DS2500
2/18 - Tues.

Admin

- exams back tomorrow
- mini prez:
  - sign up to present
  - slides (PDF) due 2/23 9pm (nothing later!)
  - presos 2/24, 2/25
- semester project
  - proposal due 2/28 9pm (nothing later!)

Agenda

1. Sentiment Analysis
2. web scraping
3. Python

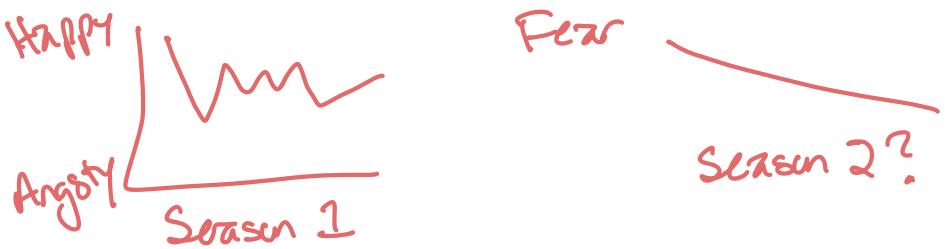
} data sources

1. Sentiment Analysis

↳ algorithm in Natural Language Processing (NLP)

Code for this week:

- Should Carey keep watering the Bear?
  ↳ do the vibes change over time?



- figure out: 👍👎 tell us the mood of the episodes based on text descriptions

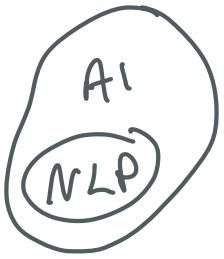
🧍 - read ep description
- tell you if happy/sad

🧍 good
👎 bad

NLP - spoken, written, humans with each other

can't capture meaning

lacks precision, add nuance/sarcasm/jokes



→ data science makes us better

meaning not important

Speech to text
one-to-one translation

meaning matters

- care about the meaning behind what was said (sentiment analysis)

Sentiment analysis

ex) Polarity score -1 to +1
one score per phrase, sentence, etc.

nice weather!

fun mail :)

SPRING BREAKKK

+1 nice ½ = .5

+1 fun ½ = .5

0/2 = 0

Algorithm:

- ignore caps, punc, stopwords
- add 1 for pos word
- sub 1 for neg word
- divide by # words

In Python: TextBlob

```
blob = TextBlob("...")
blob.sentiment.polarity   -1 to +1
blob.sentiment.subjectivity
                          0 to +1
                          fraction
```

What would make this algo better?

- emphasis words (very, really)

- slang words (ex: fire, gas, dope)
- different levels for good/bad
- text speak lol, omg, lmao :)
  imho, imo, tbh
- negation words (not, don't)

2. web scraping

↳ assume: no CSV file of episodes
but, there is a wiki!

CSV file: structured data

~, ~, ~
~, ~, ~
~, ~, ~

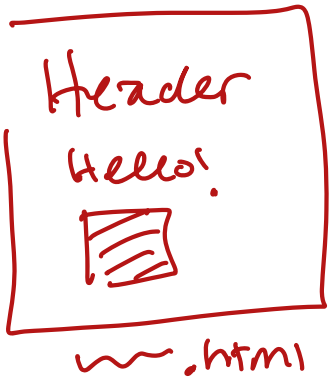
web page: semi-structured

- tags identity elements
  <~> open </~> close
- text in between tags
- created by humans
- Find the tags we need, scrape the text in between

HTML crash course

```
<html>
  <title> My Page </title>
  <body>
    <h1> Header </h1>
    <p> Hello! </p>
    <img src='~.jpg'>
  </body>
</html>
```

source of web page



~.html

The Bear Wiki

- look in source for the tags in a description

 ~~ ~~ ~~

- find all tags
- scrape text in between

3. Python

- sentiment analysis - how sophisticated?
- mini web scraping
- scrape The Bear wiki, sentiment score for every episode of season 2

Install:

- beautifulsoup4
- textblob
- wordcloud