QSS20: Modern Statistical Computing

Unit 12: Web-scraping, Part I

Goals for today

- Recap of supervised machine learning
- Lecture on web-scraping
  - Living in a digital era
  - Researching the digital era
  - How to scrape/crawl without reinventing the flat tire
  - Intro to packages for wrangling web data
  - Ethics (briefly)
- Notebook for basic web-scraping
  - Making `requests`
  - Parsing HTML
  - Scraping URLs

Goals for today

- **Recap of supervised machine learning**
- Lecture on web-scraping
  - Living in a digital era
  - Researching the digital era
  - How to scrape/crawl without reinventing the flat tire
  - Intro to packages for wrangling web data
  - Ethics (briefly)
- Notebook for basic web-scraping
  - Making `requests`
  - Parsing HTML
  - Scraping URLs

Recap of supervised machine learning

What do you remember?

Recap of supervised ML: Steps with code, optimized

1. Preprocess the data: make DTM, do train/test split

```
vec = CountVectorizer(); dtm_sparse = vec.fit_transform(texts)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size)
```

2. Initialize model and parameters to search

```
dt = DecisionTreeClassifier(min_samples_split=5, min_samples_leaf=10)
param_grid = {'min_samples_split': [2,10], 'min_samples_leaf': [2,10]}
```

3. Using training data, select optimal model via gridsearch

```
grid_dt = GridSearchCV(dt, param_grid, cv=3)
grid_dt.fit(X_train, y_train)
best_idx = np.argmax(grid_dt.cv_results_["mean_train_score"]) # get best
grid_dt.cv_results_["params"][best_idx] # best params
```

4. Re-train optimal model with full training data, get predictions

```
best_dt.fit(X_train, y_train)
y_pred = best_dt.predict(X_test)
y_predprob = best_dt.predict_proba(X_test)
```

5. Interpret model to identify important features

```
feat_imp = pd.DataFrame({'feature_imp': best_dt.feature_importances_,
                'feature_name': [col for col in X_train.columns]})
```

When to use which model?

- Depends on type of outcome:
  - If numeric, use OLS and consider fixed-effects if clustered at some second level (e.g., students in schools)
  - If categorical, use logistic regression: standard logit if distinct (multiclass OK), ordinal logit if in sequential 'steps' (use `statsmodels`
- Depends on (theorized) data generation process:
  - If expecting a smooth change in y with X, use some kind of linear regression
  - If wanting to model complex, non-linear relationships, use a tree-based method (e.g., Decision Trees) or deep learning (e.g., BERT)
- Depends on amount of data: more data allows for more complex models

Optimizing ML: terminology

- **overfitting:** When a model is trained on too little data to generalize to new data, usually when trained and tested on the same sample. In this case, the model just repeats the labels of the samples that it has just seen, and is evaluated to have a perfect score–but it fails to predict anything useful on yet-unseen data.
- **underfitting:** When a model fails to adequately capture the underlying structure of the data, usually when trained on too little data
- **cross-validation:** A way to assess the performance of an algorithm on an unseen data set. Essentially this repeats a train-test split several times and averages the result of these independent slices, giving a superior estimation of model accuracy compared to a single train-test split.
- **parameter:** An algorithm setting that may be selected for performance or substantive reasons. If a parameter is not learned directly within an estimator but instead must be set explicitly (passed as an argument), then it's called a *hyper-parameter*.
- **optimization:** Selecting the best element from some set of alternatives such that the result maximizes some criterion (e.g., model accuracy)
- **grid-search:** An exhaustive search over model parameters to discover the optimal configuration, maximizing model performance

Where we are

- Recap of supervised machine learning
- **Lecture on web-scraping**
  - **Living in a digital era**
  - Researching the digital era
  - How to scrape/crawl without reinventing the flat tire
  - Intro to packages for wrangling web data
  - Ethics (briefly)
- Notebook for basic web-scraping
  - Making `requests`
  - Parsing HTML
  - Scraping URLs

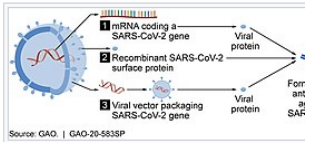
Learn about vaccines: Wikipedia

en.wikipedia.org/wiki/COVID-19_vaccine#Vaccine_types

Vaccine types

As of January 2021, nine different technology platforms – with the technology of numerous candidates remaining undefined – are under research and development to create an effective vaccine against COVID-19.[3][27] Most of the platforms of vaccine candidates in clinical trials are focused on the coronavirus spike protein and its variants as the primary antigen of COVID-19 infection.[27] Platforms being developed in 2020 involved nucleic acid technologies (nucleoside-modified messenger RNA and DNA), non-replicating viral vectors, peptides, recombinant proteins, live attenuated viruses, and inactivated viruses.[14][27][28][34]

Many vaccine technologies being developed for COVID-19 are not like vaccines already in use to prevent influenza, but rather are using "next-generation" strategies for precision on COVID-19 infection mechanisms.[27][28][34] Several of the synthetic vaccines use a 2P mutation to lock the spike protein into its prefusion configuration, stimulating an immune response to the virus before it attaches to a human cell.[90] Vaccine platforms in development may improve flexibility for antigen manipulation and effectiveness for targeting mechanisms of COVID-19 infection in susceptible population subgroups, such as healthcare workers, the elderly, children, pregnant women, and people with existing weakened immune systems.[27][28]



Conceptual diagram showing three vaccine type for forming SARS-CoV-2 proteins to prompt an immune response: (1) RNA vaccine, (2) subunit vaccine, (3) viral vector vaccine

RNA vaccines

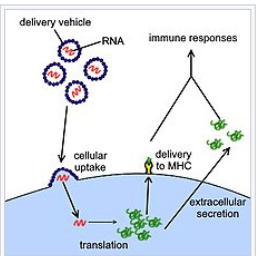
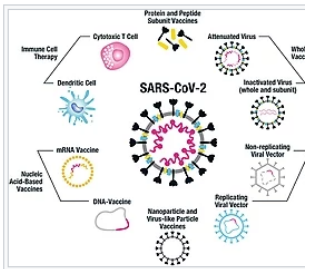


Diagram of the operation of an RNA vaccine. Messenger RNA contained in the vaccine enters cells and is translated into foreign proteins, which trigger an immune response.

An RNA vaccine contains RNA which, when introduced into a tissue, acts as messenger RNA (mRNA) to cause the cells to build the foreign protein and stimulate an adaptive immune response which teaches the body how to identify and destroy the corresponding pathogen or cancer cells. RNA vaccines often, but not always, use nucleoside-modified messenger RNA. The delivery of mRNA is achieved by a coformulation of the molecule into lipid nanoparticles which protect the RNA strands and help their absorption into the cells.[91][92][93][94]

RNA vaccines were the first COVID-19 vaccines to be authorized in the United States and the European Union.[95][96] As of January 2021, authorized vaccines of this type are the Pfizer–BioNTech COVID-19 vaccine[97][98][99] and the Moderna COVID-19 vaccine.[100][101] As of February 2021, the CVnCoV RNA vaccine from CureVac is awaiting authorization in the EU.[102]

Severe allergic reactions are rare. In December 2020, 1,893,360 first doses of Pfizer-BioNTech COVID-19 vaccine administration resulted in 175 cases of severe allergic reaction, of which 21 were anaphylaxis.[103] For 4,041,396 Moderna COVID-19 vaccine dose administrations in December 2020 and January 2021, only 10 cases of anaphylaxis were reported.[103] The lipid nanoparticles were most likely responsible for the allergic reactions.[103]



Vaccine platforms being employed for SARS-Co 2. Whole virus vaccines include both attenuated inactivated forms of the virus. Protein and peptid subunit vaccines are usually combined with an adjuvant in order to enhance immunogenicity. Th main emphasis in SARS-CoV-2 vaccine develop has been on using the whole spike protein in its trimeric form or components of it, such as the RE region. Multiple non-replicating viral vector vacci have been developed, particularly focused on adenovirus; while there has been less emphasis the replicating viral vector constructs.[89]

Adenovirus vector vaccines

These vaccines are examples of non-replicating viral vector vaccines, using an adenovirus shell containing DNA that encodes a SARS-CoV-2 protein.[104][105] The viral vector-based vaccines against COVID-19 are non-replicating, meaning that they do not make new virus particles, but rather produce only the antigen which elicits a systemic immune response.[104]

As of January 2021, authorized vaccines of this type are the Oxford–AstraZeneca COVID-19 vaccine,[106][107][108] the Sputnik V COVID-19 vaccine,[109] Convidecia, and the Johnson & Johnson COVID-19 vaccine.[110][111]

Convidecia and the Johnson & Johnson COVID-19 vaccine are both one-shot vaccines which offer less complicated logistics and can be stored under ordinary refrigeration for several months.[112][113]

The Sputnik V COVID-19 vaccine uses Ad26 for the first dose, which is the same as the Johnson & Johnson vaccine's only dose, and Ad5 for the second dose. Convidecia uses Ad5 for its only dose.[114]

Inactivated virus vaccines

Inactivated vaccines consist of virus particles that have been grown in culture and then are killed using a method such as heat or formaldehyde to lose disease producing capacity, while still stimulating an immune response.[115]

As of January 2021, authorized vaccines of this type are the Chinese CoronaVac,[116][117][118] BBIBP-CorV,[119] and WIBP-CorV; the Indian Covaxin; and the Russian CoviVac.[120] Vaccines in clinical trials include the Valneva COVID-19 vaccine.[121][122]

