Lecture 4: The Data Lecture

Prof. Robert Brunner

Quinn Jarrell
Sameet Sapra

Getting Data

- A lot of free, large, published data is out there already
  - Easy to get
  - Trusted
  - Densely populated (for the most part)
- But sometimes you need to fetch data yourselves
  - Now you need to be more cautious
  - What data is good?
  - Where are you getting the data from? Is it a trusted source?

Premade Data

- There are a lot of good data sources already
- If you can, use them instead of getting your own data
  - It has been cleaned
  - It is typically easier to download
  - There are other papers you can look to for examples of how they manipulated it

Data Licensing

- Can you use any data?
- Check data sources for restrictions (commercial uses, foreign uses, etc)
- MIT License
  - You can do anything with it as long as you keep the license and copyright
  - One of the most easy to use licensing
- 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
  - Sorta like a virus, anything that uses it must then become GPL
    - Actually a good thing when you want to keep software/data completely open
    - But companies typically hate it for obvious reasons

Decoding Data

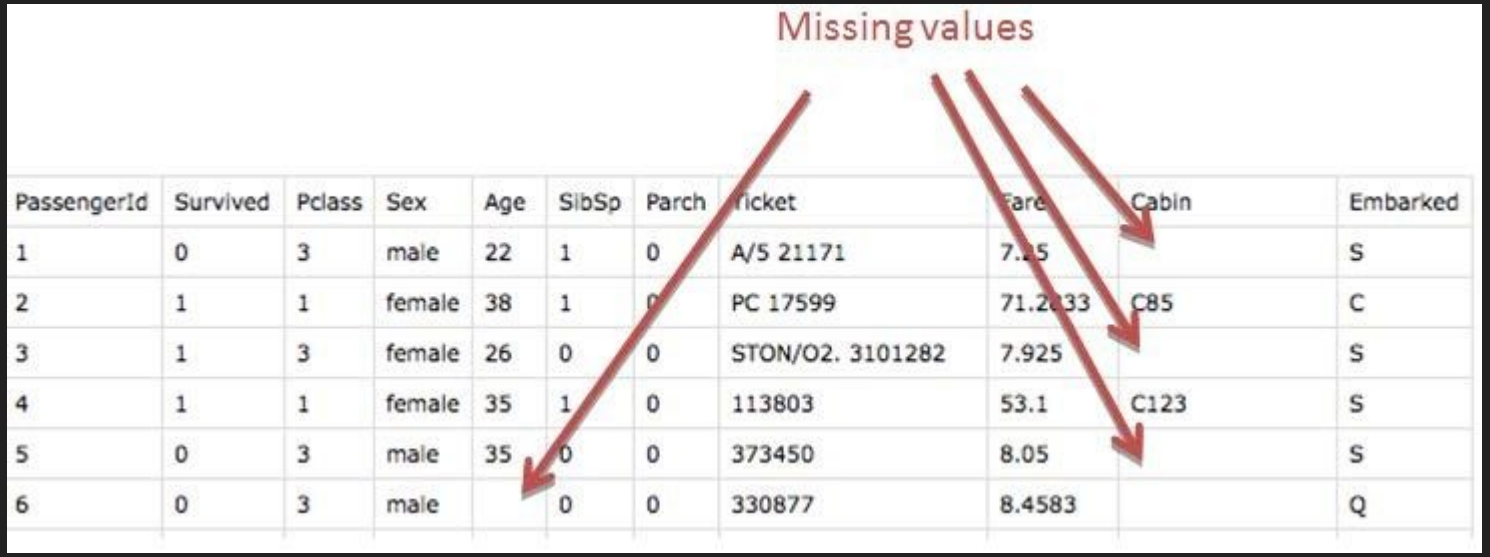
- You are used to text data in either ASCII or UTF-8 format
  - Normal data that your text editor can read
- There is a lot of rarer data formats
  - Excel files, pdfs, zip, tar, … and a billion more
- Two main different types of encoding
  - Encoding on top of ASCII/UTF-8 like CSV, JSON, XML
  - Binary encodings like excel, jpg, anything that’s not text
    - Text ASCII/UTF-8 are binary encodings themselves

Dirty Data

- So you got your hands on this data set
- But some of the values are missing, corrupted, wrong, duplicated
- You get a better answer by improving the quality of the values in your dataset
- So let’s look at some examples of bad data

