LAB

Python for Machine Learning

tinyurl.com/4664-python-lab

James Weichert January 30, 2025

James Weichert

he/him

M.S. Student @ VT ML Lab

- I’m a second-year M.S. student at VT advised by Dr. Eldardiry
- **TA for CS 4664**
- My research focuses on **AI ethics and policy** and I have an interest in CS/AI education
- I’m a big fan of cute dogs

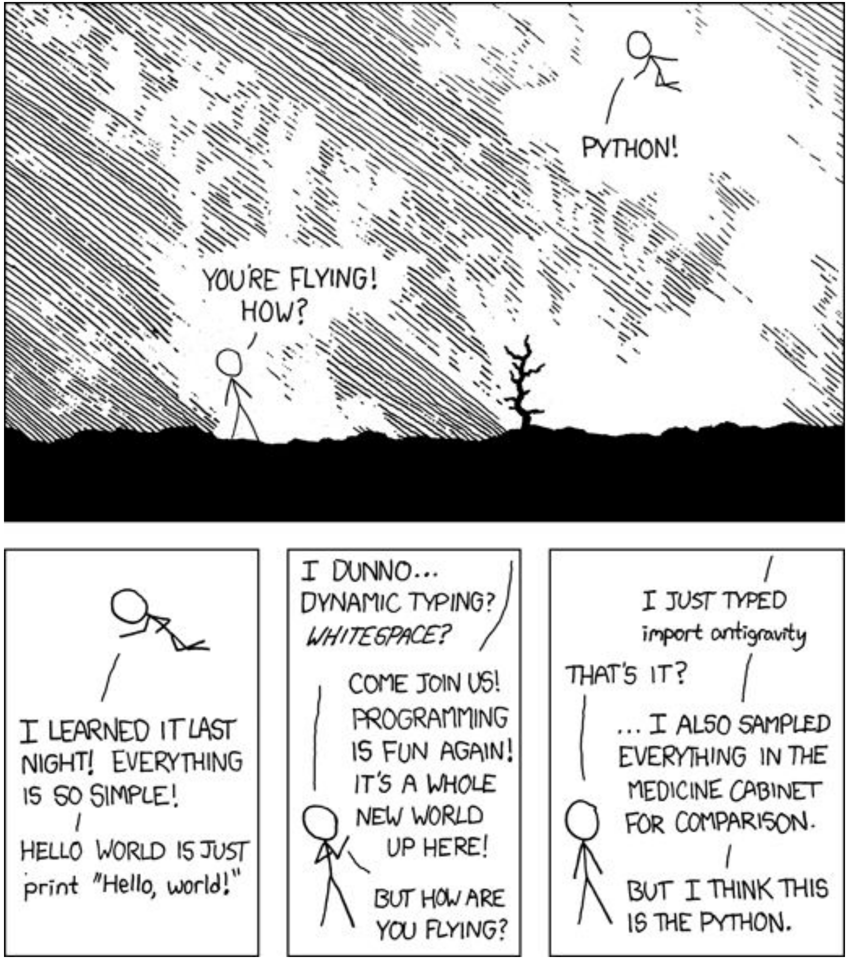
Course Logistics

- **Finalize project teams by tomorrow (1/31)**
  - Once finalized, add team members to "Teams" spreadsheet

- **Project Pitches**
  - Project pitch presentations on 2/11 and 2/13
  - Presentation slides due 2/6
  - Details on Canvas

- **Assignment 1 Released**
  - Due 2/13
  - Writing/reflection assignment
  - Details on Canvas

James' Office Hours start 2/4
Tuesdays 2-3 PM, D&DS 275


YOU'RE FLYING! HOW?
PYTHON!
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

