


Apartment-Info Scraping & Visualization

www.apartments.com

Group Members: Zhaoyu Qiao, Guoyu Wu, Tianying Chu, Yanyi Qian



Menu

Español



Sign Up / Sign In

Add a Listing

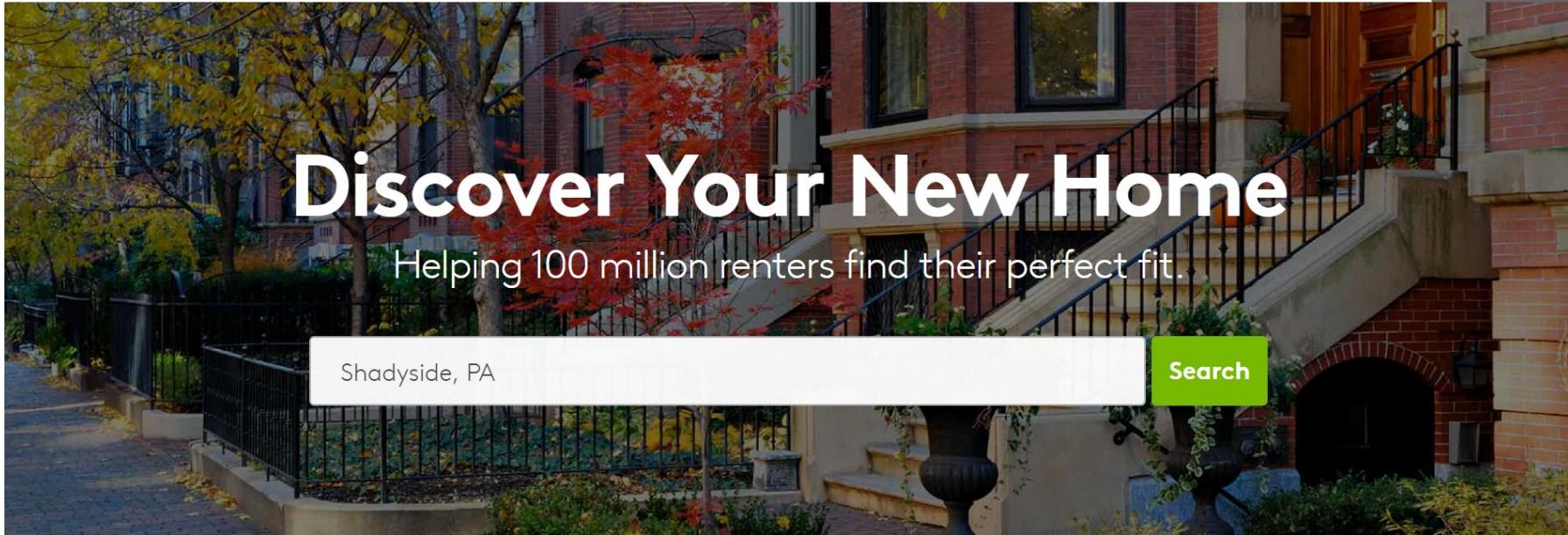


Photo by: [J. M. Garg](#)

Explore Rentals in Shadyside, PA

- Existing websites lack the combination of information and fail to customize apartment features to users' preferences.

Content

- Background
- Data Collection
- Data Processing
- Two Use Cases

Data sources

- **Web Scraping:** apartments.com
- **API:** Google Maps API
 - Geocoding API / Places API
- **CSV file:** Pittsburgh Police Arrest Data.csv
 - downloaded from Data.gov
- **Others:** Pittsburgh City Council Boundary Geojson data
 - download from Pittsburgh open data

Libraries Used

Data Collection, Cleaning and Exploration Part

- requests
- bs4 (BeautifulSoup)
- time
- pandas
- numpy
- seaborn

Mapping and Apartment Recommendation Part

- pandas
- requests
- json
- folium
- shapely.geometry
- geopandas

Let's begin scraping!