Missing Data Example

-



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

Missing Data Example

- Either drop the rows with empty data or fill in default values
  - Strategy depends on whether or not you can safely define a default value
- If you drop too much data, it can skew the data
  - Leads to biased results
- Watch out for missing data
  - Sometimes it is not a coincidence, it’s a pattern

Many values mean the same thing

-

Filter

	country	iso2	iso3	year	new	sex	age	ep	rel	sn	sp
1	Afghanistan	AF	AFG	1980	new	m	014	NA	NA	NA	NA
2	Afghanistan	AF	AFG	1981	new	m	014	NA	NA	NA	NA
3	Afghanistan	AF	AFG	1982	new	m	014	NA	NA	NA	NA
4	Afghanistan	AF	AFG	1983	new	m	014	NA	NA	NA	NA
5	Afghanistan	AF	AFG	1984	new	m	014	NA	NA	NA	NA
6	Afghanistan	AF	AFG	1985	new	m	014	NA	NA	NA	NA
7	Afghanistan	AF	AFG	1986	new	m	014	NA	NA	NA	NA
8	Afghanistan	AF	AFG	1987	new	m	014	NA	NA	NA	NA
9	Afghanistan	AF	AFG	1988	new	m	014	NA	NA	NA	NA
10	Afghanistan	AF	AFG	1989	new	m	014	NA	NA	NA	NA

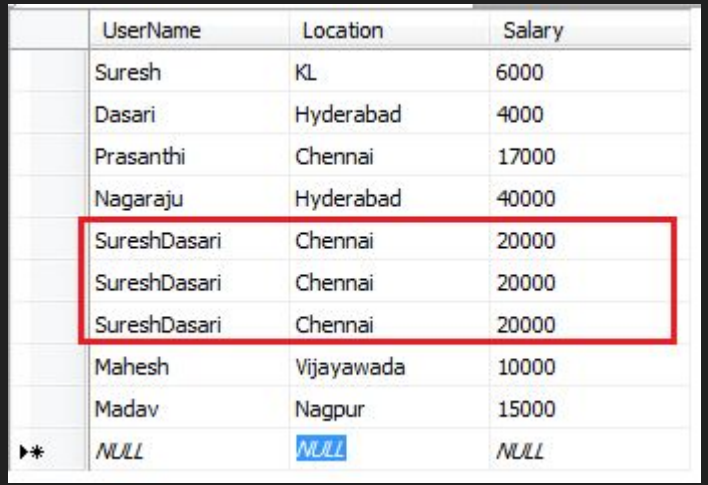
Showing 1 to 10 of 101,360 entries

Redundant Values Strategy

- Just pick the columns of data that you need
- Be consistent with what columns you pick so your data is uniform

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*

Duplicate Examples Strategy

- Check if you have processed a row before you process it again
- “Reducing” can help

Evaluating Quality of Data

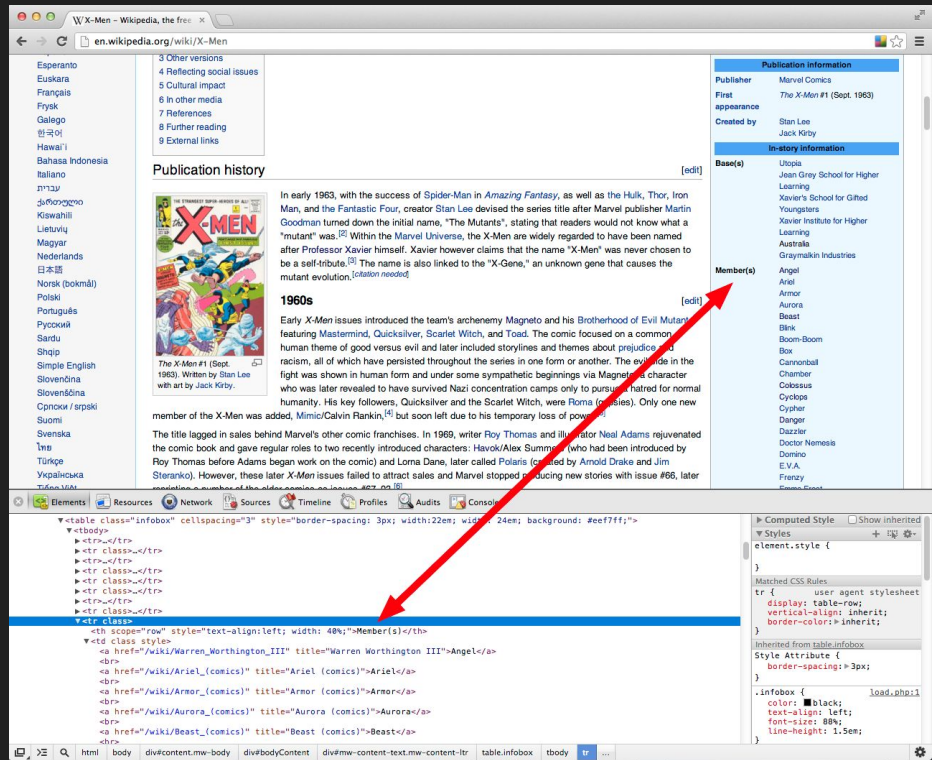
- Idea: Run exploratory scripts/commands on your unclean data to learn more about it (especially if it’s really big)
- Check for:
  - Percent of missing/corrupted/duplicate data
  - Outliers in numeric data
- See if others have commented on the quality of the data