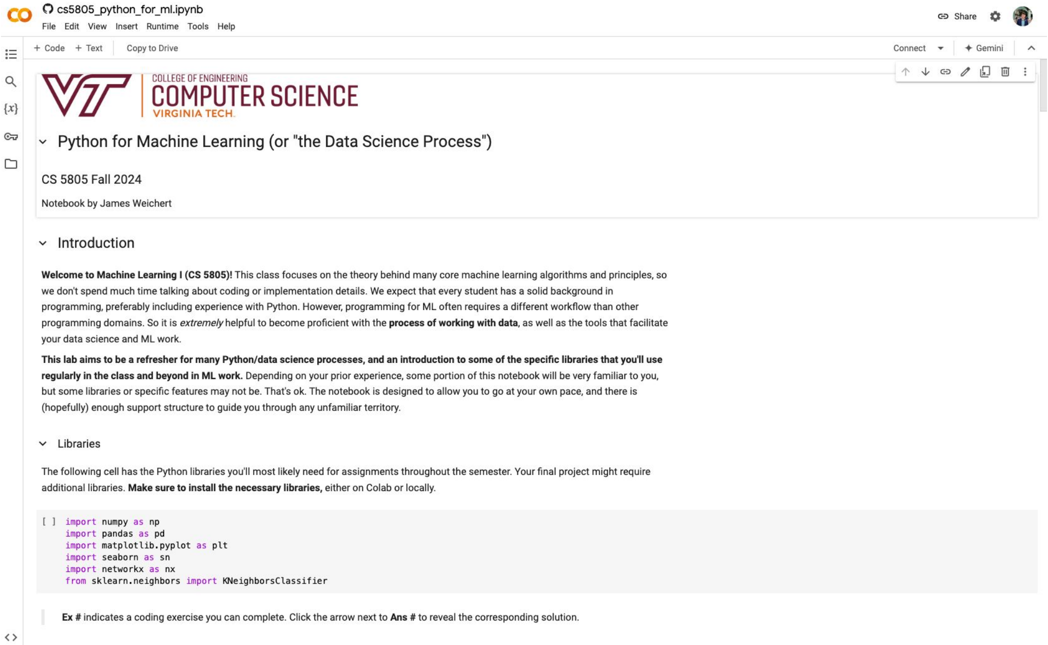
xkcd 353

Python Notebooks

Python notebooks are self-contained interactive Python development environments that combine **markdown** and **code cells** to organize your work.

Some Benefits:

- **Each cell runs independently,** so you can see the output of small code chunks
- Display **dataframes**, **figures**, **graphs** inline
- **Annotate your work** with markdown cells, with easy **Latex integration!**
- **Multiple IDE options:**
  - Google Colab
  - Locally (`pip install notebook`)
  - Visual Studio Code Extension



Python for Machine Learning (or "the Data Science Process")

CS 5805 Fall 2024

Notebook by James Weichert

Introduction

Welcome to Machine Learning I (CS 5805)! This class focuses on the theory behind many core machine learning algorithms and principles, so we don't spend much time talking about coding or implementation details. We expect that every student has a solid background in programming, preferably including experience with Python. However, programming for ML often requires a different workflow than other programming domains. So it is *extremely* helpful to become proficient with the **process of working with data**, as well as the tools that facilitate your data science and ML work.

This lab aims to be a refresher for many Python/data science processes, and an introduction to some of the specific libraries that you'll use regularly in the class and beyond in ML work. Depending on your prior experience, some portion of this notebook will be very familiar to you, but some libraries or specific features may not be. That's ok. The notebook is designed to allow you to go at your own pace, and there is (hopefully) enough support structure to guide you through any unfamiliar territory.

Libraries

The following cell has the Python libraries you'll most likely need for assignments throughout the semester. Your final project might require additional libraries. **Make sure to install the necessary libraries,** either on Colab or locally.

```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sn
    import networkx as nx
    from sklearn.neighbors import KNeighborsClassifier
```

> **Ex #** indicates a coding exercise you can complete. Click the arrow next to **Ans #** to reveal the corresponding solution.

The “Data Science Process”

Explore

- **Get & import** your data
- **Understand** the data’s structure, features
- **Exploratory Data Analysis**
    - *What relationships are worth investigating?*

Transform

- **Clean** the data
    - *How should I deal with missing data?*
- **Feature Extraction**
    - *What information is most valuable?*

Apply

- **Train model** on selected features
- **Cross-validate** on different data splits
- **Test** on unseen data
- **Evaluate** results and generalizability

But First...