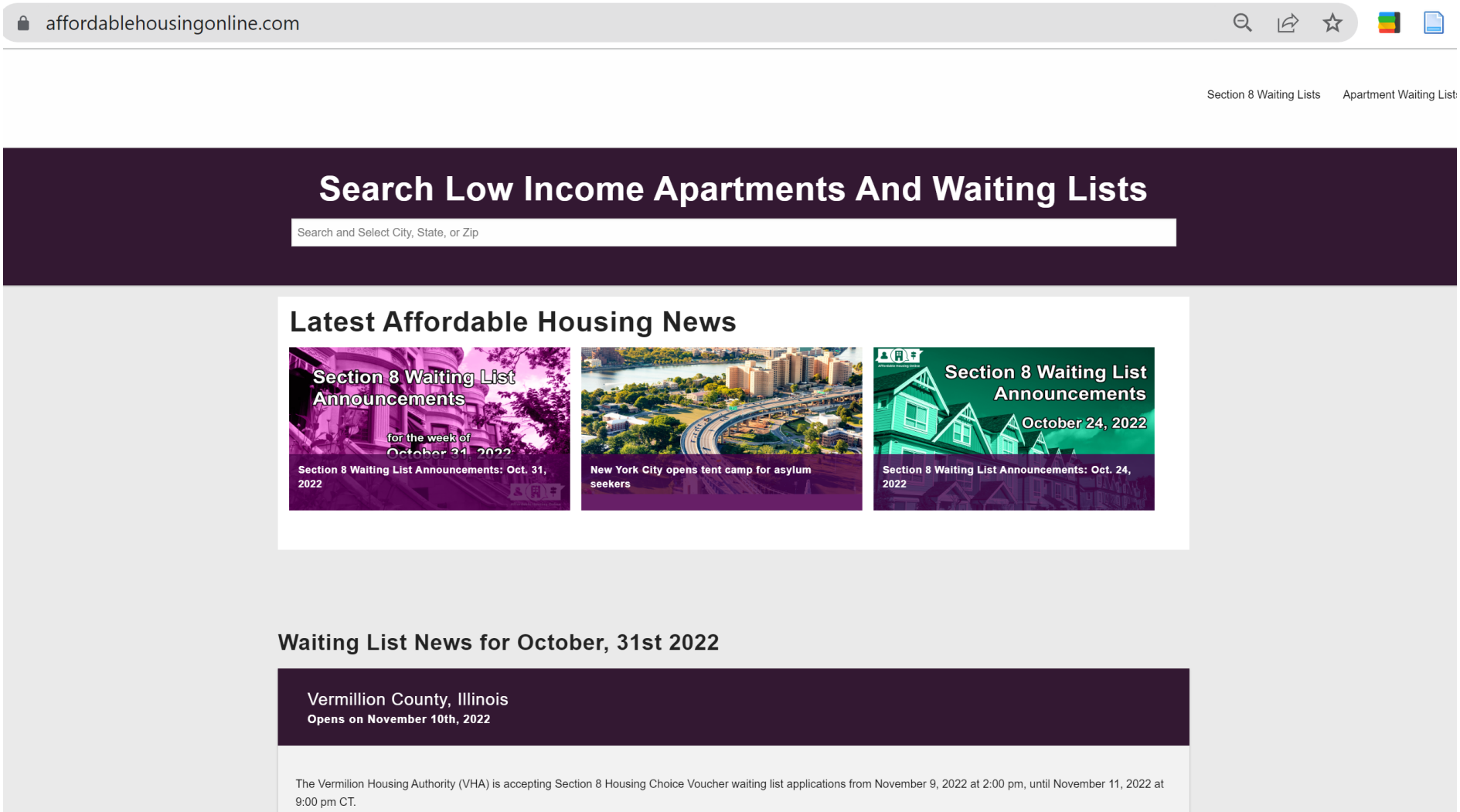
Subunit vaccines

Subunit vaccines present one or more antigens without introducing whole pathogen particles. The antigens involved are often protein subunits, but can be any molecule that is a fragment of the pathogen.[123]

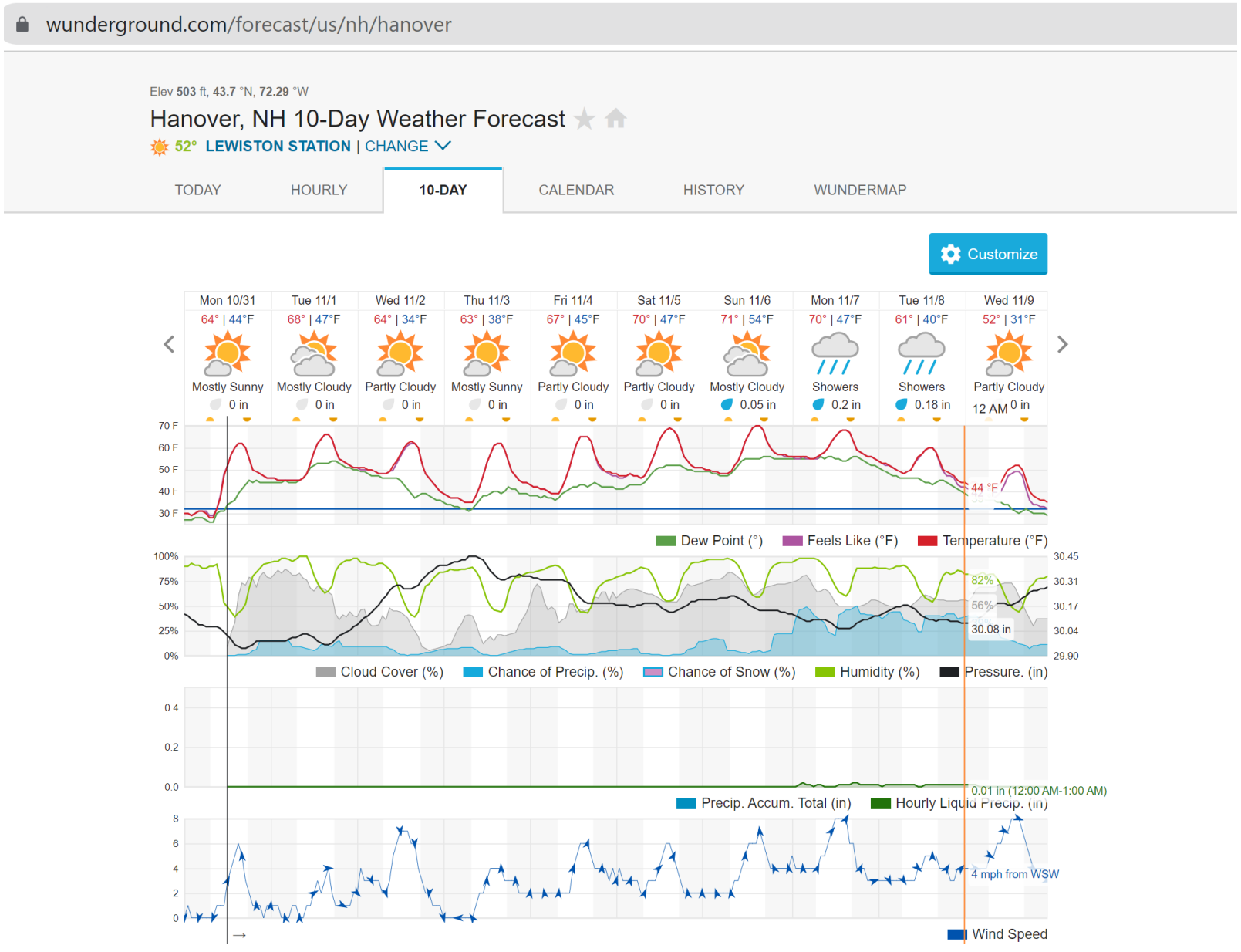
As of April 2021, the two authorized vaccines of this type are the peptide vaccine EpiVacCorona[124] and RBD-Dimer.[3] Vaccines with pending authorizations include the Novavax COVID-19 vaccine,[125] SOBERANA 02 (a conjugate vaccine), and the Sanofi–GSK va The V451 vaccine was previously in clinical trials, which were terminated because it was found that the vaccine may potentially cause incorrect results for subsequent HIV testing.[126][127]

