

# CS 398 ACC

## Data Sourcing / Cleaning

---

Prof. Robert J. Brunner

Ben Congdon  
Tyler Kim

# MP6

How's it going?

- Due March 13th at 11:55 pm.

**Submit your results as a PDF report on Moodle**

# Final Project Reminders

- **Project Proposal:** Due this **Friday, March 16th at 11:59pm**
  - See requirements on the Course Website.
  - **Submission through Moodle**
  - Only one group member needs to submit
    - Make sure you list all group members and group name in the proposal

## This Week

- Data Licensing / Sourcing
- Cleaning Data

# Data Licensing

- Can you use any data?
- Check data sources for restrictions (commercial uses, foreign uses, etc)
- MIT License
  - Very permissive, as long as you keep the license and copyright
- GPLv2/v3
  - If you use it, you must distribute the source of anything built with it
  - Must include copyright, license, link to the original, and details of your changes
  - “Spreads virally”, anything that uses it must then become GPL

# Data Sources

- Data is everywhere! We can track things at scales that we never have before
- Refined Datasets
  - These are typically academic or governmental datasets that are released to the public
  - For the most part all the cases of missing values and parsing is done for you (the data is in a table-like format)
- Raw Data
  - This can be any data source:
    - Social media, sensor data, scientific data
  - You will need to clean the data in order to get it to a usable state
    - i.e. Missing values? Formatting? Non-normalized data?

# Premade Dataset

- There are a lot of good data sources already
- If you can, use them instead of getting your own data
  - It has been cleaned; typically easier to download
  - There are other papers you can look to for examples of how they manipulated it
- You may want to combine datasets or fill in null values



<https://github.com/cazala/mnist>

# Raw Data

How would you process your newsfeed?





# Raw Data - Web Scraping

- Ideally you start with a headless browser and download a subset of the javascript objects/html from the page
- Once you have the objects, save them in some format that makes sense. This can be a CSV, a Table, what have you.
- You may need to do some HTML parsing in this case. Get yourself an HTML parser and write all the relevant files

## 20 Second Example

```
from bs4 import BeautifulSoup
soup = BeautifulSoup("""
    <html><body><ul>
    <li class="shoe-item">Air Jordans</li>
    <li class="shoe-item">Light up Sketchers</li>
    </ul></body></html>
    """, 'html.parser')
for item in soup.find_all(attrs='shoe-item'):
    print(item.text)
```

# Scraping Problems

- You may get rate limited
- You may get blocked
- You may be violating the law
- Whenever the HTML changes, your code immediately breaks



# Data Cleaning

# Why do we need to clean the data?

- Data could have:
  - Missing Values
  - Duplicate Values
  - Invalid Values
  - Useless Values
  - Etc
- We need to make sure that the data that we give to the machine learning algorithm is as close to representative as possible

# Why do we need to clean the data?

- We usually want the data in some normalized format

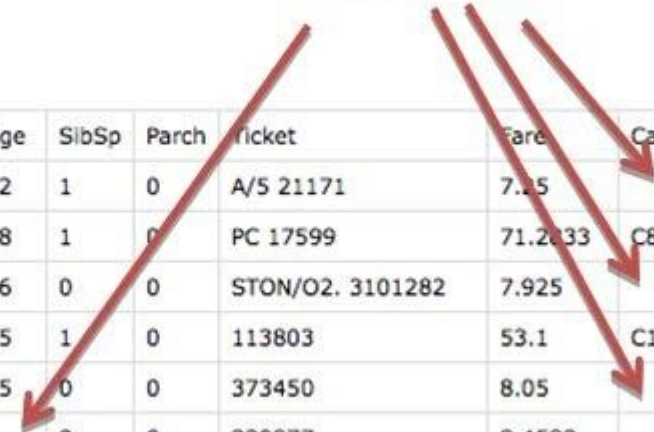
Results		Messages					
	EMP_ID	SSN	TITLE	FIRSTNAME	MIDDLEINIT	LASTNAME	EMAIL
1	5001	395031199	NULL	Caleb	NULL	Avila	NULL
2	5002	793333409	NULL	Shari	NULL	Webb	NULL
3	5003	357007477	Mrs.	Helen	NULL	Reeves	lbcq.lportbxw@allihx.com
4	5004	519506770	NULL	Yesenia	X	Moyer	NULL
5	5005	244993976	Miss.	Kathleen	NULL	Herrera	yaro.isylhgw@tsvjg.hxuhnu.net
6	5006	668369530	Mr	Tera	NULL	Kane	NULL
7	5007	229756457	NULL	Wayne	NULL	Duke	NULL
8	5008	019655316	NULL	Telly	NULL	Zavala	NULL
9	5009	436312171	Mr	Wallace	NULL	Glover	NULL
10	5010	925006654	NULL	Catherine	NULL	Johnston	NULL

