertainty
- Communicating Data Science Results
- Information Visualization II

Real world applications of data science

- Search and Recommender Systems
- Social Media Analytics
- Learning Analytics
- More to come

Culminating learning experiences

- Project I: synthesis of computational techniques to collect and process big data
- Project II: synthesis of analytics and machine learning techniques to analyze data and present results
- Project III: capstone that applies end-to-end data science techniques to real world scenarios

https://www.si.umich.edu/programs/master-applied-data-science-online/curriculum

A Bigger Image of the Data Science Pipeline — Curriculum of a Master of Applied Data Science Program

Formulating problems

- Introduction to Applied Data Science
- Contextual Inquiry
- Data Science Ethics

Collecting and processing data

- SQL & Databases
- SQL Architectures & Technologies
- Big Data: Efficient Data Processing
- Big Data: Scalable Data Processing
- **Data Manipulation**
- Experiment Design and Analysis

Analyzing and modeling data

- Math Methods for Data Science
- Visual Exploration of Data
- Data Mining I
- Data Mining II
- Supervised Learning
- Unsupervised Learning
- Deep Learning
- Machine Learning Pipelines
- Causal Inference
- Natural Language Processing
- Network Analysis

Presenting and integrating results into action

- Information Visualization I
- Presenting Uncertainty
- Communicating Data Science Results
- Information Visualization II

Real world applications of data science

- Search and Recommender Systems
- Social Media Analytics
- Learning Analytics
- More to come

Culminating learning experiences

- Project I: synthesis of computational techniques to collect and process big data
- Project II: synthesis of analytics and machine learning techniques to analyze data and present results
- Project III: capstone that applies end-to-end data science techniques to real world scenarios

https://www.si.umich.edu/programs/master-applied-data-science-online/curriculum

What is this course about

Syllabus Example (1)

#	Planned Topic
1	Introduction & Overview
2	Command Line Intro and Python Review
3	Data Sources & Storage; Problems, Issues, & Bias
4	Metadata Standards & Extraction
5	Intro to XML; Parsing XML
6	Querying & Transforming XML
7	Intro to JSON
8	More JSON
SB	Spring Break
9	Web Scrapings & APIs
10	More APIs
11	Cleaning Data
12	More Cleaning
13	Even More Cleaning
14	Advanced Topics
15	Project Presentations
FW	Finals Week

Syllabus Example (2)

Week	Mod	Date	Topic	Date	Topic	Readings (see links below)
01	00	08/29	Intro & Overview	08/31	IPython, Jupyter	DSH Ch. 1, Jupyter Notebook docs
02	01	09/05	Thinking about data	09/07	Numpy, Pandas	DSH Ch. 2, 10 min. to Pandas
03	02	09/12	Data cleaning	09/14	Idiomatic Pandas	DSH Ch. 3 (97–140), Modern Pandas
04	03	09/19	Aggregation, Plotting	09/21	Matplotlib	DSH Ch. 3 (158–169), DSH Ch. 4
05	04	09/26	Data wrangling	09/28	Pandas API	DSH Ch. 3 (141–157)
06	05	10/03	Data formats	10/05	jq	P4E Ch. 13, jq Tutorial
07		10/10	Midterm Review	10/12	Midterm	
08	06	10/17	Time series	10/19	Pandas time series	DSH Ch. 3 (188–207)
09	07	10/24	Text analysis	10/26	Awk, Python re	P4E Ch. 11
10	08	10/31	Web scraping	11/02	Scrapy	Scrapy Tutorial
11	09	11/07	Web APIs	11/09	cURL, HTTPie	HTTPie Docs, What is Mastodon?
12	10	11/14	Bias in data	11/16	Practical session	Bias on the Web
13		11/21	Practical session	11/23	Thanksgiving	
14	11	11/28	Data at scale	11/30	Dask	Dask Tutorial
15	12	12/05	Beyond INST447	12/07	Exam review	The challenges of data in future ...
16		12/12	Reading Day	12/14	Final Exam	