Other types

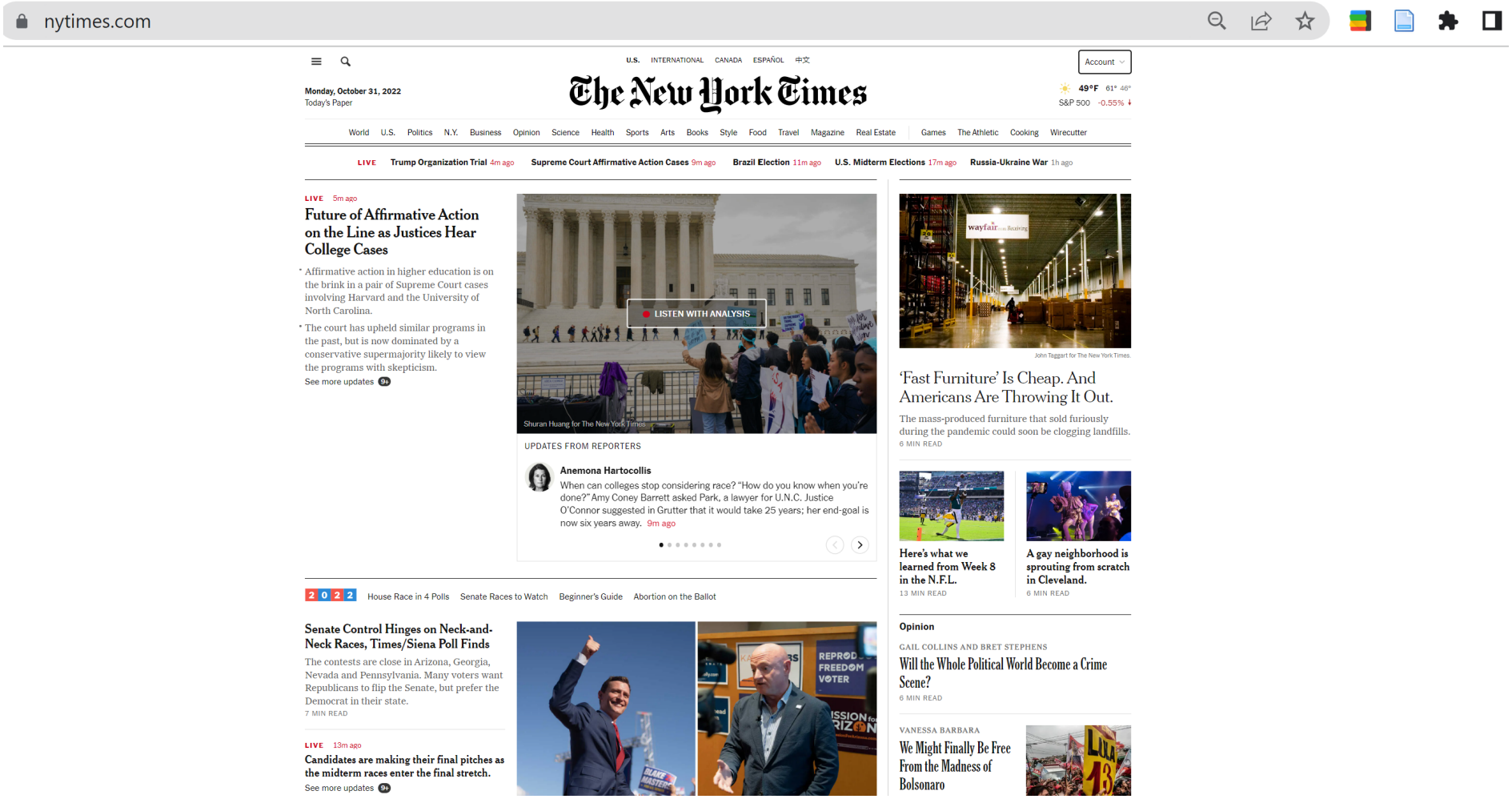
Find housing: Affordable Housing Online



Check weather: wunderground.com



Follow current events: New York Times



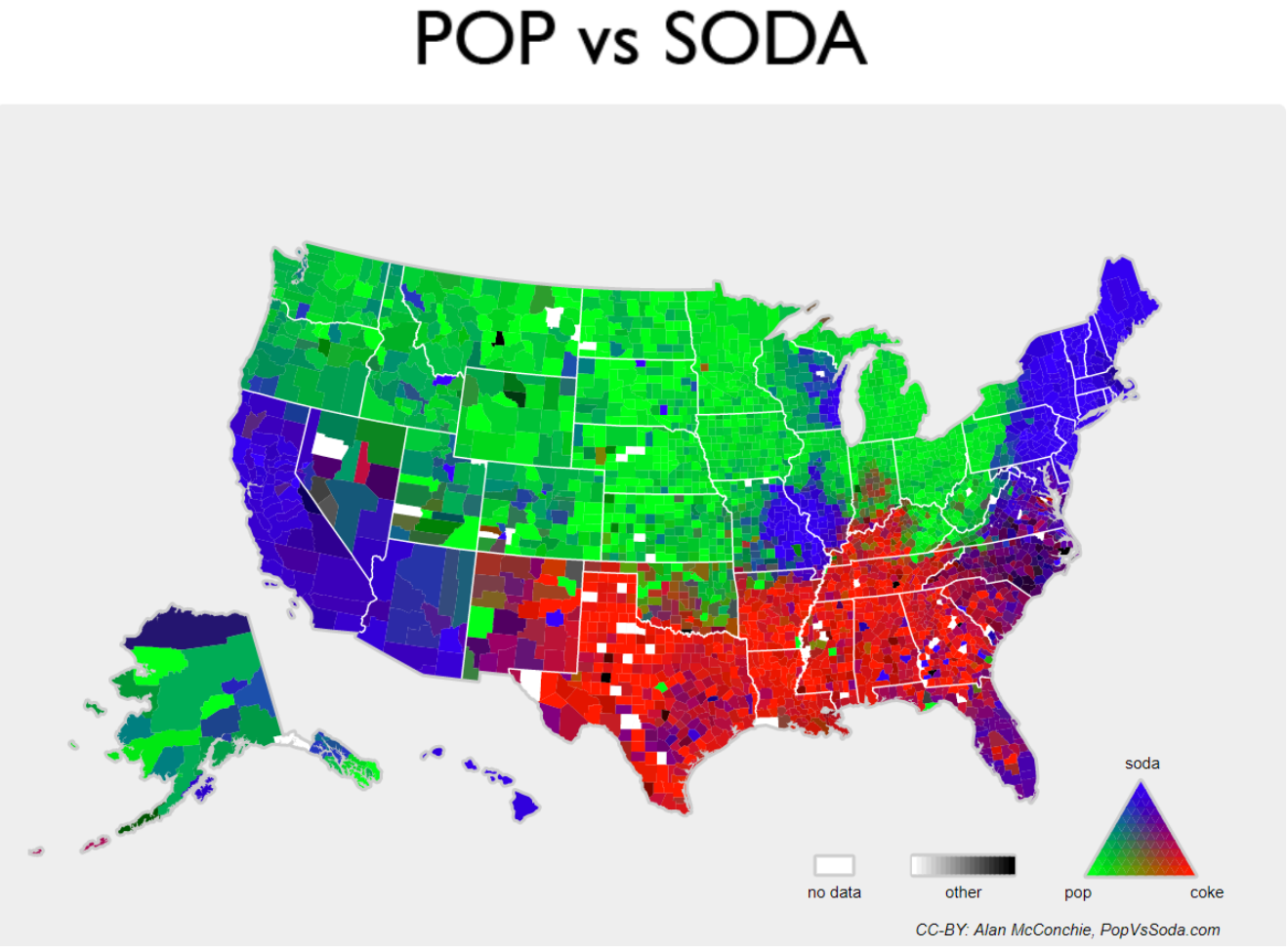
Where we are

- Recap of supervised machine learning
- **Lecture on web-scraping**
  - Living in a digital era
  - **Researching the digital era**
  - How to scrape/crawl without reinventing the flat tire
  - Intro to packages for wrangling web data
  - Ethics (briefly)
- Notebook for basic web-scraping
  - Making `requests`
  - Parsing HTML
  - Scraping URLs

Social media → geography

Input: Tweets

Output:
Latitude,
Longitude

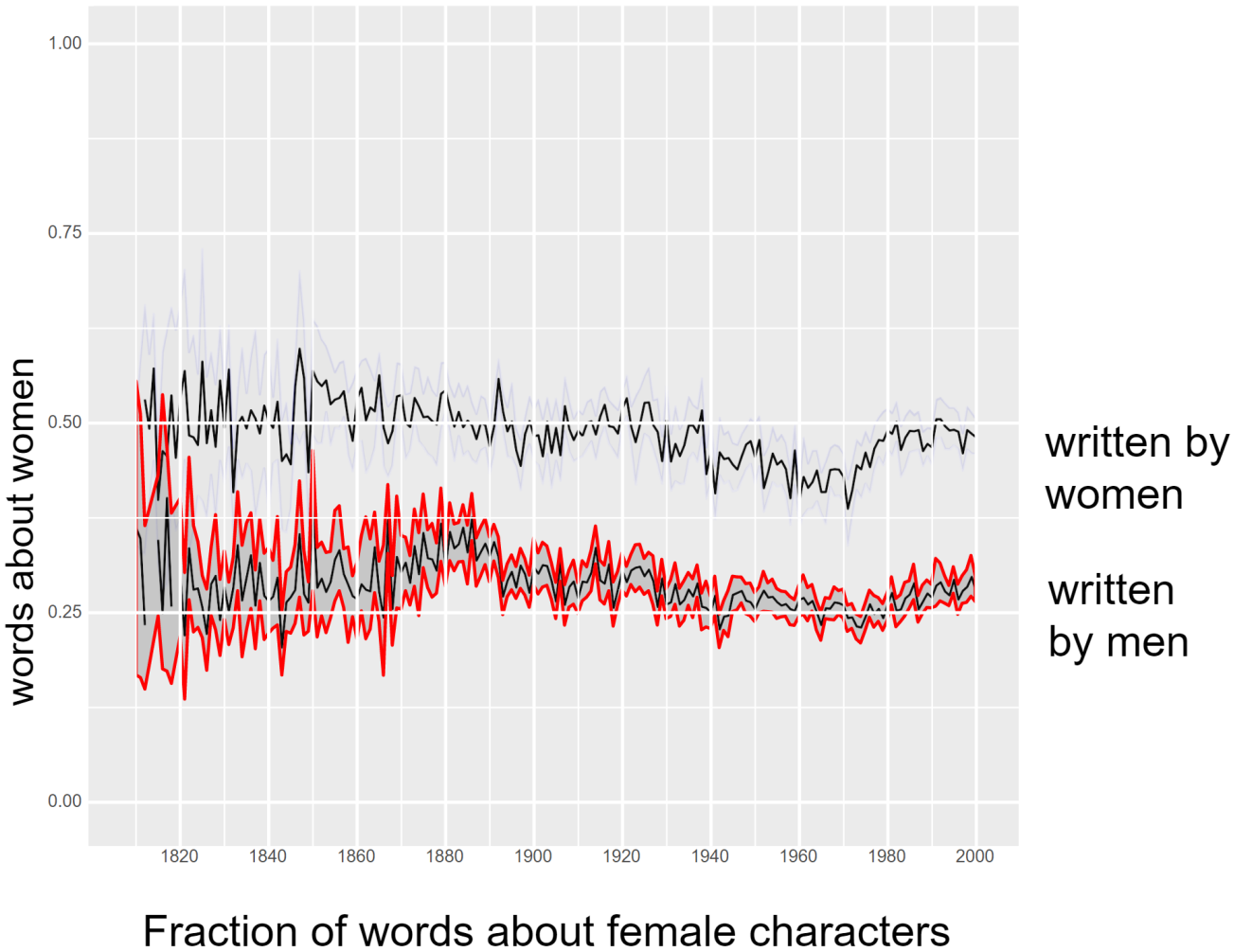


Benjamin Wing and Jason Baldridge (2011), "Simple supervised document geolocation with geodesic grids" (ACL)

Books → gendered language

Input:
Fictional texts

Output:
Prop. words about females

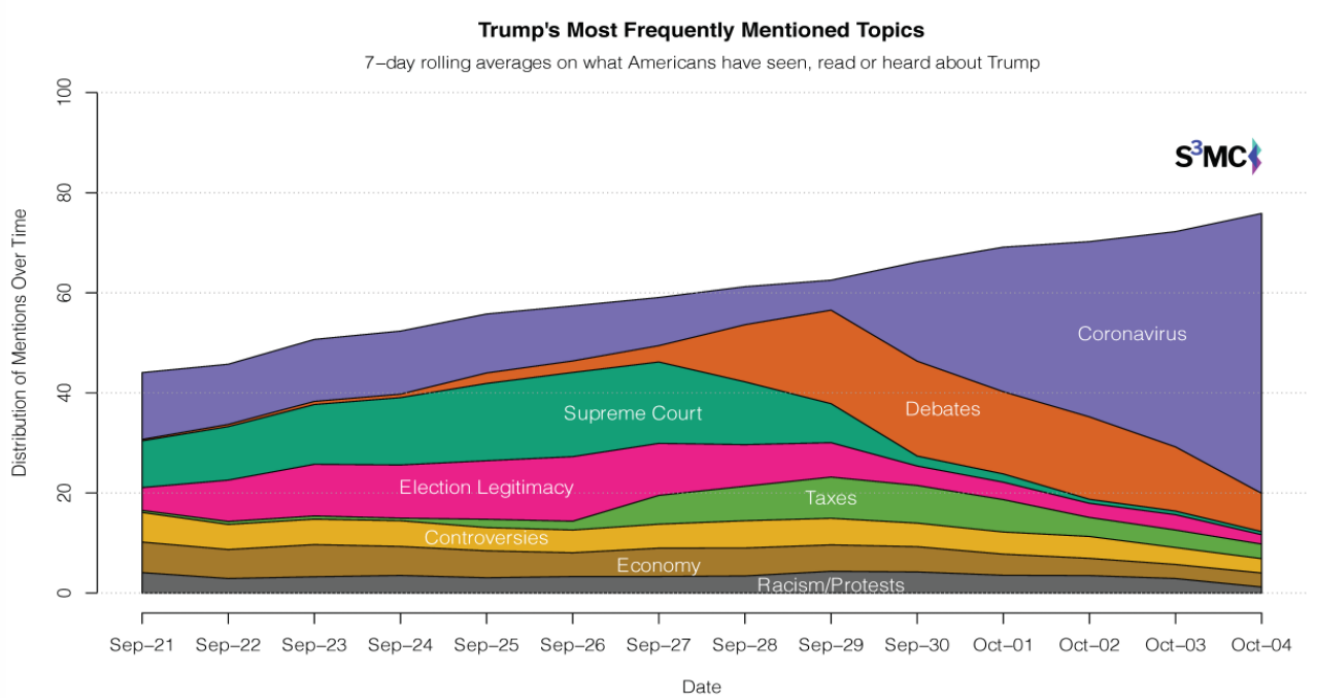


Ted Underwood, David Bamman, and Sabrina Lee (2018), "The Transformation of Gender in English-Language Fiction," (Cultural Analytics)

Presidential language → public attention

Input:
News articles

Output:
Topics recalled by public

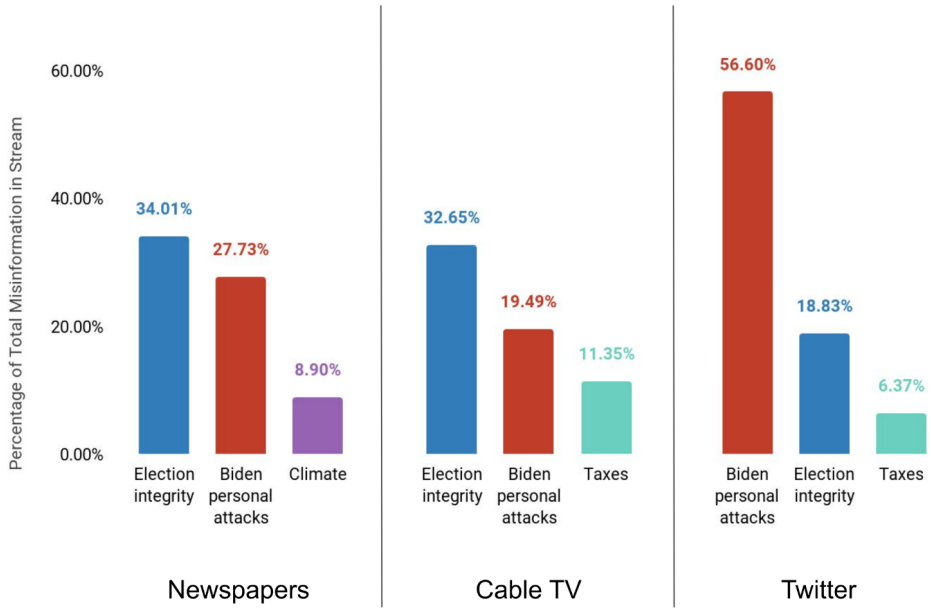


Jennifer Agiesta (2020), "In News about the Presidential Race, Coronavirus Overtakes Nearly All Else" (CNN Politics)