Part 0

Working with Data

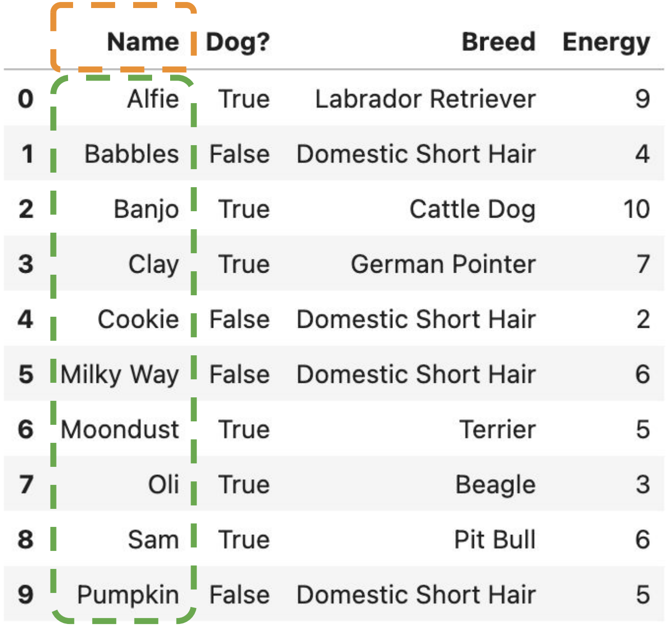
0. Working with Data

What are data?

- Data are **information**
- Data are **information *structured in a consistent manner***

Tables!

Tables consist of **arrays** and associated **labels**

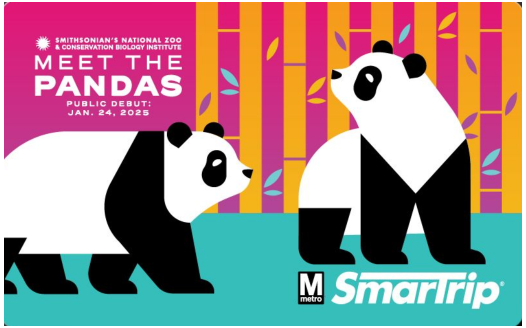


	Name	Dog?	Breed	Energy
0	Alfie	True	Labrador Retriever	9
1	Babbles	False	Domestic Short Hair	4
2	Banjo	True	Cattle Dog	10
3	Clay	True	German Pointer	7
4	Cookie	False	Domestic Short Hair	2
5	Milky Way	False	Domestic Short Hair	6
6	Moondust	True	Terrier	5
7	Oli	True	Beagle	3
8	Sam	True	Pit Bull	6
9	Pumpkin	False	Domestic Short Hair	5

pandas

pandas (not those ones) is a **Python library for working with data tables** (called “dataframes”).

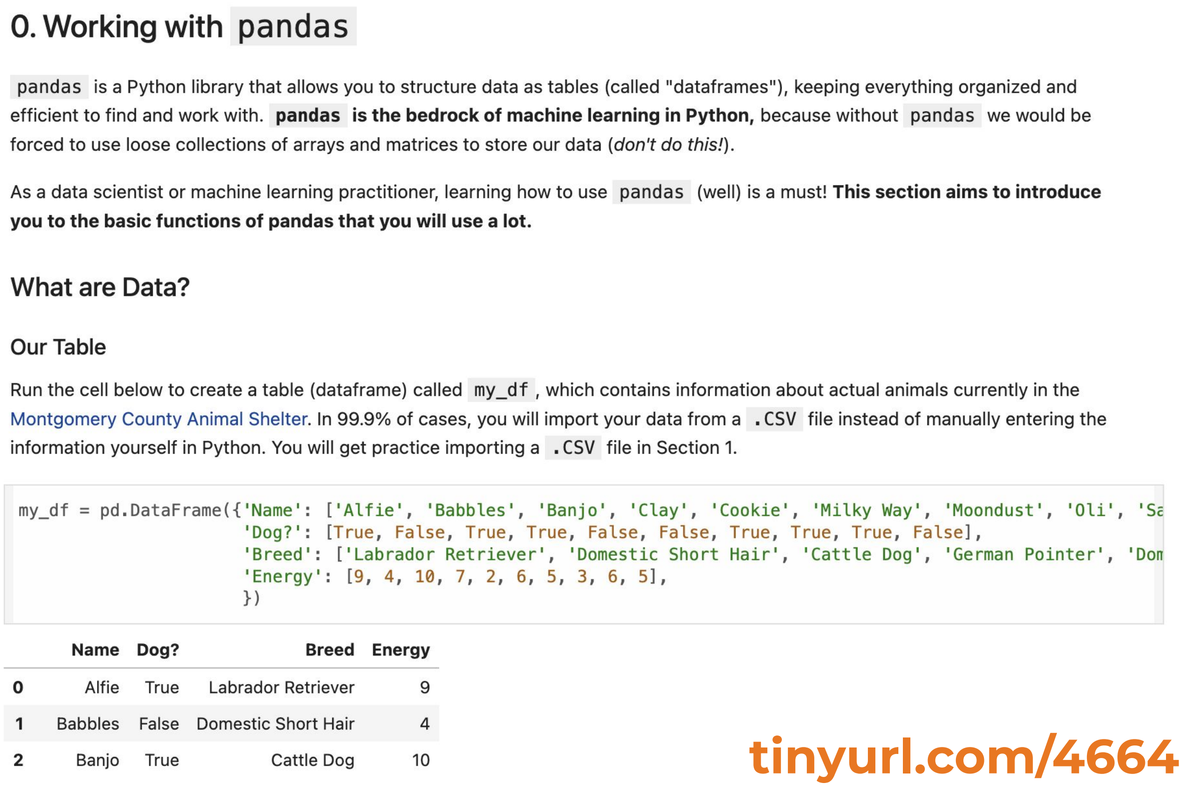
pandas tables consist of **columns** (arrays), each with a **label**.