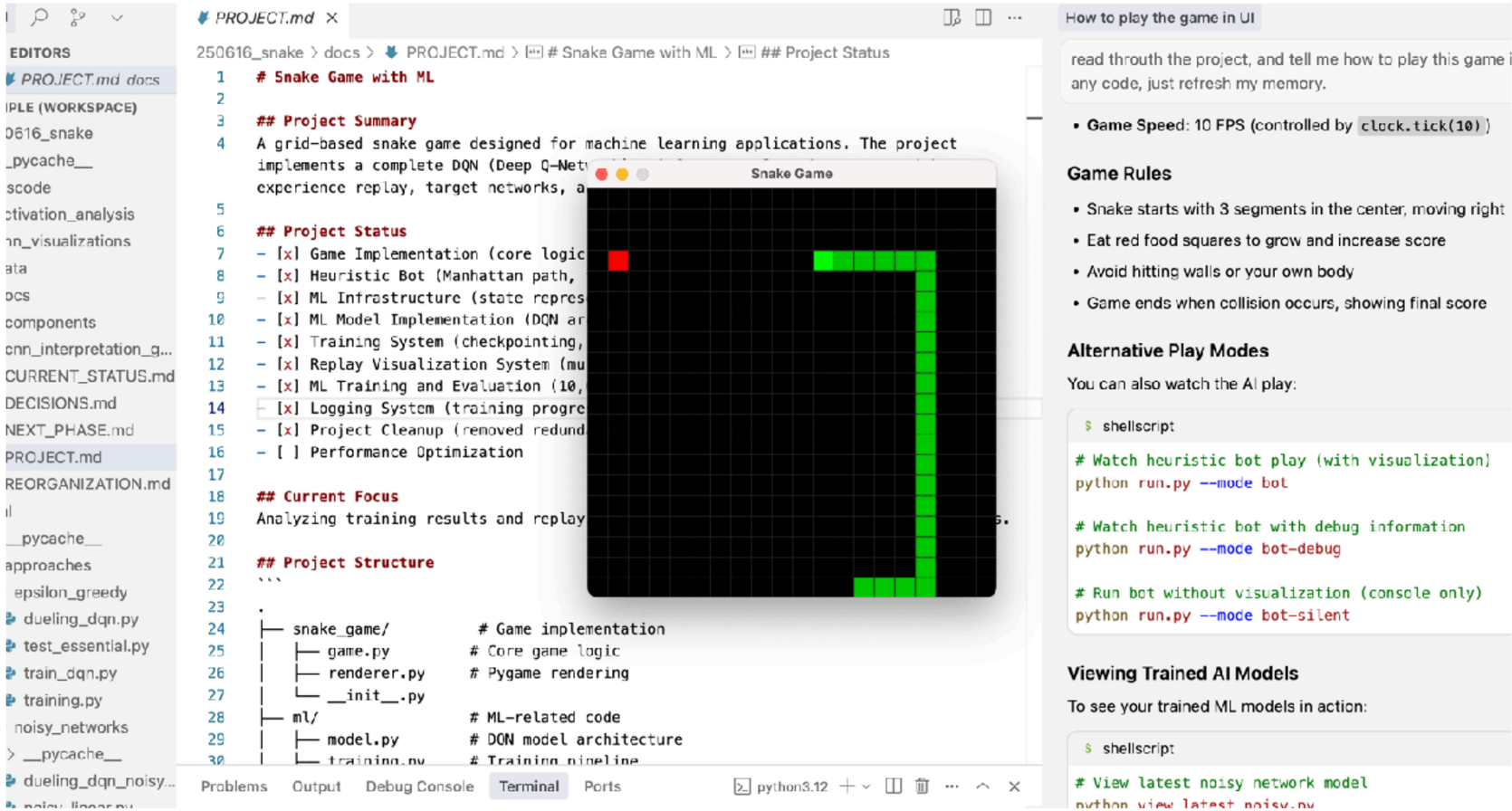
But we all know..
There is an elephant in the classroom.