Digital media → misinformation over time

Input:
Articles,
tweets,
segments

Output:
Misinformation
themes pushed
by presidential
candidates

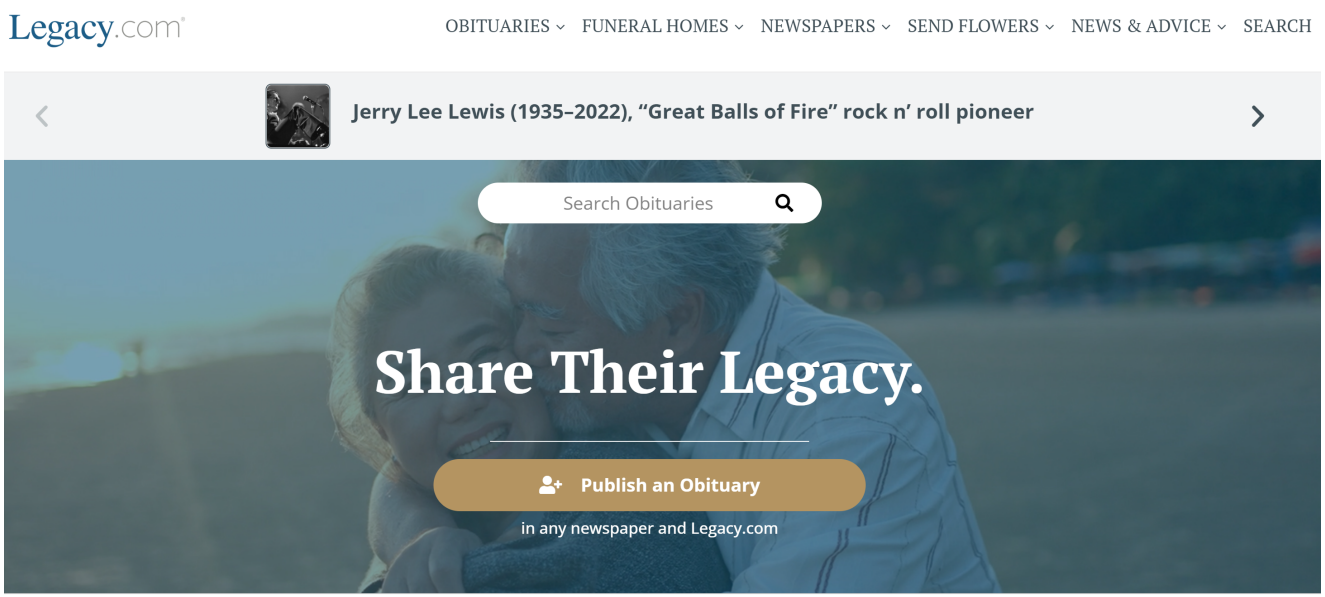


Jaren Haber, Lisa Singh, Ceren Budak, Josh Pasek, et al. (2021), Lies and presidential debates: How political misinformation spread across media streams during the 2020 election (HKS Misinfo. Review)

Obituaries → mortality by subgroups

Input:
Online
obituaries
from
Legacy.com

Output:
Models
of all-
cause
mortality;
predicted
trends



Where we are

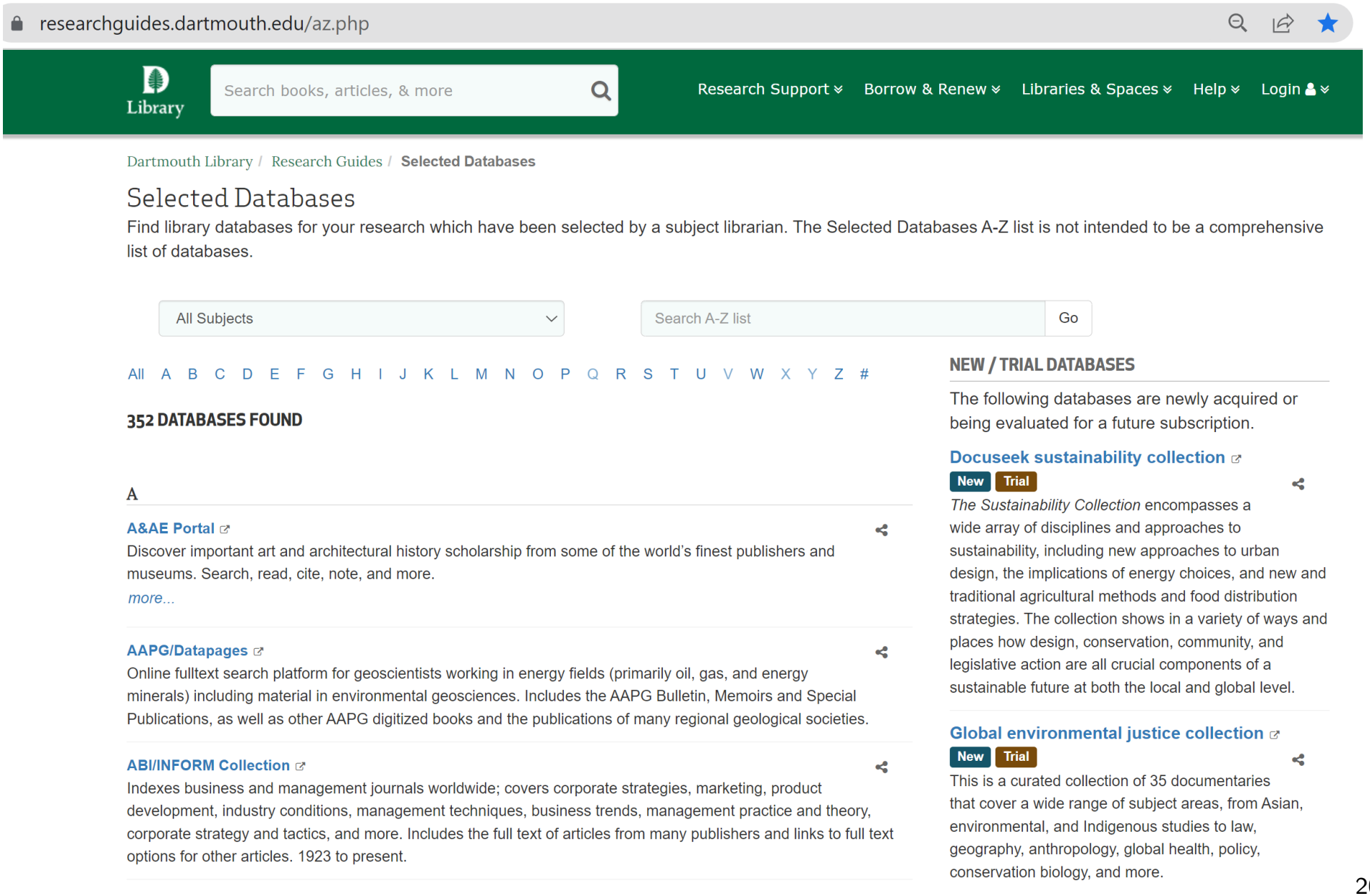
- Recap of supervised machine learning
- **Lecture on web-scraping**
  - Living in a digital era
  - Researching the digital era
  - **How to scrape/crawl without reinventing the flat tire**
  - Intro to packages for wrangling web data
  - Ethics (briefly)
- Notebook for basic web-scraping
  - Making `requests`
  - Parsing HTML
  - Scraping URLs