Data Collection

Two files:

- **Scraping_Listpage.py**

Scraping the urls for the subpage and the address

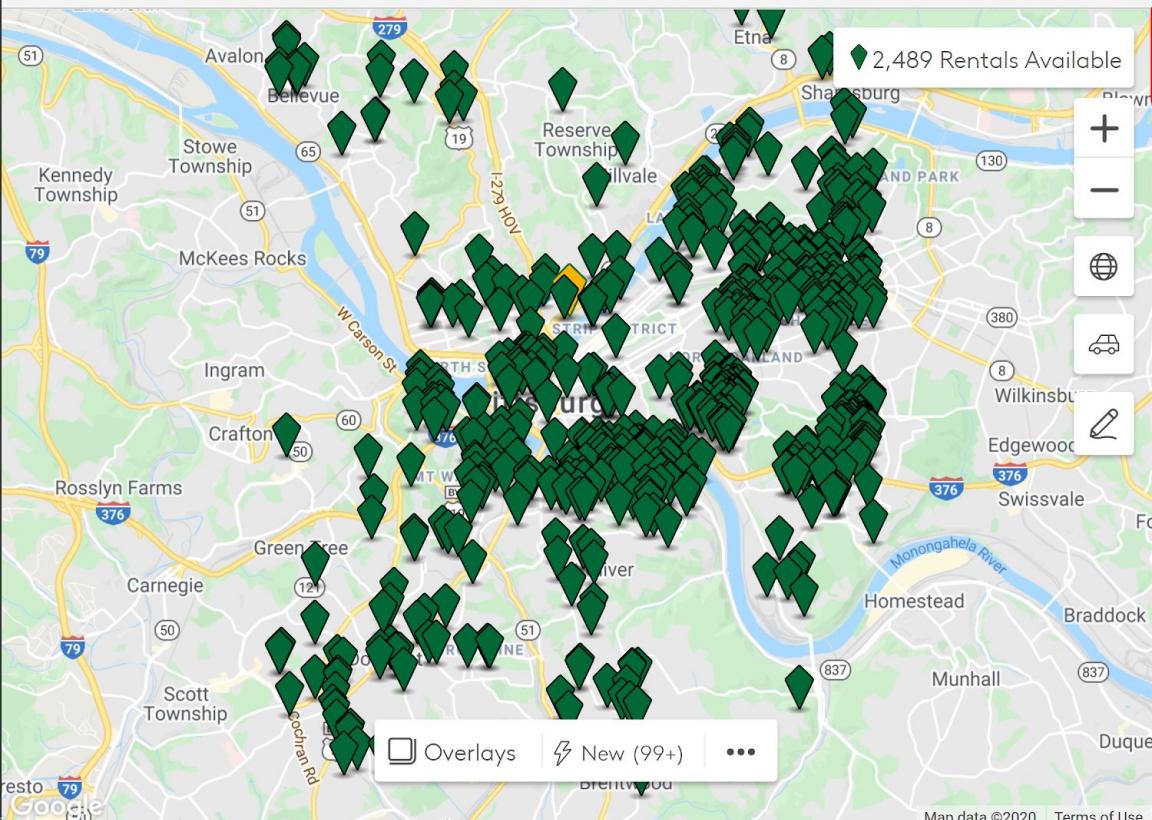
- **Scraping_Subpage.py**

Scraping the basic information of each apartment

Location or Point of Interest Price Beds Type More 

Sort

Save



Map data ©2020 Terms of Use

Url & Location

Park View Apartments

10 Allegheny Ctr, Pittsburgh, PA 15212

1/104



NEW

\$850 - 2,645

Studio - 2 Bed
Available Now

844-743-7158

 Email

3D

Heinz Lofts

300 Heinz St, Pittsburgh, PA 15212

1/57



C 7 HRS AGO

\$1,450 - 5,050

1-3 Bed
Available Now

412-847-5081

8

[← Map](#)

Menu

[Español](#)[Sign Up / Sign In](#)

Park View Apartments

[Verified ✓](#)

10 Allegheny Ctr, Pittsburgh, PA 15212 – Allegheny Center

★★★★★ (20 Reviews) ▾

Studio \$850 - 2,390 1 Bedroom \$1,075 - 2,420 2 Bedrooms \$1,400 - 2,645



FAROS
PROPERTIES

Rent_by_room

[Home](#) / [Pennsylvania](#) / [Pittsburgh](#) / Park View Apartments



🕒 Today

All

Studio

1 Bedroom

2 Bedrooms

Studio 1 Bath \$850 - 1,650

415 Sq Ft

ST A1

Available Now

[View Model](#)

📞 844-743-7158

First Name *

Last Name *

Email Address *

Move-in Date *

04/01/2020



Phone

I would like to...