MANUAL
DATA CLEANING
ChatGPT
Completed and
Learning model


Merging DataFrames
AI

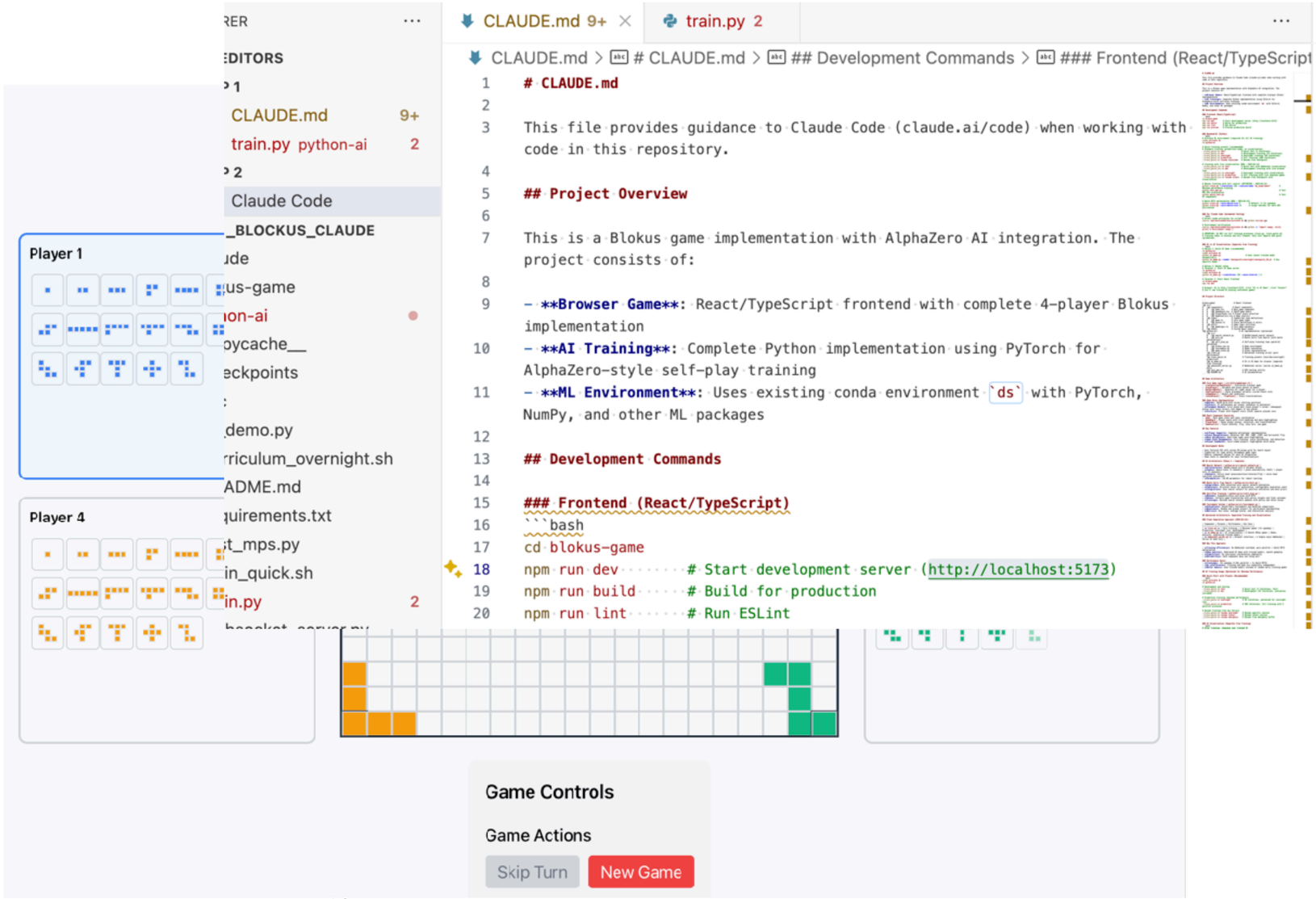
I tried Vibe Coding this summer

- A snake game and a AI game bot trained from scratch.
  - Deep Q Network
  - Epsilon-Greedy exploration
  - Noisy Network
  - With help from Cursor



I tried Vibe Coding this summer

- A snake game and a AI game bot trained from scratch.
- A blokus game and an AI
  - Monte-Carlo Tree Search (MCTS)
  - With help from Claude Code



I tried Vibe Coding this summer

- A snake game and a AI game bot trained from scratch.
- A blokus game and an AI
- A RoBERTa/DeBERTa based text classification for a data science competition
  - With help from Claude Code

EDUCATION

AIAI (Artificial Intelligence for Advancing Instruction) Challenge - Phase 1

Classify instructional activities using multimodal classroom data.