Web-scraping decision hierarchy

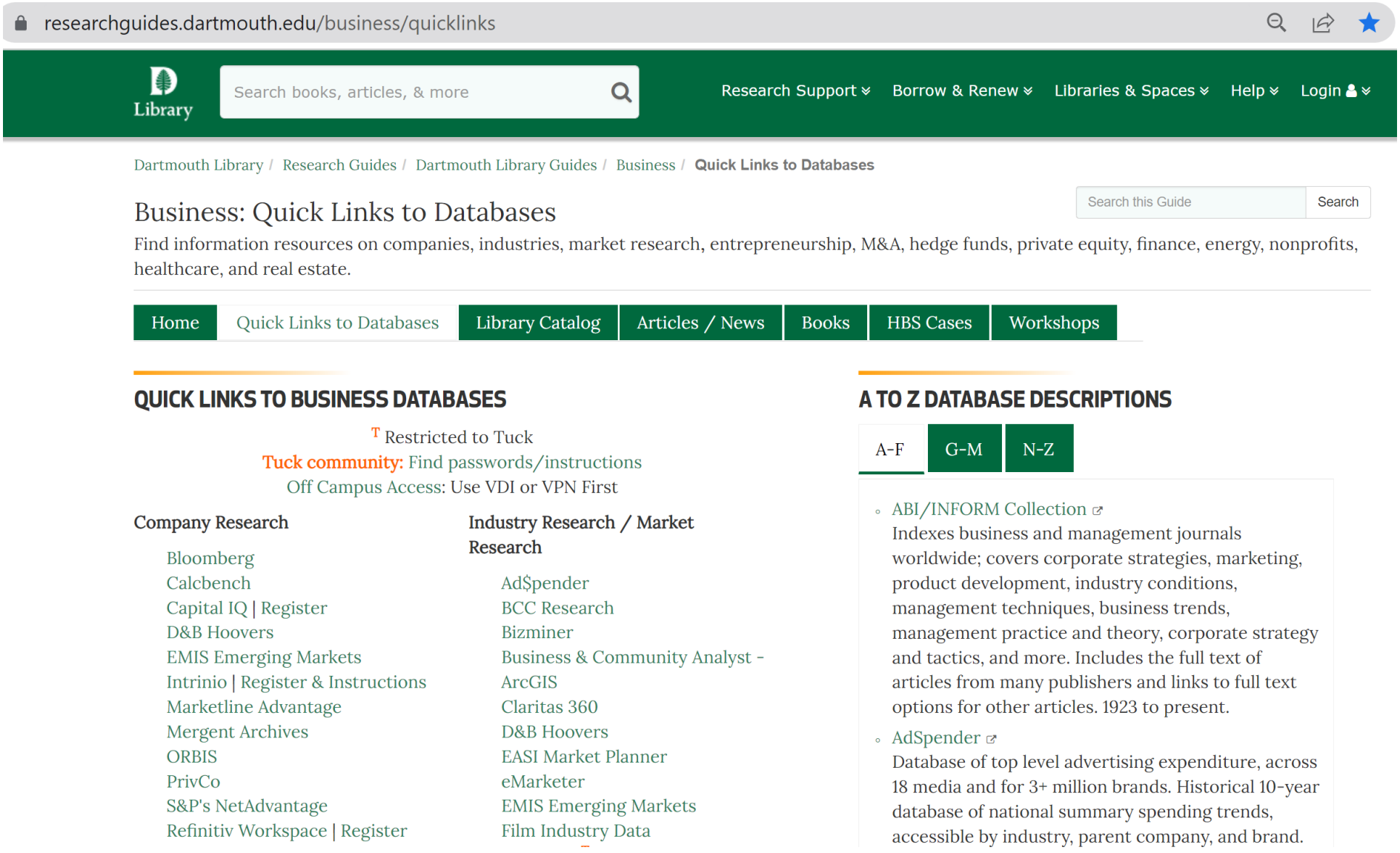
Existing data

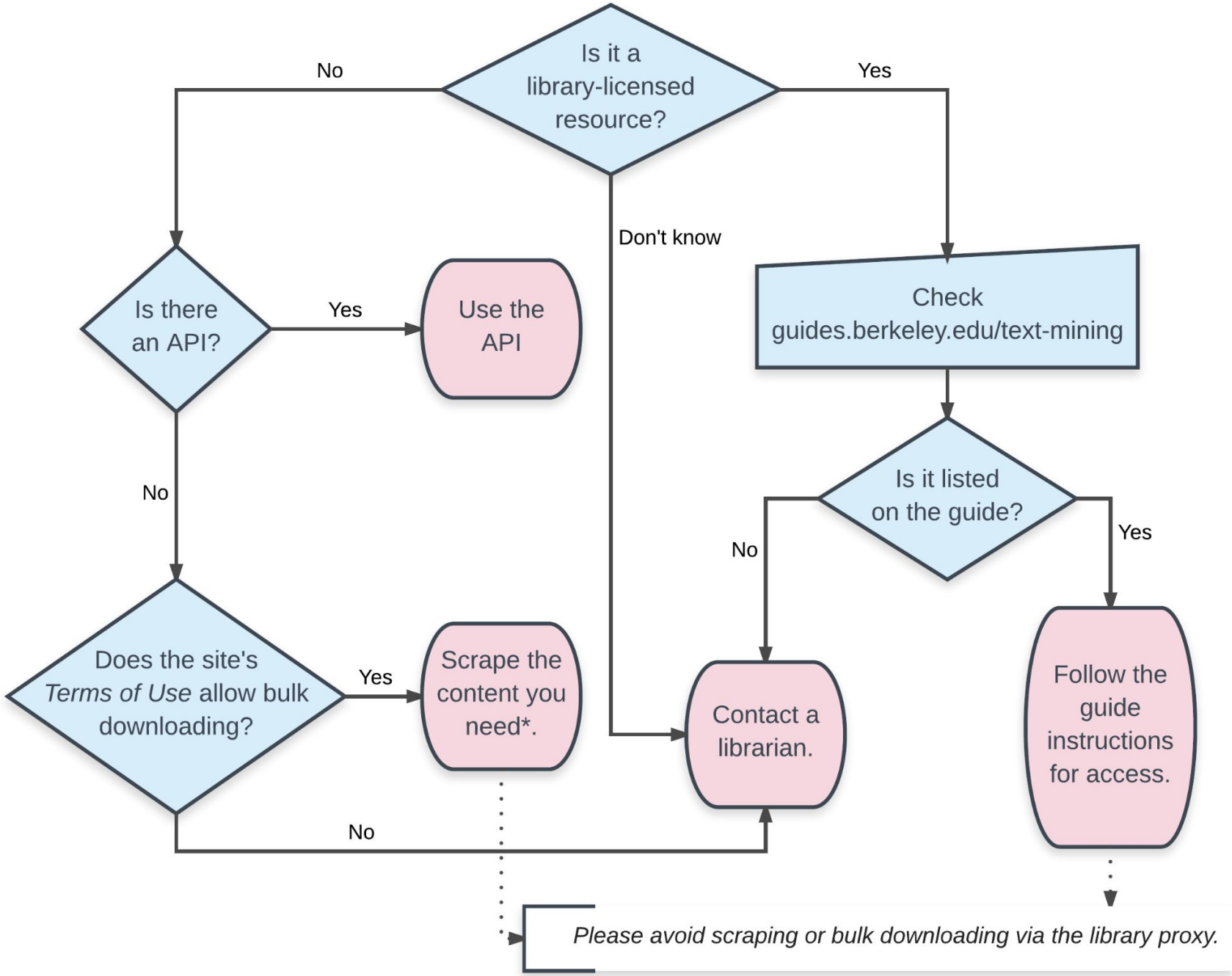
No scraping

Existing data sources: Dartmouth Library



Existing data sources: Business

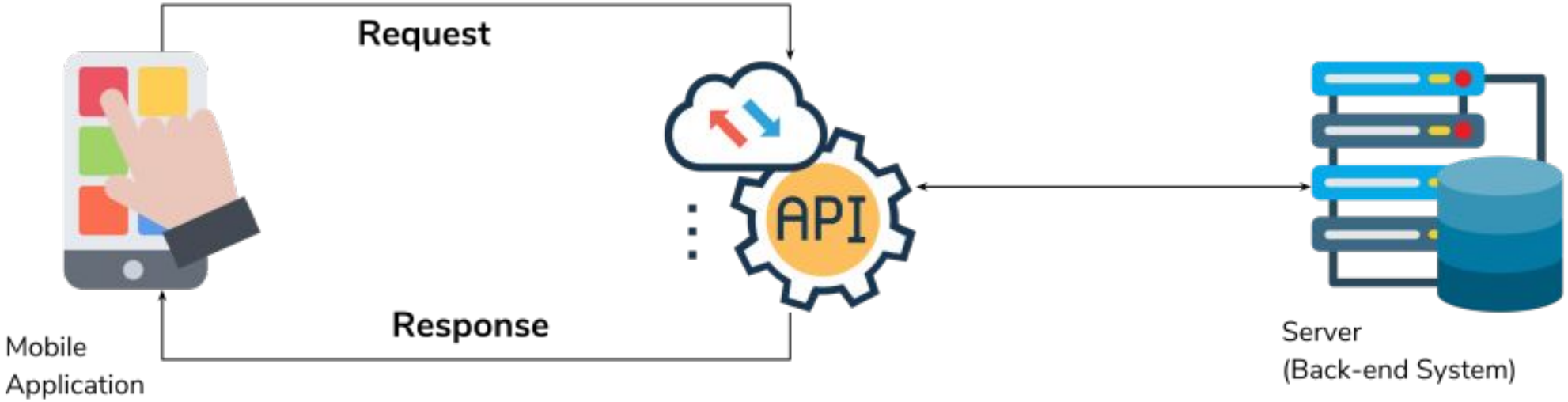


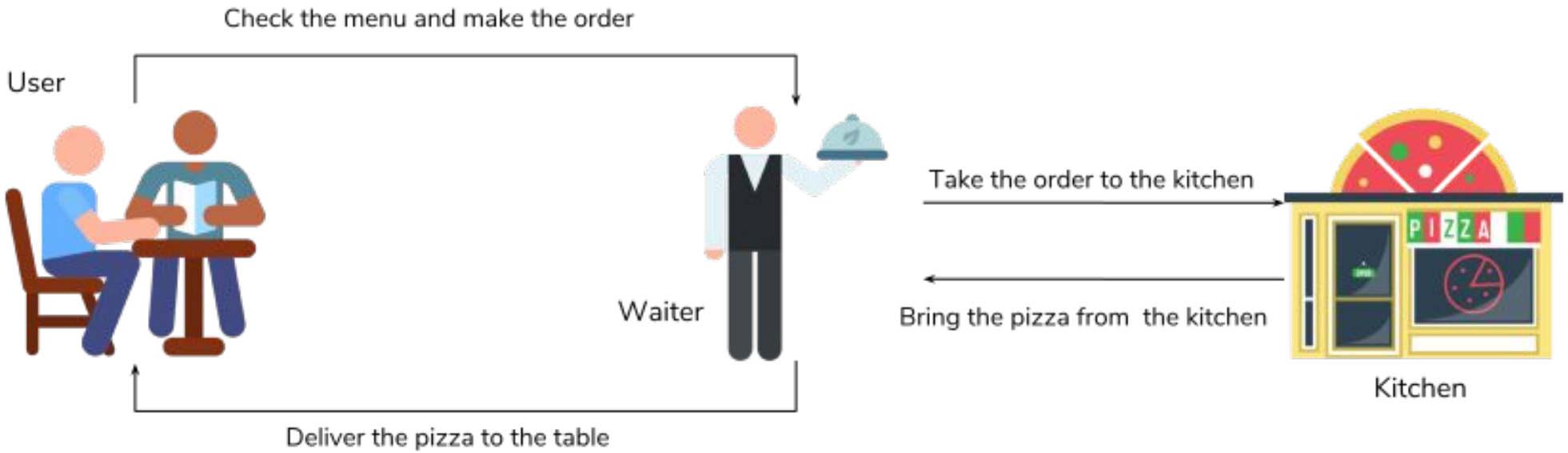

Is it a library-licensed resource?
No
Yes
Don't know
Is there an API?
Yes
Use the API
No
Does the site's *Terms of Use* allow bulk downloading?
Yes
Scrape the content you need*.
No
Contact a librarian.
Check guides.berkeley.edu/text-mining
Is it listed on the guide?
No
Yes
Follow the guide instructions for access.
Please avoid scraping or bulk downloading via the library proxy.

Web-scraping decision hierarchy

No scraping

Easy scraping





Amanda Kothawala (2018), *What is an API? How does it work?* (Medium) https://medium.com/@ama.thanu/what-is-an-api-how-does-it-work-f4ea552d741f

Web-scraping decision hierarchy

No scraping

Easy scraping

Consistent scraping

How the web works

Please send me the files to display this website (REQUEST)

Here you go, parse away (RESPONSE)

Web-scraping decision hierarchy

Existing data	Web APIs	Structured HTML	*Wild West*
No scraping	Easy scraping	Consistent scraping	• Directory • Scrape URLs • Scrape HTML • Parse HTML • Store output • Filter content

Parents & Families | Future Staff | Partners & Donors

Search | Apply | Enroll | Donate

KIPP HOUSTON PUBLIC SCHOOLS

Prime T.I.M.E.

ASPIRE PUBLIC SCHOOLS

DISCOVER ASPIRE | LOCATIONS | EN

← Central Valley Region

ASPIRE PORT CITY ACADEMY

- Intro
- School Address & Contact Information
- School Officials
- School Officials
- Social Networks
- Key Documents + Links
- Board Meetings
- Location

Enroll Now

Aspire Port

Welcome to the Aspire Pc

Grades: K – 5
Sponsoring District: Stock
CDS/Charter Numbers: CI
#1553

Enrollment: 410
Ethnicity:
Hispanic: 52.4%

Will Carleton Academy

About | Admissions | Academics | Athletics | Student Life | Parents | Calendar | Faculty | Contact

Will Carleton Academy is a K-12 tuition-free public charter school in Hillsdale, Michigan founded in 1998. Will Carleton Academy devotes itself to serving the community as a charter school where parents can choose a traditional curriculum and educational atmosphere that is committed to both the intellectual development and the character formation of the students.