SMITHSONIAN'S NATIONAL ZOO & CONSERVATION BIOLOGY INSTITUTE
MEET THE PANDAS
PUBLIC DEBUT: JAN. 24, 2025
M metro
SmarTrip

What can we do with pandas?

- Extract **column(s)** or **row(s)**
- **Sort** or **group by** values in columns
- **Filter** rows by a condition
- **Apply** functions to entire columns

Trying it out



0. Working with `pandas`

`pandas` is a Python library that allows you to structure data as tables (called "dataframes"), keeping everything organized and efficient to find and work with. **`pandas` is the bedrock of machine learning in Python,** because without `pandas` we would be forced to use loose collections of arrays and matrices to store our data (*don't do this!*).

As a data scientist or machine learning practitioner, learning how to use `pandas` (well) is a must! **This section aims to introduce you to the basic functions of pandas that you will use a lot.**

What are Data?

Our Table

Run the cell below to create a table (dataframe) called `my_df`, which contains information about actual animals currently in the Montgomery County Animal Shelter. In 99.9% of cases, you will import your data from a `.CSV` file instead of manually entering the information yourself in Python. You will get practice importing a `.CSV` file in Section 1.

```
my_df = pd.DataFrame({'Name': ['Alfie', 'Babbles', 'Banjo', 'Clay', 'Cookie', 'Milky Way', 'Moondust', 'Oli', 'Sa
                      'Dog?': [True, False, True, True, False, False, True, True, True, False],
                      'Breed': ['Labrador Retriever', 'Domestic Short Hair', 'Cattle Dog', 'German Pointer', 'Dom
                      'Energy': [9, 4, 10, 7, 2, 6, 5, 3, 6, 5],
                      })
```

	Name	Dog?	Breed	Energy
0	Alfie	True	Labrador Retriever	9
1	Babbles	False	Domestic Short Hair	4
2	Banjo	True	Cattle Dog	10

tinyurl.com/4664-python-lab

The “Data Science Process”

1. Understanding Your Data

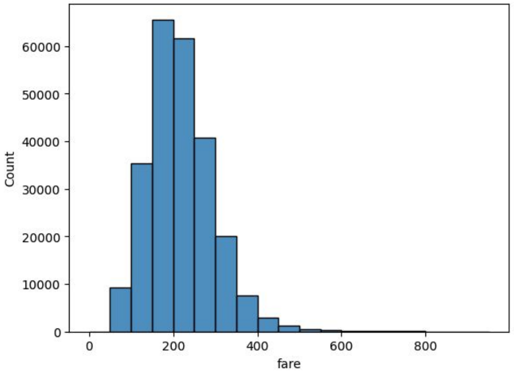
Don’t do ML right away! Take time to **explore** and **understand** the data you’re working with.

What might I need to know about my data?

Exploratory Data Analysis

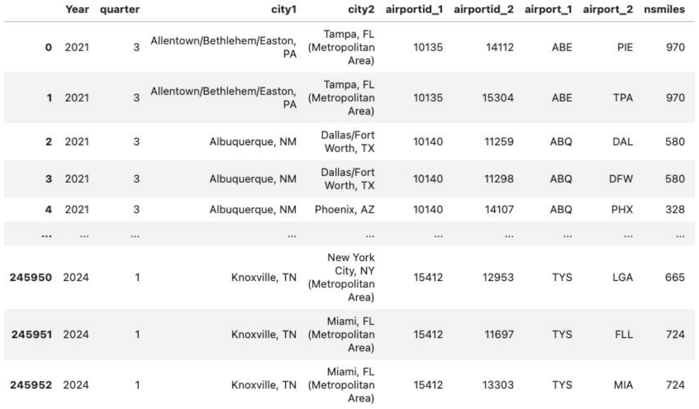
Why do EDA?