Schedule a Tour

Request Application

Confirm Availability

Custom Message

[Send Message](#)

Email me listings and apartment related info.

[View Property Website](#)

Studio \$850–2,390 1 Bedroom \$1,075–2,420 2 Bedrooms \$1,400–2,645



Property Ratings at Park View Apartments

[ⓘ How is this rating determined?](#)

4.5
Great

Out of 5

Average Rating



4.5 Average Rating

20 Renter Reviews

Share details of your own
experience with this property

[Write a Review](#)

Most Positive Review



04/05/2018

Great Experience

I relocated to Pittsburgh and
luckily discovered this apartment
complex. I really enjoyed m)

Most Recent Review



09/25/2018

**Don't live here if you can help
it.**

Don't live here if you can help it. I
noticed the ceiling over my bed

Most Helpful Review



08/29/2015

**Location Is good, but I don't
recommend**

I love the location and
spaciousness of these apartments.

844-743-7158

First Name *

Last Name *

Email Address *

Move-in Date *

04/01/2020



Phone

I would like to...

Schedule a Tour

Request Application

Confirm Availability

Custom Message

[Send Message](#)

Email me listings and apartment related info.



[View Property Website](#)

Studio \$850–2,390 1 Bedroom \$1,075–2,420 2 Bedrooms \$1,400–2,645



Walkability Near Park View Apartments

WALK SCORE®

**88**

Very Walkable

WalkScore

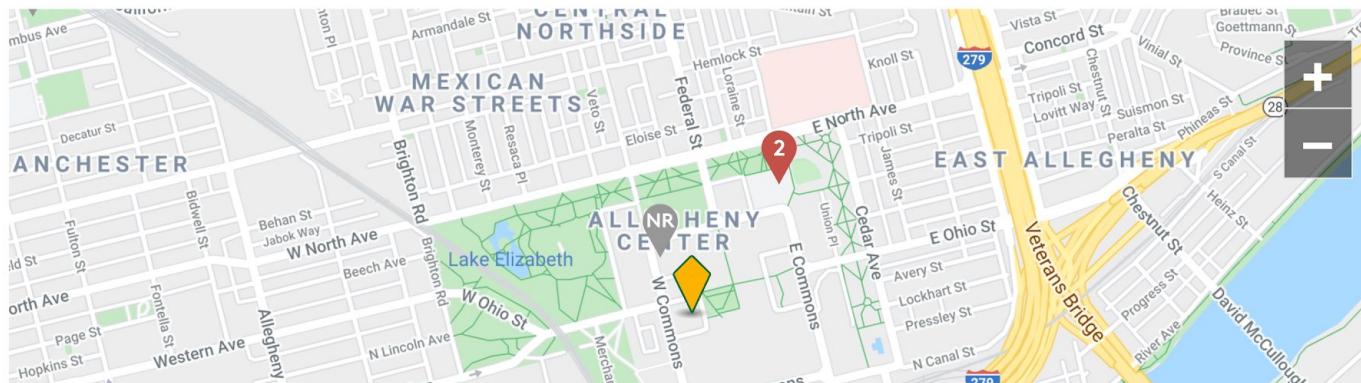
TRANSIT SCORE®

**83**

Excellent Transit

TransitScore

Points of Interest Near Park View Apartments



844-743-7158

First Name * Last Name *
Email Address *
Move-in Date * Phone
I would like to...
 Schedule a Tour Request Application
 Confirm Availability Custom Message
Send Message
 Email me listings and apartment related info.