Our Mission Is

To Educate
- Back-to-basics curriculum
  - K-8th-Core Knowledge Sequence
  - 9th-12th: Michigan Merit/College Preparatory Curriculum
- Small classroom sizes at

To Enrich
- Advanced Placement class offerings on demand
- Dual Enrollment, Early/Middle College, and Vocational-Technical Programs
- Extensive extracurricular offerings at all grade

To Enlighten
- Orderly, disciplined environment
- Focus on the whole child, with the goal of helping to mold outstanding citizens
- K-12th Art and Music curriculum
- 4th-12th French

Aspire Port City Academy

— ACCESS SCHOOLMINT

chedule a Tour

AIM College and Career Preparatory Academy
5200 Ave N 1/2, Galveston, TX, 77550

Select a School | Translate

Sign In Register

For Parents | Counseling | Clubs | For Staff | Calendar

New School Hours for the 2018 - 2019 School Year

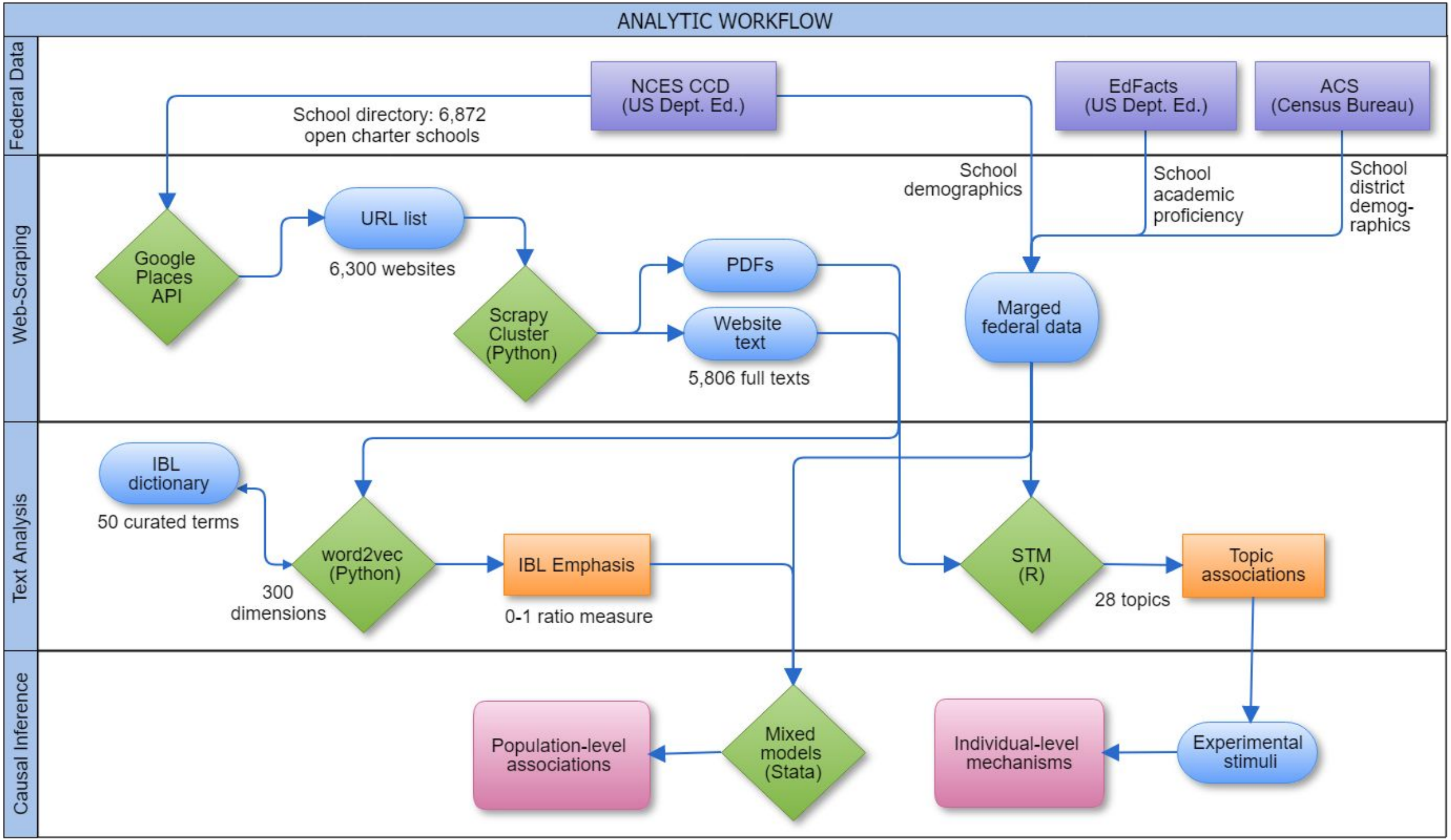
8:15 AM - 3:45 PM

Welcome to AIM College and Career Preparatory Academy

AIM is a GISD charter school that provides students the opportunity to accelerate their own learning. We are a 6th – 12th grade campus with a variety of programs. Middle school students have opportunities to test and promote to the next grade level. High school students attend face-to-face, hybridized and online courses. We also offer the HSEP (High School Equivalency Program) for qualifying students.

The mission of Ann Arbor Learning Community is to nurture independent learners as they acquire the tools they need to shape an environmentally and socially responsible future.

Enroll Your Student K-8


ANALYTIC WORKFLOW
Federal Data
Web-Scraping
Text Analysis
Causal Inference
NCES CCD
(US Dept. Ed.)
EdFacts
(US Dept. Ed.)
ACS
(Census Bureau)
School directory: 6,872
open charter schools
School
demographics
School
academic
proficiency
School
district
demog-
raphics
Google
Places
API
URL list
6,300 websites
Scrapy
Cluster
(Python)
PDFs
Website
text
5,806 full texts
Marged
federal data
IBL
dictionary
50 curated terms
word2vec
(Python)
300
dimensions
IBL Emphasis
0-1 ratio measure
STM
(R)
28 topics
Topic
associations
Mixed
models
(Stata)
Population-level
associations
Individual-level
mechanisms
Experimental
stimuli

Essential tools we'll use to wrangle web data

Requests

```
# Simple page HTML grabbing
import requests

url =
'https://www.politifact.com/factchecks/2021/apr/02/citizens-united/citizens-united-calls-bidens-infrastructure-plan-g/'
response = requests.get(url)
html = response.text
```

Beautiful Soup

```
# Anchor extraction from HTML document
from bs4 import BeautifulSoup
from urllib.request import urlopen
with
urlopen('https://en.wikipedia.org/wiki/Main_Page')
as response:
    soup = BeautifulSoup(response, 'html.parser')
    for anchor in soup.find_all('a'):
        print(anchor.get('href', '/'))
```

Parsing with an exclusion list

```
soup = BeautifulSoup(open(HTML_file), "lxml")
inline_tags = ["b", "big", "i", "small", "tt", "abbr", "acronym", "cite", "dfn",
"em", "kbd", "strong", "samp", "var", "bdo", "map", "object", "q", "span", "sub",
"sup"] # junk tags to eliminate

[s.extract() for s in soup(['style', 'script', 'head', 'title', 'meta',
'[document]'])] # Remove non-visible tags

for it in inline_tags:
    [s.extract() for s in soup("</" + it + ">")] # Remove inline tags
```

Scrapy

```python
# Simple quote scraper
import scrapy
class SimpleSpider(scrapy.Spider):
    name = "simple"
    start_urls = ['http://quotes.toscrape.com/page/1/']

    def parse(self, response):
        page = response.url.split("/")[-2]
        filename = 'quotes-%s.html' % page
        with open(filename, 'wb') as f:
            f.write(response.body)
```

Mission and Vision

Stony Point Academy is committed to provide a rigorous preparatory program that ensures all students are ready for successful post-secondary pathway. This includes an academically rich curriculum with rigorous content. It includes an expectation that students will apply their knowledge through higher-order skills, and will develop the habits of mind and character traits known to support personal standards within a student-centered environment that emphasizes the Common Core Anchor Standards for College and Career Readiness (CCR) across all content areas at every grade level. It is with this preparation that students will become contributing members of the local and global communities.

Stony Point Academy

Scrapy **output**

Home
Our School
...
Mission and Vision

Stony Point Academy is committed to provide a rigorous preparatory program that ensures all students are ready for successful post-secondary pathway. This includes an academically rich curriculum with rigorous content. It includes an expectation that students will apply their knowledge through higher-order skills, and will develop the habits of mind and character traits known to support personal standards within a student-centered environment that emphasizes the Common Core Anchor Standards for College and Career Readiness (CCR) across all content areas at every grade level. It is with this preparation that students will become contributing members of the local and global communities.
Stony Point Academy
Building Future Leaders
Job Opportunities
Contact us
Website Issues
...

Also useful: wget

```
wget {URL}
```

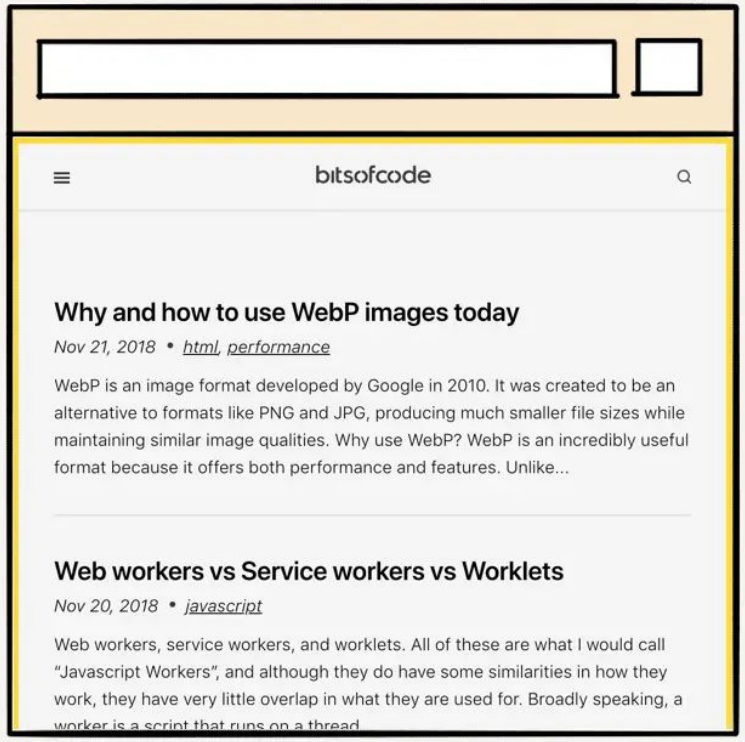
```
wget --no-parent --level 0 --no-check-certificate --recursive --adjust-extension
--convert-links --page-requisites --wait=10 --random-wait --execute robots=off
--follow-ftp --secure-protocol=auto --retry-connrefused --tries=12 --no-remove-listing
--local-encoding=UTF-8 --no-cookies --default-page=default --server-response
--trust-server-names --header="Accept:text/html" --user-agent=Mozilla
--warc-file=bridge_warc --warc-cdx --warc-max-size=0.5G {URL}
```

`--recursive`	Recursively download files and follow links
`--no-parent`	Follow links, but not beyond the last parent directory
`--page-requisites`	Grab all of the linked resources necessary to render the page (images, CSS, JS, etc.)
`--adjust-extension`	Append .html to the files when appropriate
`--convert-links`	Turn links into local links as appropriate
`--execute robots=off`	Turn off wget's automatic robots.txt checking
`--user-agent`	Mask the user agent and show wget like a browser

