DS2000

11/22 - Fri :)

Admin

- final 2 lectures: 11/26 and 12/3
  - 11/26 → linear regression
  - 12/3 → ? fun :)
- Please do TRACE

Agenda

1. Recommendation Algorithm
2. Class methods
3. Python
4. TRACE :)

1. Recommendation Algorithm

→ TV show, product to buy, accounts to follow, playlists, college to apply, course to take, big life decisions

Assume: 🧍 are good at this

- if they know info about you, or the show(s), a friend can make a good recommendation

🧍 trust 🧍 w/recommendations

- rec from a friend > rec from an algorithm

rec system is ML algorithm

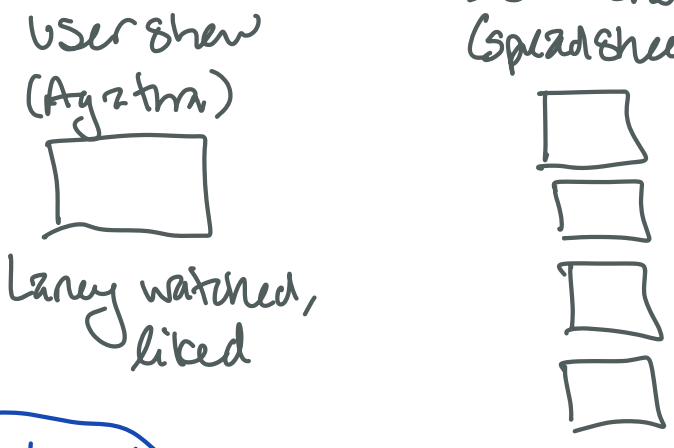
→ use experience/knowledge to make decisions

current: morning 0 afternoon 1

next: Laney promises to watch ≥1 episode of each section's rec during break

Data → about ppl, shows
↳ title, rating, sentiment, length, #seasons, year

Algorithm



For each in DB...
- is it good? (rating)
- how similar to user show?
- if similar enough, recommend!

Content-based filtering

2. Class methods

last time... TVshow Class
method: __init__
attrs: title, rating, tone, length, seasons, year, ...

Driver
- create a TVshow object

Starter code:
- read csv file
- make list of dictionaries {title: ..., rating: ...}
- make list of TVshows

Algorithm → method (recommend)

↳ DB of shows

```
def recommend(self, other):
```

↳ DB show

↳ user show

- do we recommend **self** given **other**?
- return a boolean yes/no
- calling the method show.recommend(zgatha)

Yes

self.rating > 5

self.tone vs. other.tone within 3

self.seasons vs. other.seasons within 2 or 3

and

No

self.rating ≤ 5

tone > 3 apart

seasons > 2 or 3 apart

or