Sampling your data

- Don’t experiment with all your data at first
- Take samples, use unix tools like head, tail, etc
  - The cluster has a script called sample which takes 1% of a file
- You want to test your code fast

Scraping

- When our data source isn't a nicely downloaded file
- Get it straight from the Internet
- Multithread it
- Run it on a cluster (Not Hadoop)



Scraping Tools / Libraries

- BeautifulSoup
- Scrapy
- Potential Issues:
  - Rate limits (Exceed request limits)
  - Is it always legal?

```
1 from bs4 import BeautifulSoup
2 import requests
3
4 r = requests.get("https://en.wikipedia.org/wiki/Main_Page")
5 soup = BeautifulSoup(r.text, "html.parser")
6 other_areas = soup.find("div", {"id": "mp-other"})
7 for li in other_areas.find_all("li"):
8     print li.find("b").text
```

Relational Data

- If data is structured like SQL
  - Has rows and columns
  - It has relationships like IDs which tie together separate columns
- Use Hive or Postgres or something similar to process it
  - You want a traditional SQL database

Document like Data

- Is not inherently structured
- Basically a bunch of text data
- You normally have it when you get raw data
- Parse it as XML
  - (using BeautifulSoup, for example)

```
<contact>
  <firstname>Bob</firstname>
  <lastname>Smith</lastname>
  <phone type="Cell">(123)
555-0178</phone>
  <phone type="Work">(890)
555-0133</phone>
  <address>
    <type>Home</type>
    <street1>123 Back St.</street1>
    <city>Boys</city>
    <state>AR</state>
    <zip>32225</zip>
    <country>US</country>
  </address>
 </contact>
```

Document-like Data Examples

```
{
   FirstName: "Bob",
   Address: "5 Oak St.",
   Hobby: "sailing"
}
```

OR

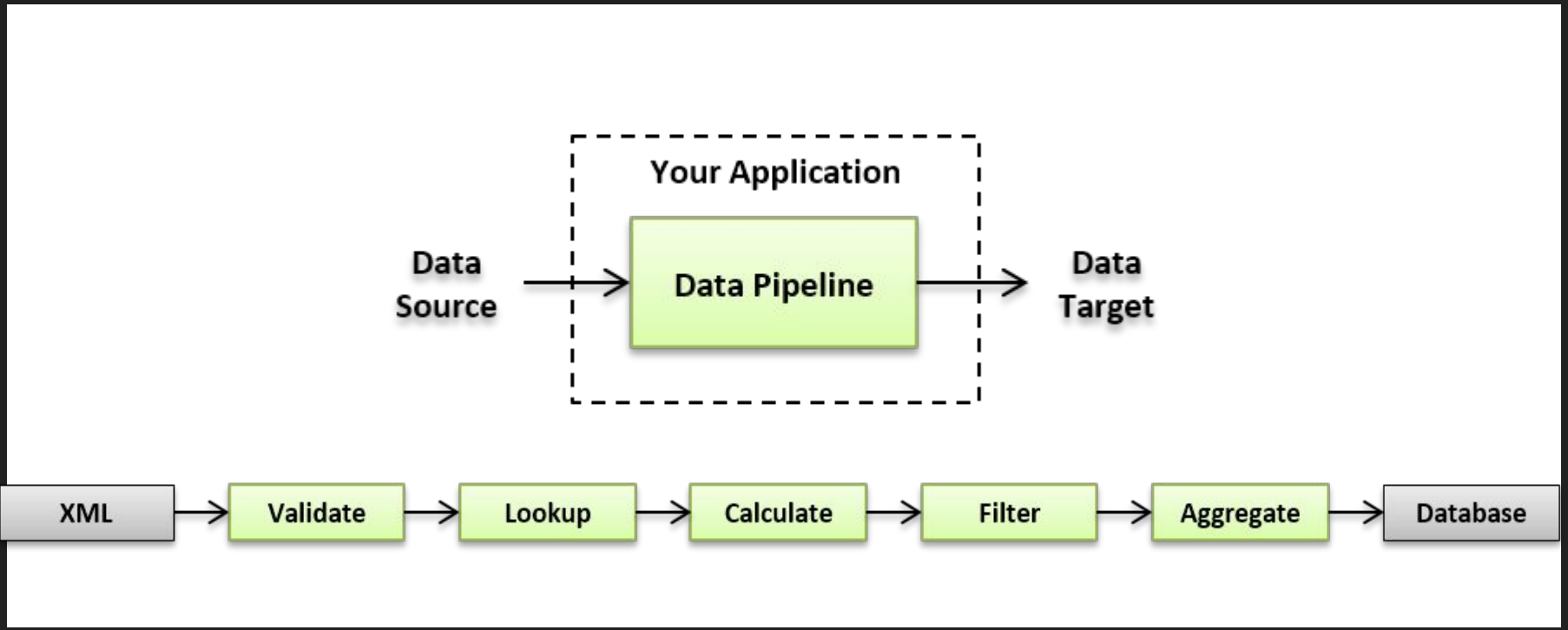
2009-06-30 23:59:57 http://twitter.com/faithkenia QUILT FOR SALE - Nine 9 patch variation vintage bed quilt hand quilted - Go to http://bit.ly/81tnp