$70,000 in prizes · Completed Aug 2025 · 28 joined

Navigation

- Home
- Problem description
- Accessing data
- About
- Official rules
- Leaderboard results

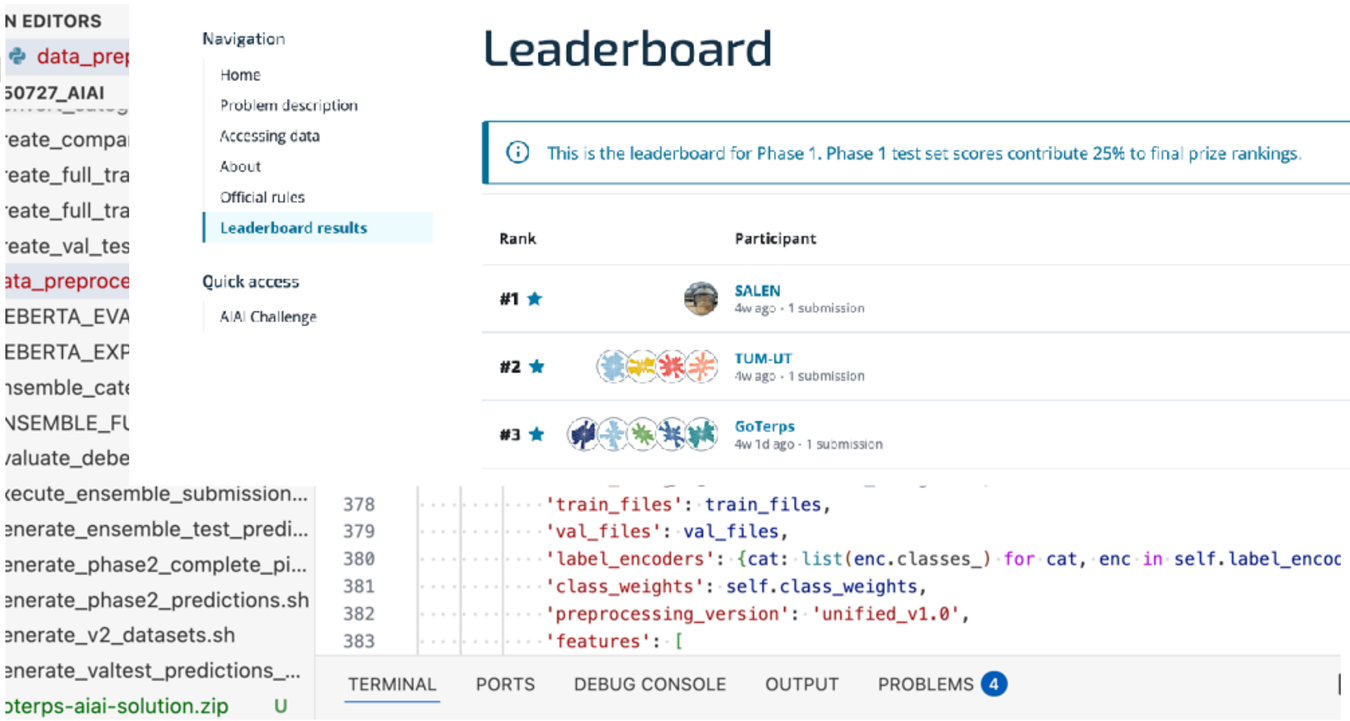
Quick access

- AIAI Challenge

Leaderboard

This is the leaderboard for Phase 1. Phase 1 test set scores contribute 25% to final prize rankings.

Rank	Participant	Best private Macro F1	Shared work
#1	SALEN 4w ago · 1 submission	0.5827	
#2	TUM-UT 4w ago · 1 submission	0.5283	
#3	GoTerps 4w 1d ago · 1 submission	0.4283	



What I learned from my vibe coding experience

- You need to have a good taste.
- It is very easy to loose track once the project get past 10k lines.

- Vibe coding is only as good as your own programming skill is

How does that affect our professional career?

- Senior programmer got a boost, with AI assistant taking on junior programmer’s work.

- Junior programmer lost chance to practice and get the experience and taste.

