Pittsbook – Cover Letter

Data Focused Python Final Project

Group C14

09/17/2022

Yi Zhou   Veronica (Yujie) Wang   Qi Zheng   Jiaqi Li

Our app Pittsbook is an app to give recommendations for locals to travel better and live better. We want to connect consumers with Pittsburgh's tourist attractions and surrounding restaurants to provide a better travel experience. To better give suggestions and design our ranking rule, we need to understand the attractions, restaurants, scores and reviews and over safety for our users.

This is how we want to clean and merge the data for each data source:

- Yelp (web scraping)
  - This data describes the Yelp recommended restaurant in the Pittsburgh area, which contains a total of 240 restaurants with their names and link to the business page, further parsing will be continued on each business site, in order to collect rating and customers' reviews as part of our own rating system. Currently, when using Yelp to search restaurants in Pittsburgh, PA, a maximum of 10 businesses was returned on a single page, so a function was written to loop through all 24 pages for exhaustive results. The data was scraped using beautiful soup (bs4) and the code for scraping and exporting can be found in "yelp_scrap".ipynb. The cleaned data can be found in "yujie_yelp_clean.xlsx".
  - Reference:
    - https://beautiful-soup-4.readthedocs.io/en/latest/#kinds-of-objects
    - https://morioh.com/p/135006369263
    - https://towardsdatascience.com/web-scraping-with-python-a-to-copy-z-277a445d64c7
  - Clean data prototype:
    - https://www.yelp.com/search?find_desc=Restaurants&find_loc=Pittsburgh%2C+PA
    - A collection of all 24 pages can be found in yujie_raw.txt

- Crime case (web scraping)

  The data describes the 2018 crime number in Pittsburgh, PA and the surrounding areas, which contains the number of murders, rapes, robberies, assaults, burglaries, thefts, and arsons.

  Data source:

  https://www.city-data.com/crime/crime-Pittsburgh-Pennsylvania.html https://tableau.alleghenycounty.us/t/PublicSite/views/CJ_UCR_PGH_8-22-17_v3/Home_1?iframeSizedToWindow=true&%3Aembed=y&%3AshowAppBanner=false&%3Adisplay_count=no&%3AshowVizHome=no&%3Aorigin=viz_share_link

Here is the scraping code:

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
html = urlopen('https://home.treasury.gov/'
'resource-center/data-chart-center/'
'interest-rates/TextView?type=daily_treasury'
'_yield_curve&field_tdr_date_value=2019')
bsyc = BeautifulSoup(html.read(), "lxml")
fout = open('D:/Universities/CMU/CMU/Python focues on data/bsyc_temp.txt',
'wt',encoding='utf-8')
fout.write(str(bsyc))
fout.close()
table_list = bsyc.findAll('table')
print('there are', len(table_list))
table = table_list[0]
print(table)
rows = table.findAll('tr')
print('there are', len(rows), 'table rows')
headers = rows[0].findAll('th')
print('there are', len(headers), 'columns')
for h in headers:
print(h.contents)
for row in rows[1:]:
    row_data = row.findAll('td')
    print(row_data[0].contents[0].contents)
    for d in row_data[1:]:
    print(d.contents)
```

- Wikipedia
  Wikipedia provides basic descriptions of places of interest as well as historical and cultural information. In our product PittsBook, we wanted to include brief descriptions of attractions to give users a better understanding of the characteristics of the places they visit, to better connect with different user needs, and to optimize their travel experience. This feature can be achieved by crawling wikipedia content. Specifically, we will implement a loop in the generated list of

attractions, define the names of the attractions as keywords, connect to wikipedia search, implement a web text crawl and then match them with specific attractions. The final result is a simple introduction to the attraction.

  - Data source: https://www.wikipedia.org/
  - Reference:
    https://levelup.gitconnected.com/two-simple-ways-to-scrape-text-from-wikipedia-in-python-9ce07426579b
  - Clean data prototype:
    Matching scenic area descriptions with attraction names.
    May add location or zip code (still discussing).

- Opentable (web scraping)
  OpenTable is an online restaurant-reservation service company for both customers and restaurants. We want to get access to users' search data for restaurants and reservations based on such parameters as dates, times, cuisine, reviews and price range. The scoring data from this data source can be merged with yelp and then we can decide our own scoring system accordingly.
  - Data source: https://www.opentable.com/pittsburgh-restaurants
  - Reference:
    https://github.com/faymatt99/opentable_scraping_analysis/tree/master/restaurants_data
  - Clean data prototype:
    - convert price_tier to numeric
    - clean name field and remove
    - remove duplicates
    - create weighted overall rating column
    - clean query fields out of restaurant page url
    - merge restaurant info with first review date
    - create a new field calculating reviews/day