2009-06-30 23:59:57 http://twitter.com/lnjnana Wanting to sell a timeshare, $5000, Cabo.

2009-06-30 23:59:57 http://twitter.com/twisted4you @YoungQ Still waiting for my chance to meet ya. I've been to 5 shows during Full Service and no love from Rob. Hope to meet ya in Houston!

2009-06-30 23:59:58 http://twitter.com/troubledyouth1 Trying to figure out how to best use this. Is it worth my time?

Data processing pipelines



Transferring all that data

- When data gets really really big, HTTP downloads and USBs don’t cut it
- If you have a super fast network between the computers
  - Use a torrent system
  - Use HDFS distributed copy
- Otherwise

“Never underestimate the bandwidth of a station wagon full of tapes hurtling down the highway” - Andrew S. Tanenbaum

Anonymizing Datasets

- If you ever reach the point where you are releasing a sensitive dataset
  - Reduce the accuracy of identifiable columns
    - Ex: instead of an exact age, state a range the age is in
  - K-Anonymity - Make sure any record does not have a unique set of identifiable columns
    - Ex: If you have columns of location, age, gender make sure every record has at least N other records which have the same values
  - Average values which are close to each other
  - Encrypt or hash or with a salt

Deanonymizing Datasets

- How not to do anonymization
  - NYC released a ton of taxi data with license numbers and medallion IDs
  - They hashed identifiable data using the MD5 hash function
  - Unfortunately license numbers and medallion IDs follow a pattern
  - So you can use MD5 to generate every possible hash and deanonymize it
    - Like rainbow tables
- Netflix has a database of ratings for movies
  - You can deanonymize by tracking correlation between IMDB ratings and Netflix ratings

Lab 3 - Twitter

- We’re giving you a large set of tweets from Twitter over several months
- This is the first real dataset which you do not want to run on your laptop
- We want to stress test the server more so for this week’s lab only use the cluster for testing
- Use the `sample` script so you don’t test on all the data at once

Debugging Hadoop

The stack traces are very very long, but there are a few easy ones

Error Launching job : Input path does not exist:

Load the input file into HDFS using hdfs dfs -copyFromLocal

Error Launching job : Output directory hdfs://192-168-100-234.local:8020/out3 already exists

Delete the folder or rename your output folder

hdfs dfs -rm -r out3

More debugging

java.lang.RuntimeException: PipeMapRed.waitOutputThreads(): subprocess failed with code 1

If it happened and map % was less than 100% it’s an error in your map script

If it happened and map % == 100 it’s probably your reducer script

Use unix commands to test on a sample of data

head book.txt | ./mapper.py # for mapper error

head book.txt | ./mapper.py | sort -n -k 1 | ./reducer.py | sort -n -k 2 | tail -10 # for reducer error