In this course, we should embrace AI

- Level 1: AI as a coding assistant
  - e.g. How could I change the name of this column?
  - e.g. How could I pivot the DataFrame into this format?

- Level 2: AI as a data processor (via LLM APIs)
  - e.g. Extract the name entity (people, places, etc.) in each Tweet of this dataset.
  - e.g. Augment this congress speech dataset with the NYT headline of the day

- Level 3: AI as an analysis pipeline
  - Brainstorm, planning, and auditing

Part 1 - The Pandas Toolkit

- Focus: Mastering the core verbs of data manipulation. AI is used as a **coding assistant** to generate, debug, and explain code.

Class #	Date	Topic
1	09/04	Introduction, course workflow & pipeline
2	09/11	Filtering & transformation
3	09/18	Grouping, joining & reshaping
4	09/25	Visualization for validation; thinking at scale (map-reduce)
5	10/02	Working with text: string methods & practical regex

Part 2 - Acquiring & Structuring Data

- Focus: Getting data from real-world sources. AI is used as a **coding assistant** for parsing and as a **data processor** via LLM APIs.

Class #	Date	Topic
6	10/09	Semi-structured data: JSON/XML
7	10/16	APIs & web data
8	10/23	AI for data processing
9	10/30	Data representation abstractions

Part 3: The Data Analysis Workflow

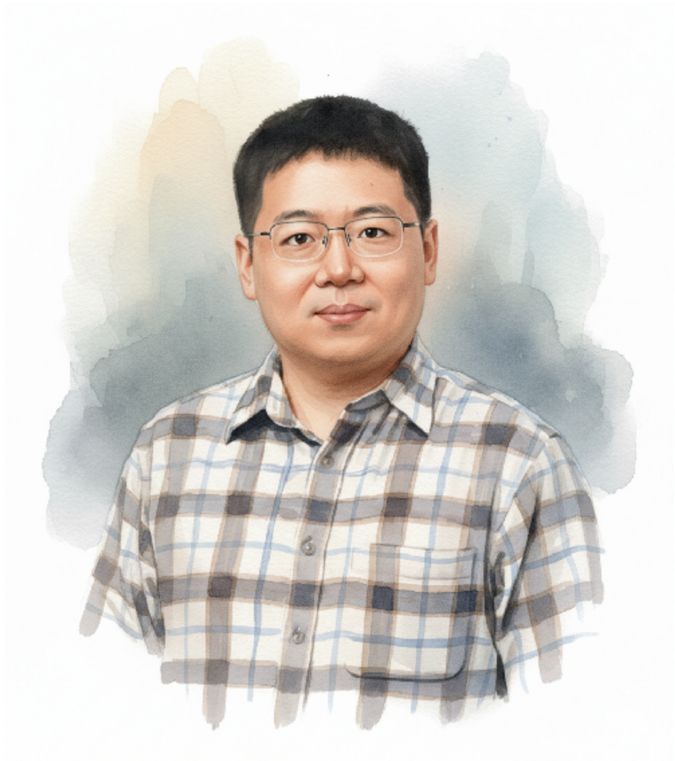
- Focus: Synthesizing skills into an end-to-end process. AI is used as an **analysis partner** for brainstorming, planning, and auditing.

Class #	Date	Topic
10	11/06	Practical vibe coding: iterative exploration with AI
11	11/13	The data quality pipeline: missing data, outliers, dedupe, etc.
12	11/20	Responsible analysis: brief bias/ethics audit & documentation
	11/27	(Thanksgiving break)
13	12/04	Advanced topics TBD
14	12/11	Course synthesis & reflection

Course Structure

- **Lectures**: Conceptual foundations and live demonstrations
  - Grading: in-class quizzes
- **Labs**: Guided, hands-on practice applying the previous lecture's concepts.
  - Grading: lab notebook
- **Programming Exercises**: Short, more rigorous checkups that consolidate recent skills.
- **Take-Home Midterm**: Practical, windowed assessment. (Date TBD)
- **Final Project**: An end-to-end analysis where you source/prepare data, explore it, and produce a write-up; responsible use of AI tools is *encouraged* with appropriate validation.