[View Property Website](#)

Data Cleaning

- Split the column ‘Rent_by_room’ into 5 new columns (‘Studio’, ‘1 Bed’, ‘2 Beds’, ‘3 Beds’, ‘4 Beds’)
- Remove apartment records without detailed location
- Remove abnormal value in ‘WalkScore’ and ‘TransitScore’

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 179 entries, 0 to 187
Data columns (total 21 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   Url              179 non-null   object  
 1   Title             177 non-null   object  
 2   Company           112 non-null   object  
 3   Location           179 non-null   object  
 4   Last_Updated       179 non-null   object  
 5   Rent               179 non-null   object  
 6   Unit               179 non-null   object  
 7   Availability       121 non-null   object  
 8   Phone              179 non-null   object  
 9   AverageRating      178 non-null   float64 
 10  # of Reviewer     110 non-null   object  
 11  WalkScore          179 non-null   object  
 12  WalkScore_desc     179 non-null   object  
 13  TransitScore       179 non-null   object  
 14  TransitScore_desc  179 non-null   object  
 15  Rent_by_room       179 non-null   object  
 16  Studio              70 non-null    float64 
 17  1 Bed              161 non-null   float64 
 18  2 Beds             156 non-null   float64 
 19  3 Beds             56 non-null    float64 
 20  4 Beds             6 non-null     float64 
dtypes: float64(6), object(15)
memory usage: 30.8+ KB
None
```

Data Exploration (Apartments data)

- Missing value:

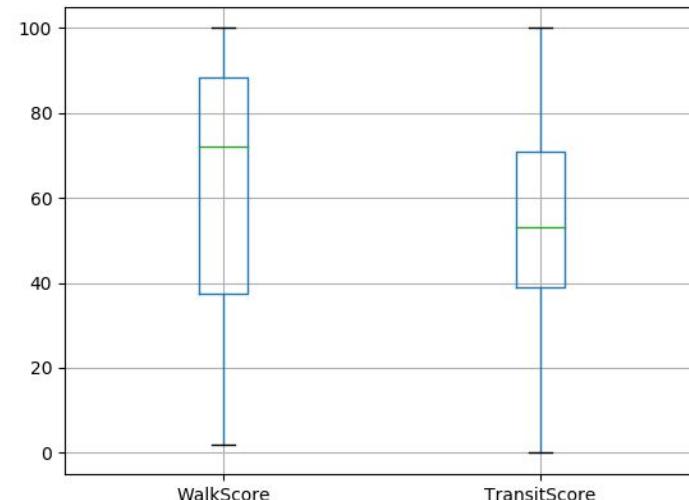
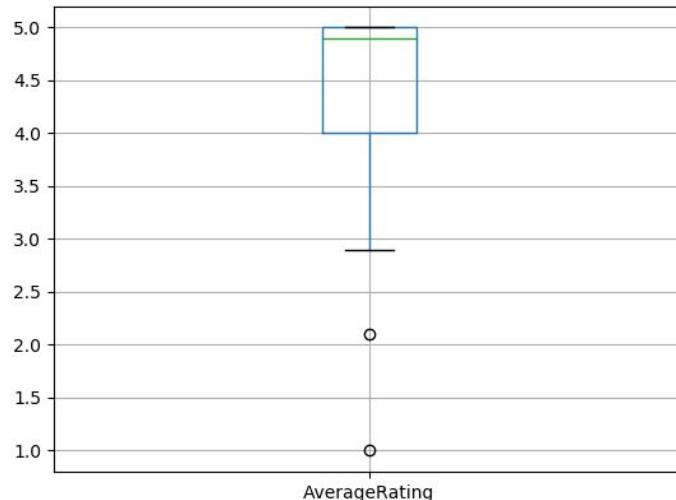
Unnamed:	0
Url	0
Title	2
Company	67
Location	0
Last_Updated	0
Rent	0
Unit	0
Availability	58
Phone	0
AverageRating	1
# of Reviewer	69
WalkScore	0
WalkScore_desc	0
TransitScore	0
TransitScore_desc	0
Rent_by_room	0
Studio	109
1 Bed	18
2 Beds	23
3 Beds	123
4 Beds	173

- Data Description:

	AverageRating	WalkScore	TransitScore
count	178.000000	179.000000	179.000000
mean	4.510674	63.078212	55.703911
std	0.621869	28.345273	23.243134
min	1.000000	2.000000	0.000000
25%	4.000000	37.500000	39.000000
50%	4.900000	72.000000	53.000000
75%	5.000000	88.500000	71.000000
max	5.000000	100.000000	100.000000

Data Exploration (Apartments data)

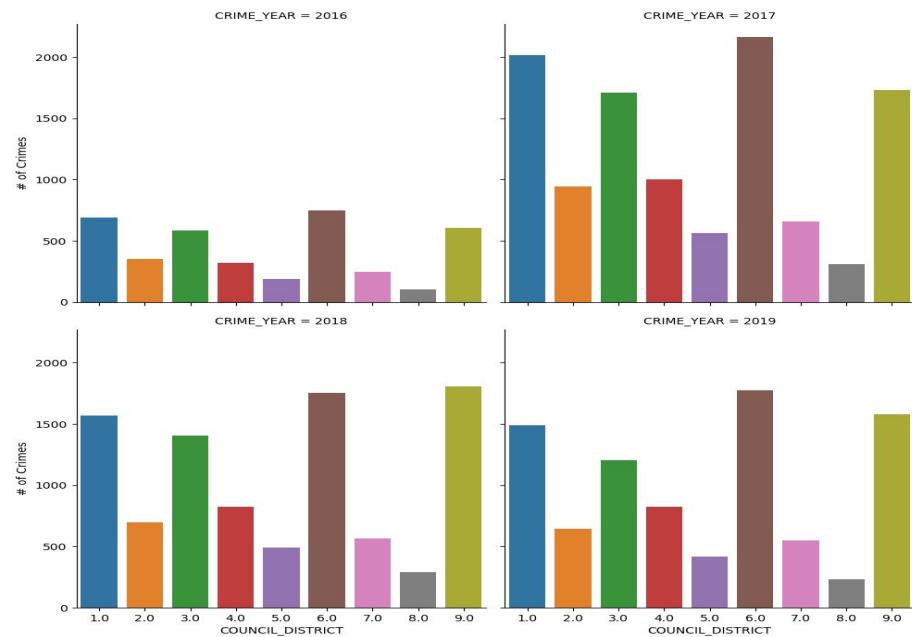
- Data Description (cont.):



Data Exploration (Crimes data)

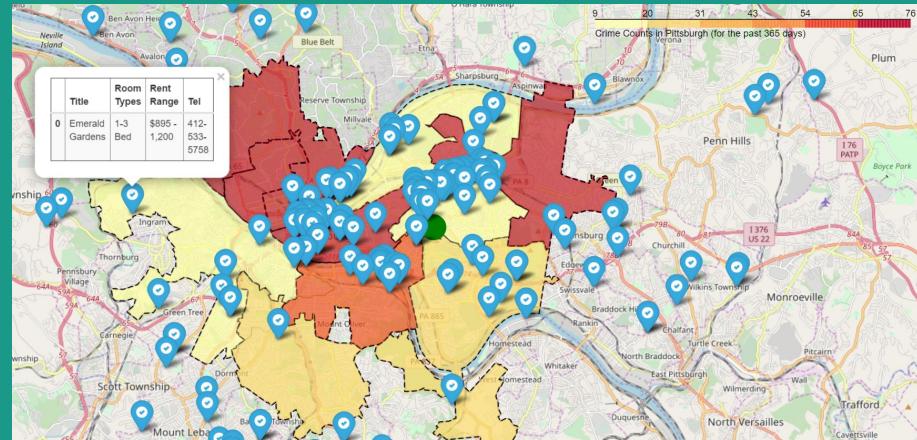