- Better **understand** your data
- Identify **missing data**, other issues
- Discover **relationships**
- Easily **test hypotheses**

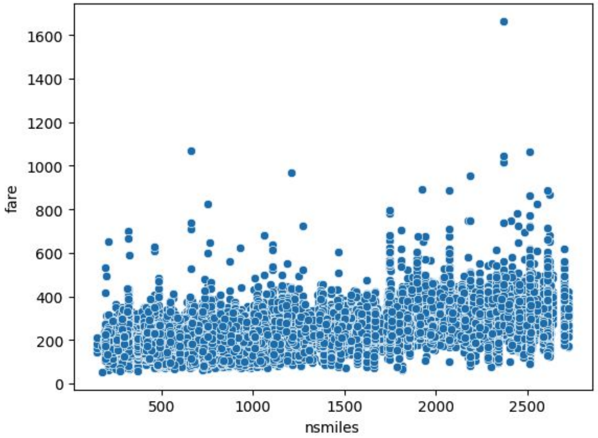


Our Data

US airline route data 1993-2024 from Kaggle



	Year	quarter	city1	city2	airportid_1	airportid_2	airport_1	airport_2	nsmiles
0	2021	3	Allentown/Bethlehem/Easton, PA	Tampa, FL (Metropolitan Area)	10135	14112	ABE	PIE	970
1	2021	3	Allentown/Bethlehem/Easton, PA	Tampa, FL (Metropolitan Area)	10135	15304	ABE	TPA	970
2	2021	3	Albuquerque, NM	Dallas/Fort Worth, TX	10140	11259	ABQ	DAL	580
3	2021	3	Albuquerque, NM	Dallas/Fort Worth, TX	10140	11298	ABQ	DFW	580
4	2021	3	Albuquerque, NM	Phoenix, AZ	10140	14107	ABQ	PHX	328
...	...	...	...	...	...	...	...	...	...
245950	2024	1	Knoxville, TN	New York City, NY (Metropolitan Area)	15412	12953	TYS	LGA	665
245951	2024	1	Knoxville, TN	Miami, FL (Metropolitan Area)	15412	11697	TYS	FLL	724
245952	2024	1	Knoxville, TN	Miami, FL (Metropolitan Area)	15412	13303	TYS	MIA	724



2. Transforming the Data

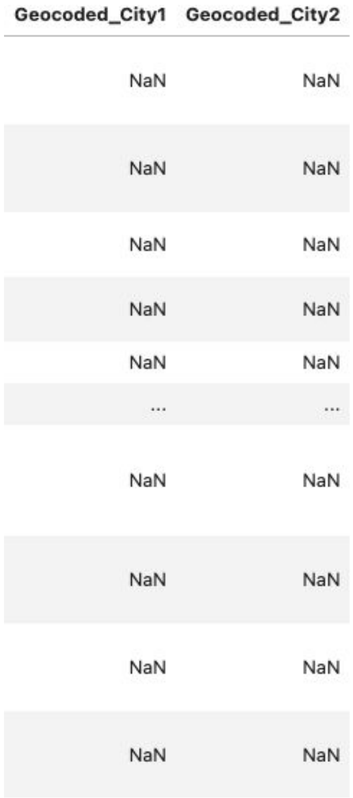
Data are messy. Almost always, your dataset **won't be useable 'out of the box,'** You'll need to spend time on **data cleaning** and **feature extraction.**

Do I need more data than I have?

Data Cleaning

How should I clean my dataset?

- **Delete the row** if cell data is missing
- **Remove the column** with missing data
- Treat the missing data as an **exception**
- **Infer** the missing data



Geocoded_City1	Geocoded_City2
NaN	NaN
NaN	NaN
NaN	NaN
NaN	NaN
NaN	NaN
...	...
NaN	NaN
NaN	NaN
NaN	NaN
NaN	NaN

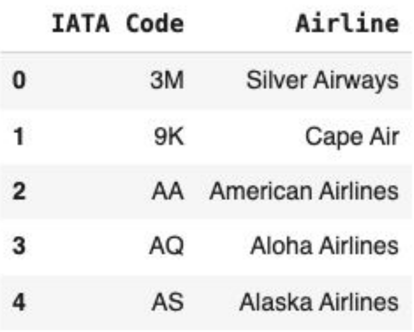
Feature Extraction

Why bother?