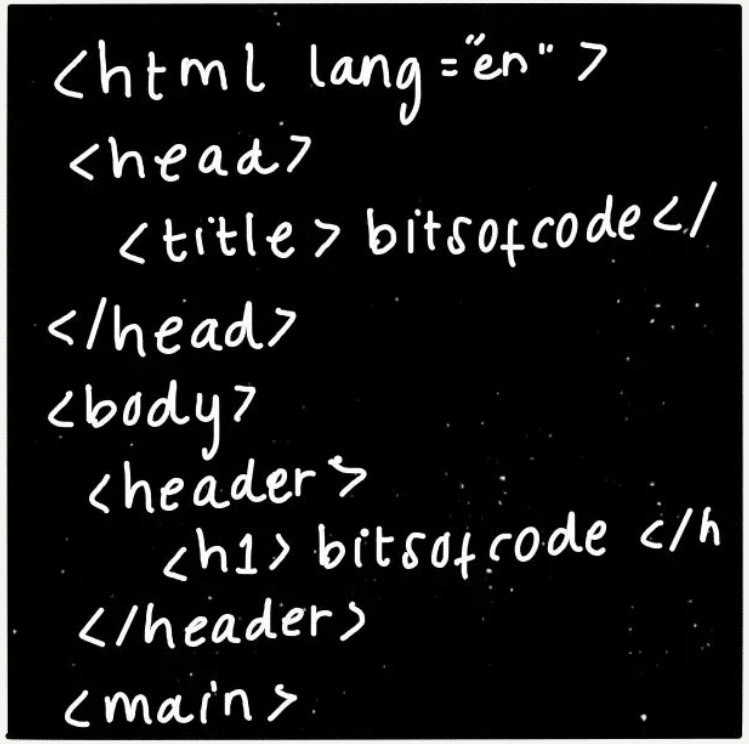
wget **output**

… xp=s\x92éëÊ00ÿ\r[ÖMission and VisionÈèüû]Dg¨\x98\x95\xad-\x01ê\x1bÿìö?YüoeF2æAv\x150TS
\x8bS\x82 ÄÝ¤\x98rY-#cÌ\x16µääâ\x8d8@¯L\x01û\x80sÍ\x89xp=s\x92éëÊ00ÿ\r[ÖÈèüû]Dg ¨
\x98\x95\xad-\x01ê\x1bÿìö?Yüo … x80sÍ\x89xp=s\x92éëÊ00ÿ\r[ÖBell ScheduleÈèüû]Dg ¨
\x98\x95\xad-\x01ê\x1bÿìö?YüoeF2æAv\x150TS\x8bS\ x82Ä Ý¤\x98rY-#cÌ\x16µääâ\x8d8@ ¯
L\x01û\x80sÍ\x89xp=s\x92éëÊ00ÿ\r[ÖÈèüû]Dg¨\x98\x95\xad-\x01ê\x1bÿìö … â\x8d8@ ¯
L\x01û\x80sÍ\x89xp=s\x92éëÊ00ÿ\r[ÖCollege & Career ReadinessÈèüû]Dg¨\x98\x95\xad-\x01ê\x1 …
x16µääâ\x8d8@¯L\x01û\x80sÍ\x89xp=s\x92éëÊ00ÿ\r[ÖStony Point Academy is committed to provide a rigorous preparatory program that ensures all students are ready for successful post-secondary pathway. This includes an academically rich curriculum with rigorous content. It includes an expectation that students will apply their knowledge through higher-order skills, and will develop the habits of mind and character traits known to support personal standards within a student-centered environment that emphasizes the Common Core Anchor Standards for College and Career Readiness (CCR) across all content areas at every grade level. It is with this preparation that students will become contributing members of the local and global communities.Èèüû]Dg¨\x98\x95\xad-\x01ê\x1bÿìö?YüoeF2æAv\x150TS\x8bS\x82ÄÝ¤\
x98rY-#cÌ\x16µääâ\x8d8 @ ¯L\x01û\x80sÍ\x89xp=s\x9 2éëÊ00ÿ\r[ÖÈèüû]Dg ¨
\x98\x95\xad-\x01ê\x1bÿìö?YüoeF2æAv\x150TS\x8bS\x82

Also useful: `Selenium`

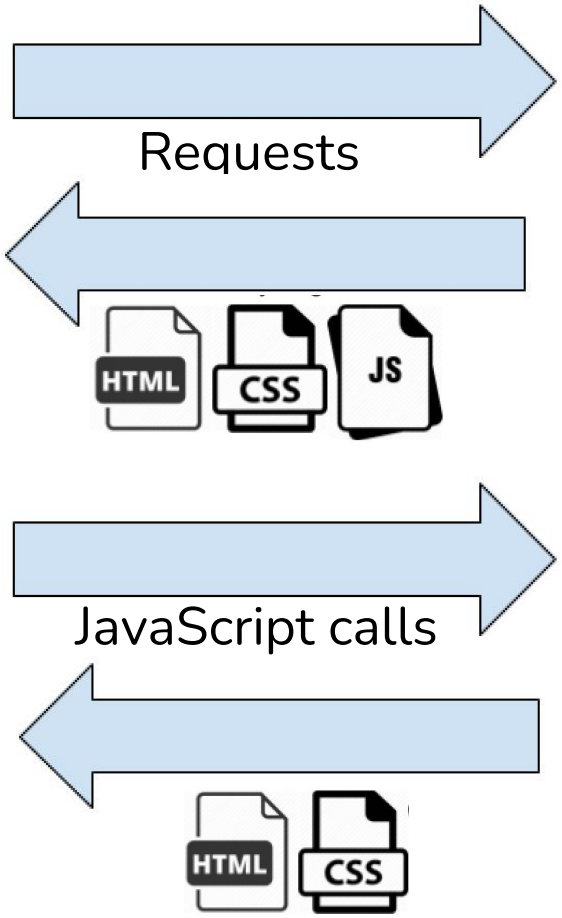


headful



headless

How interactive websites work



Limits of Selenium

```
[31mcrawler_10       |[0m 2018-04-05 21:42:10 [selenium.webdriver.remote.remote_connection]
DEBUG: GET http://172.18.0.2:4444/wd/hub/session/a9700948ceef7d04568452492f07e968/element/
0.786584017266313-80/attribute/href {"sessionId": "a9700948ceef7d04568452492f07e968", "name":
"href", "id": "0.786584017266313-80"}
[36;1mcrawler_2      |[0m TypeError: can't pickle thread.lock objects
[32mcrawler_5        |[0m 2018-04-05 21:41:37 [selenium.webdriver.remote.remote_connection]
DEBUG: GET http://172.18.0.2:4444/wd/hub/session/dcc13b350db90f2c33fccf3a6a15f850/element/
0.30704028792259 35-111/attribute/href {"sessionId": "dcc13b350db90f2c33fccf3a6a15f850",
"name": "href", "id": "0.30704028792259 35-111"}
[32;1mcrawler_9      |[0m 2018-04-05 21:42:29 [selenium.webdriver.remote.remote_connection]
DEBUG: Finished Request
[36mcrawler_6        |[0m {'appid': u'testapp',
[35;1mcrawler_1      |[0m 2018-04-05 21:42:44 [selenium.webdriver.remote.remote_connection]
DEBUG: GET http://172.18.0.2:4444/wd/hub/session/a0eaf0be3a4c7e187452951948b8a333/element/
0.1420010272851573-52/attribute/href {"sessionId": "a0eaf0be3a4c7e187452951948b8a333",
"name": "href", "id": "0.1420010272851573-52"}
```

Ethics of web-scraping:

Politeness good

Hacking bad

More on ethics

- How do I know what's OK to scrape?
  - Ask your IRB (they don't always know)
  - Follow the website's Terms of Service (usually)
- Am I legally liable if I break the Terms of Service?
  - *Computer Fraud and Abuse Act* (1986): Kindof
  - *LinkedIn v. HiQ Labs* (2019, 2022): Nope, web-scraping is a public right
- So is the whole web in the public domain?
  - It depends (*contextual privacy*; see *Bit by Bit*, 2017)

Take-aways

- Web-crawling new corpora (still) the Wild West
- But don’t hurt anyone or reinvent the flat tire
  - First check for existing data and APIs!
  - Scraping is OK, but be polite and share wisely
- Cornerstone web-crawling tool (for Python): `scrapy`
  - Flexible! And depends on a parser like BeautifulSoup
  - Extensible! And still time-consuming to customize
- URL scraper, Scrapy spiders, etc.:
  - https://github.com/jhaber-zz/web_scraping

Where we are

- Recap of supervised machine learning
- Lecture on web-scraping
  - Living in a digital era
  - Researching the digital era
  - How to scrape/crawl without reinventing the flat tire
  - Intro to packages for wrangling web data
  - Ethics (briefly)
- **Notebook for basic web-scraping**
  - Making `requests`
  - Parsing HTML
  - Scraping URLs

Notebook for basic web-scraping

https://github.com/jhaber-zz/QSS20_public/blob/main/activities/10_partI_intro_scraping.ipynb

- Making `requests`
- Parsing HTML
- Scraping URLs

QSS20: Modern Statistical Computing

Unit 12: Web-scraping, Part II (Scrapy)

Goals for today

- Upcoming deadlines
- Your feedback as of pset 3
- Recap of web-scraping so far
- Lecture on kinds of web-crawling (brief)
- Scrapy activity: notebook & command line

Goals for today

- **Upcoming deadlines**
- Your feedback as of pset 3
- Recap of web-scraping so far
- Lecture on kinds of web-crawling (brief)
- Scrapy activity: notebook & command line

Upcoming deadlines

- Problem set four
  - Course schedule: Due this Friday 11-04
  - Canvas Assignment: Due this Sunday, 11-06
  - **Which do you prefer?**
- Final project milestone 2
  - **Due Sunday, 11-06**
- Problem set five
  - **Due Friday, 11-18**
- Final project presentation
  - **Delivered in class next Wednesday, 11-14**
- Final paper
  - **Due Tuesday, 11-22**
- DataCamp modules
  - Full credit if you complete them all by **Tuesday, 11-22** (same as paper)
  - If doing DataCamp (default), will be graded for ALL assigned modules (deadlines other than 11-23)
  - If not doing DataCamp, **must** notify Prof. & TA

(All these deadlines are at 11:59 PM on the date indicated)

Where we are

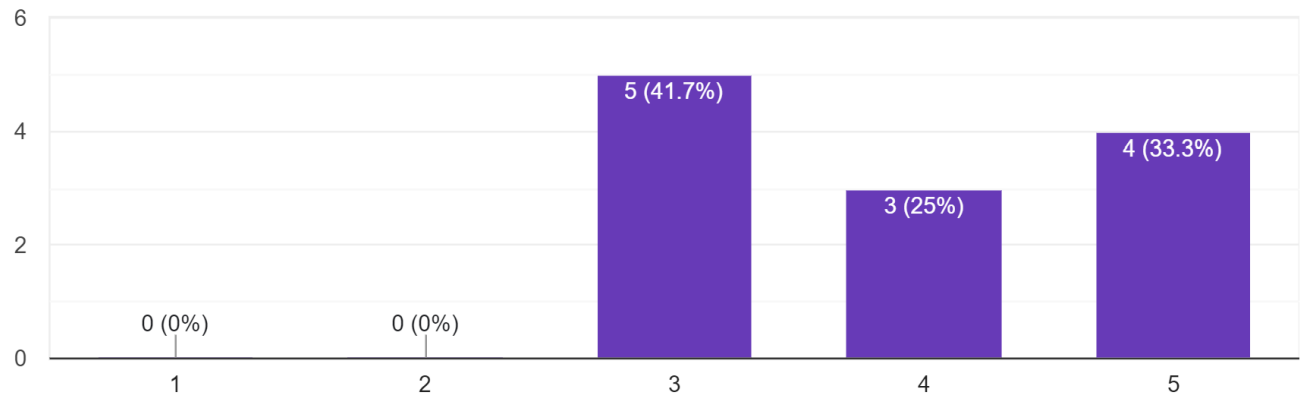
- Upcoming deadlines
- **Your feedback as of pset 3**
- Recap of web-scraping so far
- Lecture on kinds of web-crawling (brief)
- Scrapy activity: notebook & command line

Your feedback as of pset 3: Difficulty

How would you rate the difficulty of problem set 3?