- Missing value:

PK	0
CCR	0
AGE	302
GENDER	0
RACE	0
ARRESTTIME	0
ARRESTLOCATION	0
OFFENSES	3
INCIDENTLOCATION	0
INCIDENTNEIGHBORHOOD	1482
INCIDENTZONE	59
INCIDENTTRACT	1460
COUNCIL_DISTRICT	5124
PUBLIC_WORKS_DIVISION	5124
X	1398
Y	1398



Use Cases

1



2

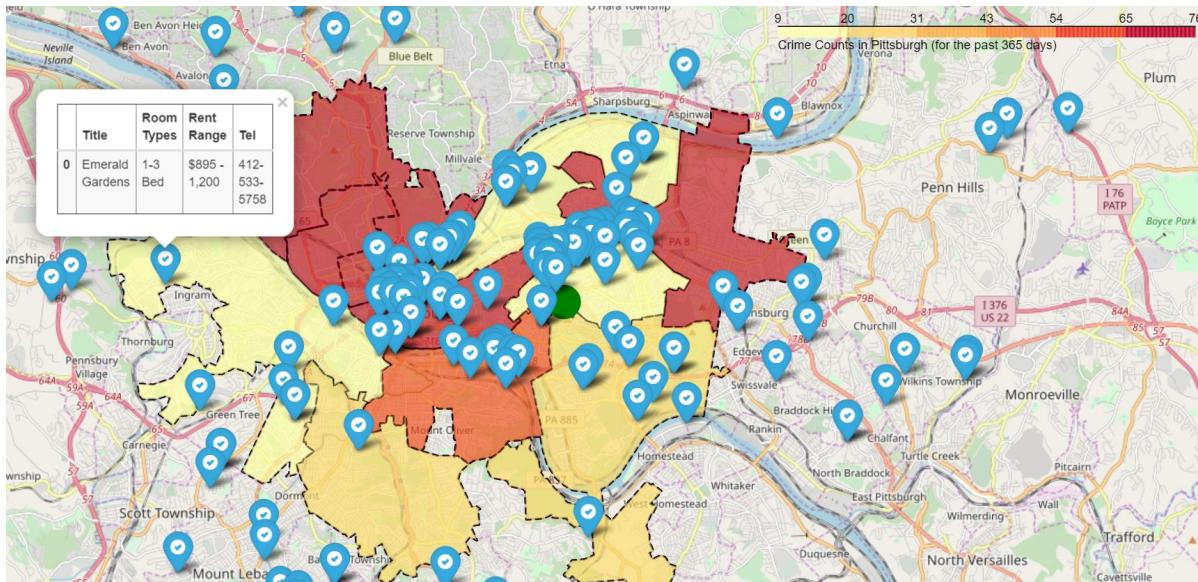
please enter your targeted room type:
Enter 1 to denote Studio
Enter 2 to denote 1 bedroom
Enter 3 to denote 2 bedroom
Enter 4 to denote 3 bedroom
Enter 5 to denote 4 bedroom

your choice: _____

Use case 1: We care about your safety!!

Basic Concept:

Mark the available apartments on Crime Heatmap.



Realization methods:



- pandas,requests, json
- read Cleaned_Data.csv to df
- **def find_lat_lng(address):**
- apply the function to apartment address
- df.to_csv('apt_data.csv')

- pandas, **folium**; “PGH_CityCouncil.geojson”
- read apt_data.csv and crime_data.csv to df
- draw the boundary lines of Pitts with **folium.GeoJson()**
- add the apt data as markers with **folium.Marker()**
- color the map with crime data with **folium.Choropleth()**
- add pop-up info (dataframe type)
- *Don't forget to mark the location of Hamburg Hall! :)*

How we use Geocoding API?