Instructional Team



Wei Ai
Instructor

Hassan Edwan
Graduate TA

Chenyue Pan
Graduate TA

?
UTA

Grading Policy

- Weekly Labs: 30%
- In-Lecture Quizzes: 10%
- Programming Assignments (4): 20%
- Take-Home Midterm: 20%
- Final Project: 20%

Letter Grade Policy

A+	97-100*	B+	87-89.99	C+	77-79.99	D+	67-69.99	F	0-59.99
A	93-96.99	B	83-86.99	C	73-76.99	D	63-66.99		
A-	90-92.99	B-	80-82.99	C-	70-72.99	D-	60-62.99		

To receive an A+ you must have demonstrated significant contributions to the class in addition to achieving this numeric grade. We reserve the right to curve grades upward (but will not curve grades downward).

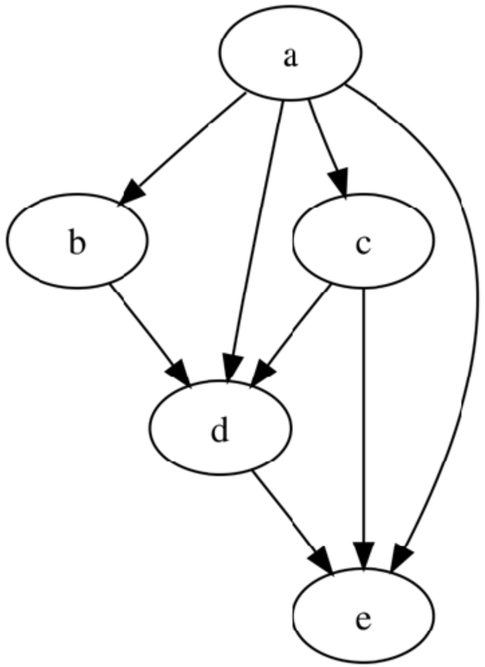
Any Questions?

IDE

- Local IDE - Visual Studio Code, Cursor, PyCharm, Browser
- Cloud IDE - Google CoLab, GitHub codespaces, etc.

Practical Principles to make Data Science Pipeline Reproducible

- Data analysis is a directed acyclic graph
  - Move data through reproducible, rerunnable steps in one direction. Anyone should be able to rerun your analysis using only your code + the raw data
  - Raw data is immutable. Never change your raw data!



.ipynb vs. .py

- Notebooks are for exploration, source code is for repetition
- Jupyter notebooks are great for exploration & storytelling. They are *not* great for code automation, code reviews, and common tasks
- Refactor any code you'll use multiple times into source code, eg. .py files.
- Notebooks are also hard for *vibe coding*

Directory Structure (an Example)

```
├── LICENSE            <- Open-source license if one is chosen
├── Makefile           <- Makefile with convenience commands like `make data` or `make train`
├── README.md          <- The top-level README for developers using this project.
├── data
│   ├── external       <- Data from third party sources.
│   ├── interim        <- Intermediate data that has been transformed.
│   ├── processed      <- The final, canonical data sets for modeling.
│   └── raw            <- The original, immutable data dump.
├── docs               <- A default mkdocs project; see www.mkdocs.org for details
├── models             <- Trained and serialized models, model predictions, or model summaries
├── notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
│                         the creator's initials, and a short `-` delimited description, e.g.
│                         `1.0-jqp-initial-data-exploration`.
├── pyproject.toml     <- Project configuration file with package metadata for
│                         {{ cookiecutter.module_name }} and configuration for tools like black
├── references         <- Data dictionaries, manuals, and all other explanatory materials.
├── reports            <- Generated analysis as HTML, PDF, LaTeX, etc.
│   └── figures        <- Generated graphics and figures to be used in reporting
├── requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
│                         generated with `pip freeze > requirements.txt`
├── setup.cfg          <- Configuration file for flake8
└── {{ cookiecutter.module_name }}   <- Source code for use in this project.
    ├── __init__.py             <- Makes {{ cookiecutter.module_name }} a Python module
    ├── config.py               <- Store useful variables and configuration
    ├── dataset.py              <- Scripts to download or generate data
    ├── features.py             <- Code to create features for modeling
    ├── modeling
    │   ├── __init__.py
    │   ├── predict.py          <- Code to run model inference with trained models
    │   └── train.py            <- Code to train models
    └── plots.py                <- Code to create visualizations
```