Copy

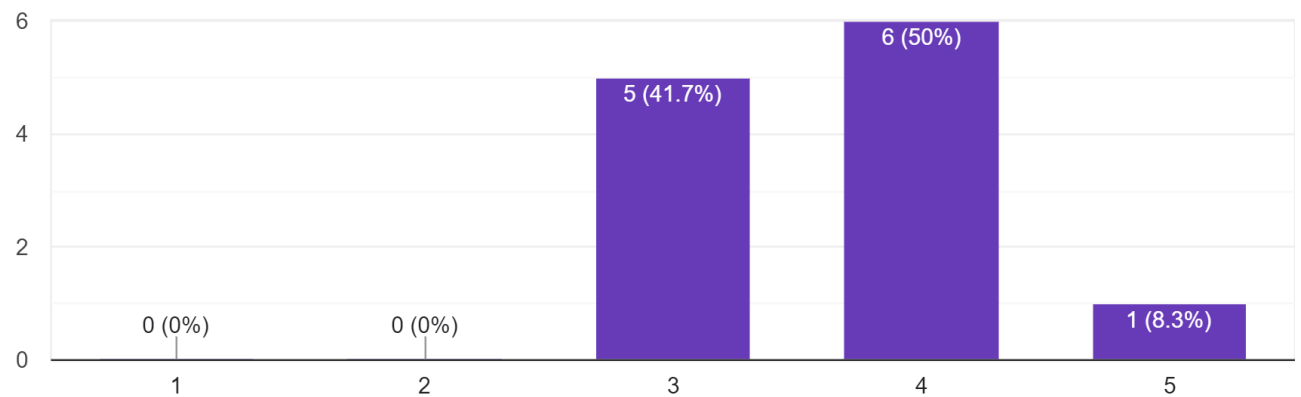
12 responses



How would you rate the overall pace of the course?

Copy

12 responses

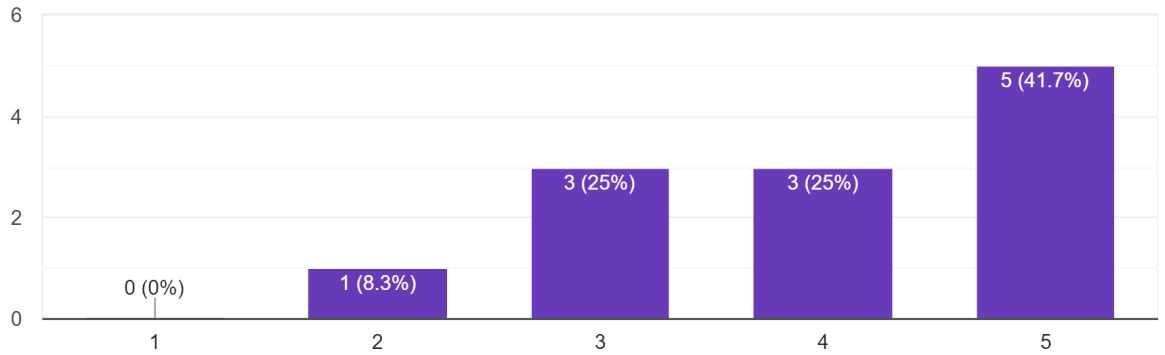


Your feedback as of pset 3: Partners

How much do you agree with this statement:
"Collaborating with a classmate resulted in a higher quality submission of pset 3."

Copy

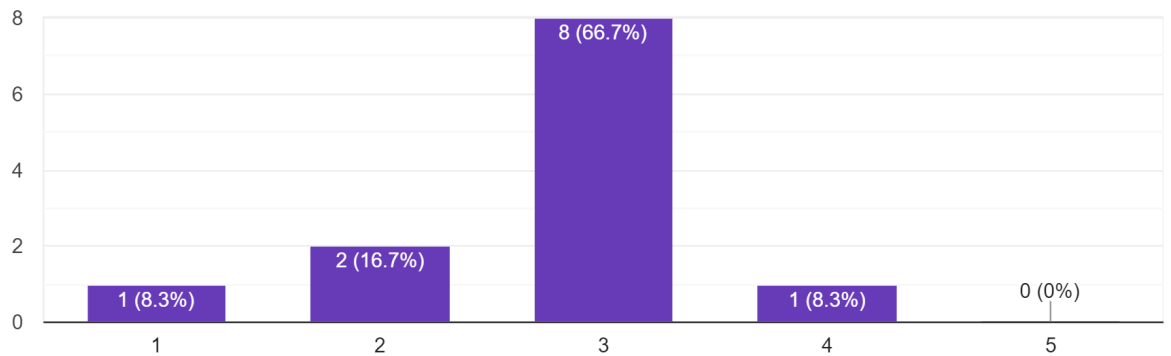
12 responses



Between you and your classmate/partner, who did more work on your submission of pset 3?
(If it was balanced, choose the middle button, 3.)

Copy

12 responses



Your feedback as of pset 3

Quant. summary:

- 7/12 respondents chose advanced scraping for extra class
- 10/12 say they're learning a lot!
- 5/12 doubt they have the background needed for course

Text comments:

- People like in-class activities (5/6 mentioned), partner psets, recaps
- One dislikes: long lectures, content too advanced, warmups with touch
- Two say pset 3 instructions not clear enough
- One says DataCamp after lecture (as practice) works better

Where we are

- Upcoming deadlines
- Your feedback as of pset 3
- **Recap of web-scraping so far**
- Lecture on kinds of web-crawling (brief)
- Scrapy activity: notebook & command line

Recap of web-scraping so far

What do you remember?

Recap of web-scraping so far

Tips:

- Scraping is hard, so check first for existing data or an API
- HTML exists in **tags** like <a> for links and <p> for paragraphs
- Custom scraping uses HTML structure: right-click → `Inspect` in browser
- Web-scraping can take one of two basic approaches:
  1. extracting specific content from some pages with a custom, fragile scraper: most common and time-consuming because HTML changes
  2. extracting everything from those pages with a more resilient but messier scraper—filtering may take time; some methods OK with mess)
- Is scraping 'ethical'? **Usually**—assuming you're polite: scrape slowly, respect privacy, don't share the data

Useful commands:

```
response = requests.get(url) # get content
soup = BeautifulSoup(response.text) # read content
exclusions = ['meta', 'script', etc.] # define junk elems to avoid
tag.extract() # take tag out, i.e. append to list
tag.decompose() # obliterate junk tag
[tag.decompose() for tag in soup(exclusions)] # iterate over exclusions
```

Example from last class: Scraping fact check description

Check out this url:
https://www.politifact.com/factchecks/2021/apr/02/citizens-united/citizens-united-calls-bidens-infrastructure-plan-g/

Challenge:
Use your browser to inspect the HTML and identify any unique classes that enclose the explanation (and nothing else).

Example from last class: Scraping fact check description

```
# Solution:

# Set URL to scrape
url = 'https://www.politifact.com/factchecks/2021/apr/02/citizens-united/citizens-united-calls-bidens-infrastructure-plan-g/'

# Scrape HTML with requests and beautifulsoup
response = requests.get(url)
soup = BeautifulSoup(response.text)

explanation = soup.find('article', class_='m-textblock').get_text() # identify this class from looking at HTML

import re
explanation = re.sub(r"\n+", "\n", explanation)
```

Where we are

- Upcoming deadlines
- Your feedback as of pset 3
- Recap of web-scraping so far
- **Lecture on kinds of web-crawling (brief)**
- Scrapy activity: notebook & command line

Why web-crawling?

- *Web-scraping*: programmatically going over web pages to extract data
- Custom scraping depends on knowing where the information you want is located: URLs and HTML tags (and CSS selectors)
- Don't have this? Then do URL searching (see previous notebook) and *web-crawling*: following links around the internet to locate content to scrape
- Rather than building your own crawler using low-level libraries like `requests`, we'll use an existing module built for ease of use: `Scrapy`
- `Scrapy` is popular, flexible, and fast, and handles common features like concurrency, duplicate requests, following links, etc.
- Our test website, quotes.toscrape.com, lists quotes by famous authors and doesn't mind lots of people scraping it (else blocking might be an issue)

Goal for today: Build a functional web-scraper that walks through pages and extracts data from each, as a foundation for future scraping projects

Further references on `Scrapy`: scrapy basics, scrapy architecture, & scrapy FAQ

Types of web-crawling

- `Scrapy` can do *narrow crawling*, focusing on a few specific domains to repeatedly exploit their HTML/CSS structures; or *broad crawling*, tackling a range of websites (e.g., all schools) with a flexible, much less targeted algorithm
- More on the trade-off between narrow and broad crawling:
  - *Narrow crawling* maximizes precision—grabbing just the stuff you want, none of what you don't want (same idea as in ML)—at the expense of *extensibility*: the ability to incorporate new domains and be resilient to changes in website structure
  - *Broad crawling* maximizes extensibility, promoting flexibility and resilience through techniques like exclusion lists (as we worked with last class) and *link extraction*: finding all within-domain links on a given webpage, then from its children links, and so on
  - Messier output from broad crawling can make filtering and analysis challenging ("garbage in, garbage out"), but some methods less sensitive to noise (menus, footers, etc.), such as raw word counts or word embeddings

Terminology for web-scraping/crawling

- **Web-scraping (i.e., 'screen-scraping')**: Extracting structured information from the files that make up websites (i.e., what's shown in web browsers), relying on their HTML, CSS, and sometimes JS files
- **Web-crawling:** Finding web pages through links, automated search, etc. Once discovered, pages can be scraped
- **Narrow crawling (less extensible)**: Scraping a limited set of pre-defined domains: studying their HTML and CSS structures and exploiting these to extract specific information repeatedly. This maximizes precision in scraping while sacrificing extensibility. What people usually mean when they say 'web-scraping'
- **Broad crawling (more extensible)**: Collecting information on a range of websites and promoting flexibility in its scraping algorithm (way of extracting website information) at the expense of generally less clean output. It can be extended to a wide range of websites identified by, e.g., automated google search (what do people click on most?), network analysis (what websites tend to link to one another?), or link extraction
- **Extensibility**: Ability for a scraping approach to incorporate new domains or be resilient to changes in website structure. Generally higher for broad crawls than narrow crawls, at the expense of precision
- **Link extraction**: Finding all within-domain links on a given webpage, then all within-domain links on its children links, and so on to a specified depth
- **Horizontal crawling**: Crawling on the same hierarchical level as the input domain, such as going from the first to the second page of google results
- **Vertical crawling**: Crawling at a higher or lower level from the input domain, such as navigating to the 'About Us' page directly linked from a home page

Where we are

- Upcoming deadlines
- Your feedback as of pset 3
- Recap of web-scraping so far
- Lecture on kinds of web-crawling (brief)
- **Scrapy activity: notebook & command line**

Scrapy activity: Notebook & command-line

Notebook: https://github.com/jhaber-zz/QSS20_public/blob/main/activities/10_partII_crawling_scrapy.ipynb

- This notebook is a coding guide, not a coding tool (as we're used to)
- Scrapy is command-line-based, so we'll be using shell and text editor to make & run spiders