If your dataset already has the features you want, great! But **more often than not, you'll want to modify or add columns to improve training.** Feature extraction is sometimes necessary to **provide the 'context' that you implicitly know about the data**, but that an algorithm does not.

For example, a house price estimator might need an additional column containing the population of the city the house is located in. Otherwise, the estimator can't differentiate between, for example, a 3-bedroom standalone house in Blacksburg and a 3-bedroom standalone house in Washington, D.C.

$$\%_{\text{diff}} = \frac{\text{avg fare} - \text{low fare}}{\text{avg fare}}$$



	IATA Code	Airline
0	3M	Silver Airways
1	9K	Cape Air
2	AA	American Airlines
3	AQ	Aloha Airlines
4	AS	Alaska Airlines

3. Applying a Model

Now **apply ML. Train your model** with the features you chose. Make sure to **(cross) validate!** And remember that more **complex isn't always better.**

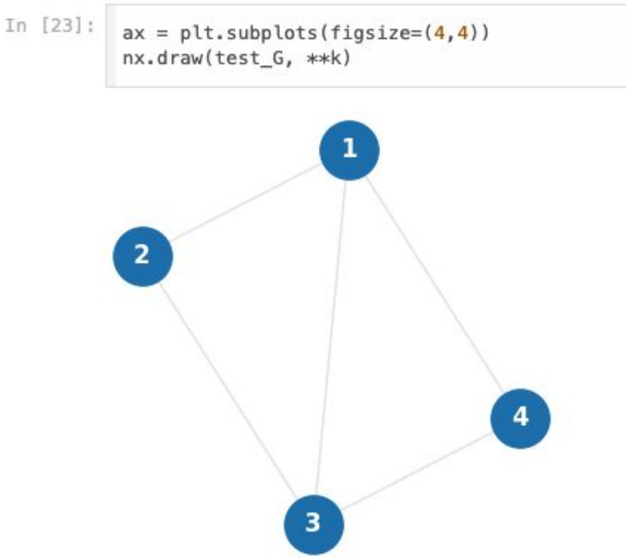
What is my model capable of?

Graphs

Assignment 4 will deal with graph data, so it is helpful to be somewhat familiar with the **networkx** library.

Key Functionality:

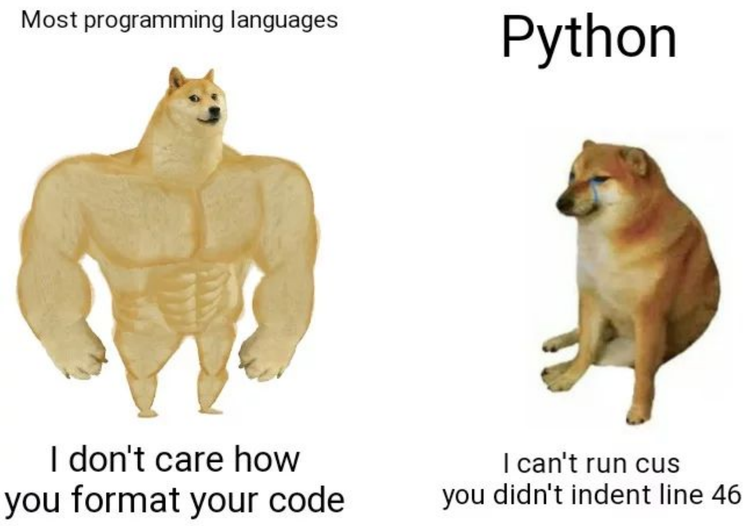
- Create a new graph: **nx.Graph()**
- Add nodes: **Graph.add_nodes_from(itr)**
- Add edges: **Graph.add_edges_from([(a,b),(b,c),...])**
- Add weighted edges:
  Graph.add_weighted_edges_from([(a,b,w),...])
- Draw graph: **nx.draw(graph)**



```
In [23]:  ax = plt.subplots(figsize=(4,4))
          nx.draw(test_G, **k)
```

General Tips

- **Use classes!**
- **Variable assignment shortcuts:**
  - Incrementing/decrementing (`+=` , `-=` , `*=`)
  - Multiple variable assignment
- `zip(itr1, itr2)` is useful in for-loops
- `int(True) → 1` and `int(False) → 0`
- **Scientific notation:** `1.2e10`


Most programming languages
Python
I don't care how you format your code
I can't run cus you didn't indent line 46

PYTHON FOR ML

Questions?