# Missing Values

- What can we do?
- We can drop data
  - Careful: this may skew our dataset, especially if we have a lot of missing values.
- We can make an educated guess of the values
  - Hard to do for categorical data
  - For numerical data:
    - Simple replacement with the mean
    - Sample a probability distribution to fill in the values (randomly)
- Some algorithms don't need all the values filled in
  - In that case, we leave as is because we want to put as little of our bias into the data

# Missing Data Example

Missing values



The diagram illustrates missing values in a dataset. Red arrows point from the text 'Missing values' to specific cells in the table: the 'Age' cell for PassengerId 5, the 'Ticket' cell for PassengerId 2, the 'Fare' cell for PassengerId 2, and the 'Cabin' cell for PassengerId 5.

PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	male	22	1	0	A/5 21171	7.25		S
2	1	1	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	female	26	0	0	STON/O2. 3101282	7.925		S
4	1	1	female	35	1	0	113803	53.1	C123	S
5	0	3	male	35	0	0	373450	8.05		S
6	0	3	male		0	0	330877	8.4583		Q



# Duplicated Data

- Simple answer:
  - Deduplicate the data
  - Can be challenging with very large datasets
- Pitfalls:
  - Duplicate data can have meaning
  - How do you determine duplicate data?
    - Exact match? Close match?

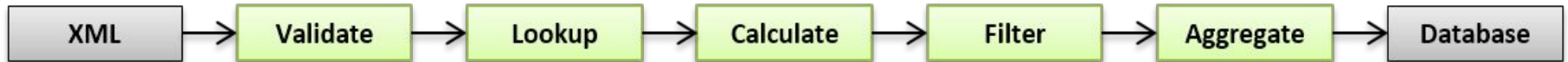
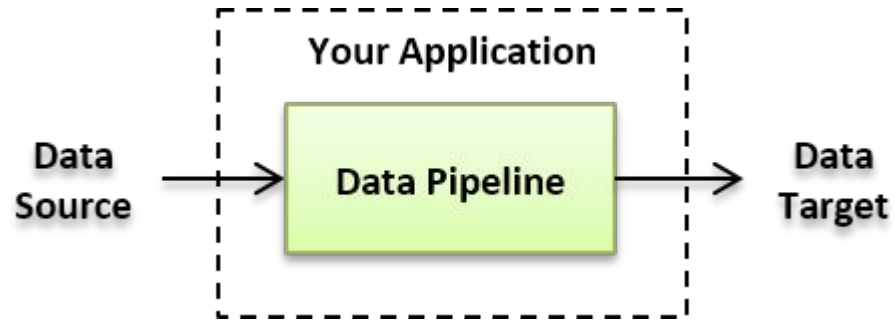
# Duplicates Example

	UserName	Location	Salary
	Suresh	KL	6000
	Dasari	Hyderabad	4000
	Prasanthi	Chennai	17000
	Nagaraju	Hyderabad	40000
	SureshDasari	Chennai	20000
	SureshDasari	Chennai	20000
	SureshDasari	Chennai	20000
	Mahesh	Vijayawada	10000
	Madav	Nagpur	15000
►*	NULL	NULL	NULL

# Invalid and Useless Values

- Problem domain specifies what is valid / useful
- Context matters significantly:
  - i.e., age > 0
- For “useless” values
  - Keep them around, they may be useful later

# Data processing pipelines



# Final Destination

- Your data needs to be accessible to your processing framework
  - i.e. It needs to be placed in HDFS, S3, MySQL, etc.
- For Hadoop / Spark:
  - HDFS has the “copyFromLocal” tool
- For SQL Databases:
  - Many tools available for loading data
  - Write your own queries/scripts to load data from some other source
- Data can be very large (> 1 PB):
  - Services like AWS Data Transfer (Snowball) exist to bulk transfer large amounts of data

# Last Thing - (De)anonymization

- Personally Identifiable Information:
  - Usually good practice to censor / anonymize PII
  - I.e. For a patient/disease dataset:
    - Instead of using <name, disease>, use <name\_id, disease\_id>
    - Name\_id can be a random UUID chosen for the given patient
- Malicious actors have incentive to deanonymize your data
  - Use cryptographically secure methods when anonymizing data

# Wednesday

- Project Proposal Help