```
4 def find_lat_lng(address):
5 # input formatted address of the location (string type)
6 # output Latitude and Longtitude of the input address
7
8     # apply your API KEY at https://code.google.com/apis/console
9     KEY = input('Enter your GoogleMaps API key:')
10
11    url = 'https://maps.googleapis.com/maps/api/geocode/json?'
12    parameter = address
13    r = requests.get(url + 'address=' + parameter + '&key=' + KEY)
14    geocode_result = r.json()
15 #
16 #obtain Latitude and Longtitude
17 location = geocode_result['results'][0]['geometry']['location']
18 lat,lng = location['lat'],location['lng']
19 return lat,lng
20
```

```
...: find_lat_lng('5000 Forbes Ave, Hamburg Hall, Pittsburgh, PA 15213')
Out[3]: (40.4443494, -79.9455454)
```

```
In [4]:
```

Let's have a try :)

Use case 2: We offer personalized apartment recommendation!

Basic Concept:

Find the top 10 apartments according to your preferences.

please enter your targeted room type:

Enter 1 to denote Studio

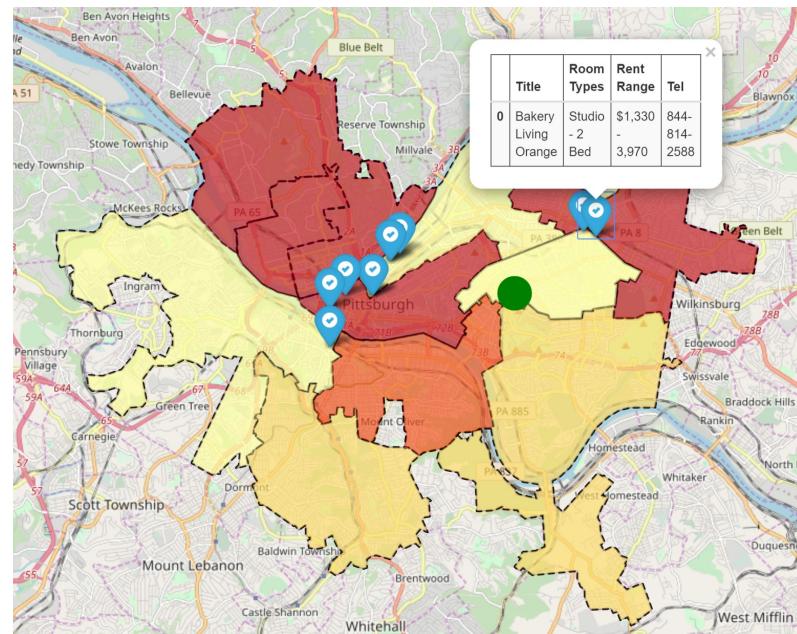
Enter 2 to denote 1 bedroom

Enter 3 to denote 2 bedroom

Enter 4 to denote 3 bedroom

Enter 5 to denote 4 bedroom

your choice: _____



Use case 2: We offer personalized apartment recommendation!

Basic Concept:

Find the top 10 apartments according to your preference.

Enter preference to each metrics

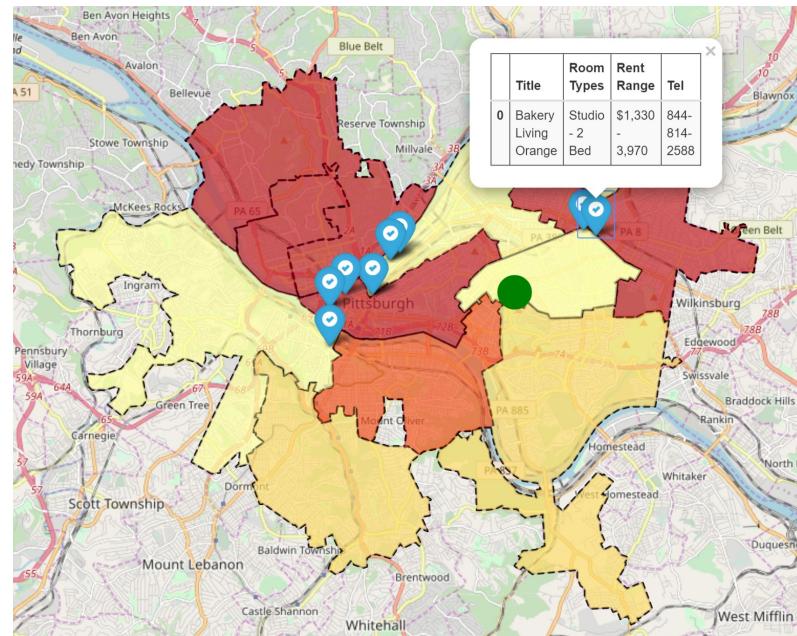
price score:5

safety score:5

distance score:3

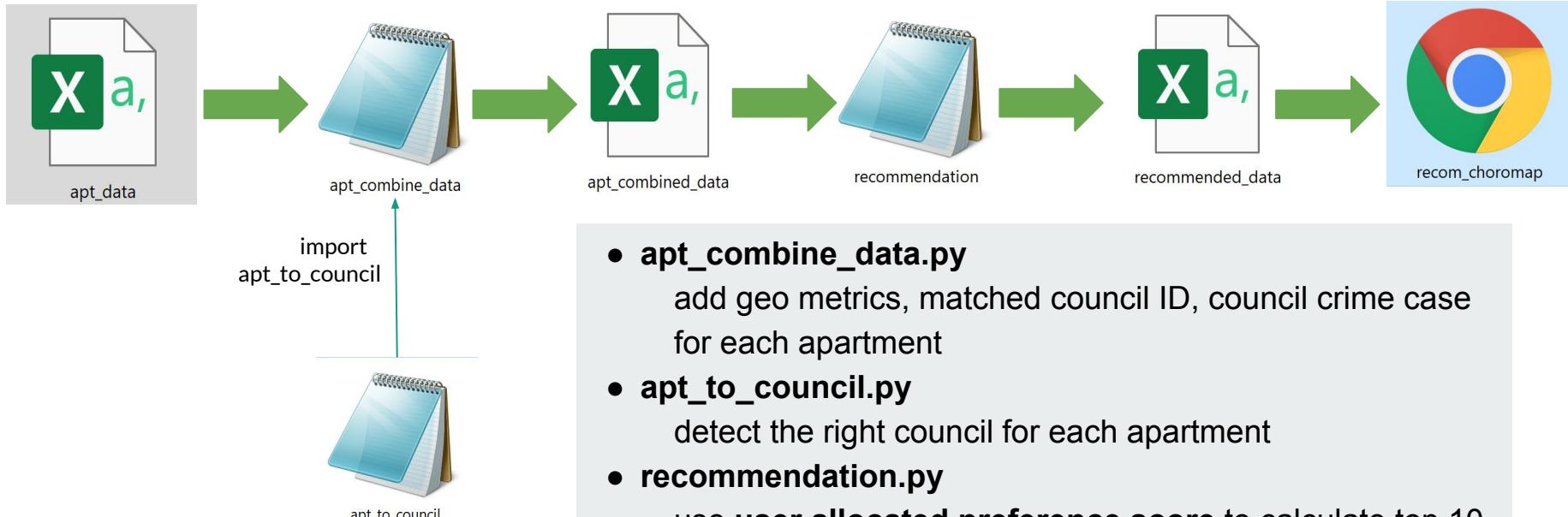
restaurant number score:4

Transit convenience score:3



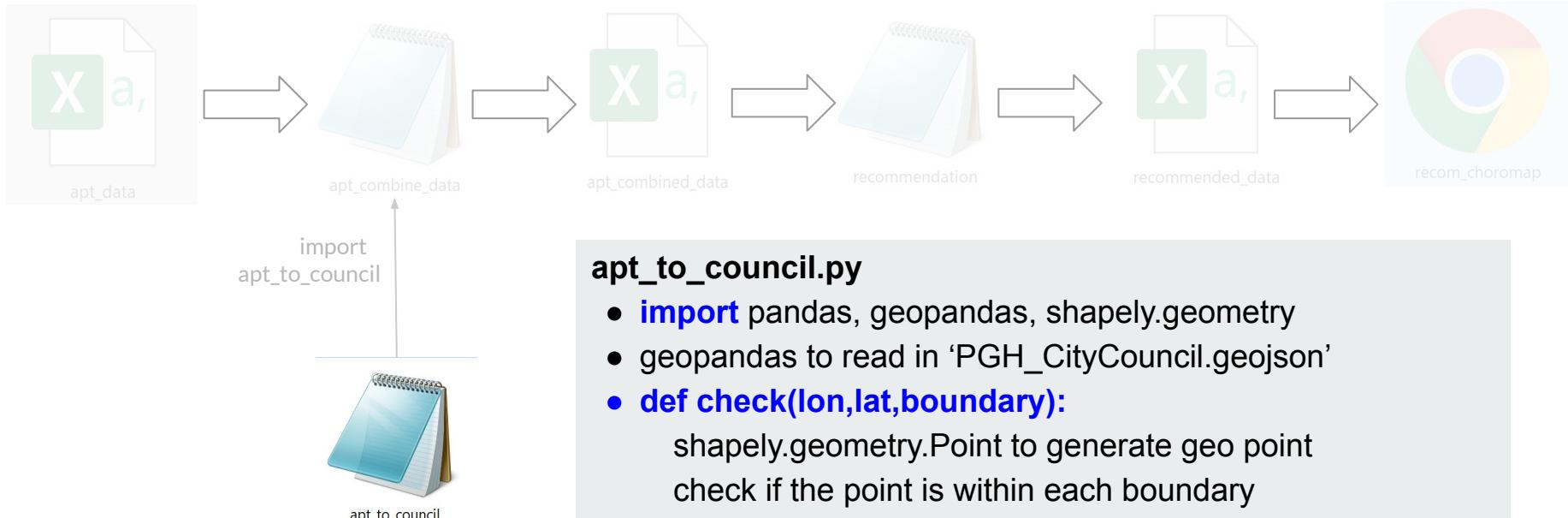
Use case 2: We offer personalized apartment recommendation!

Realization methods:



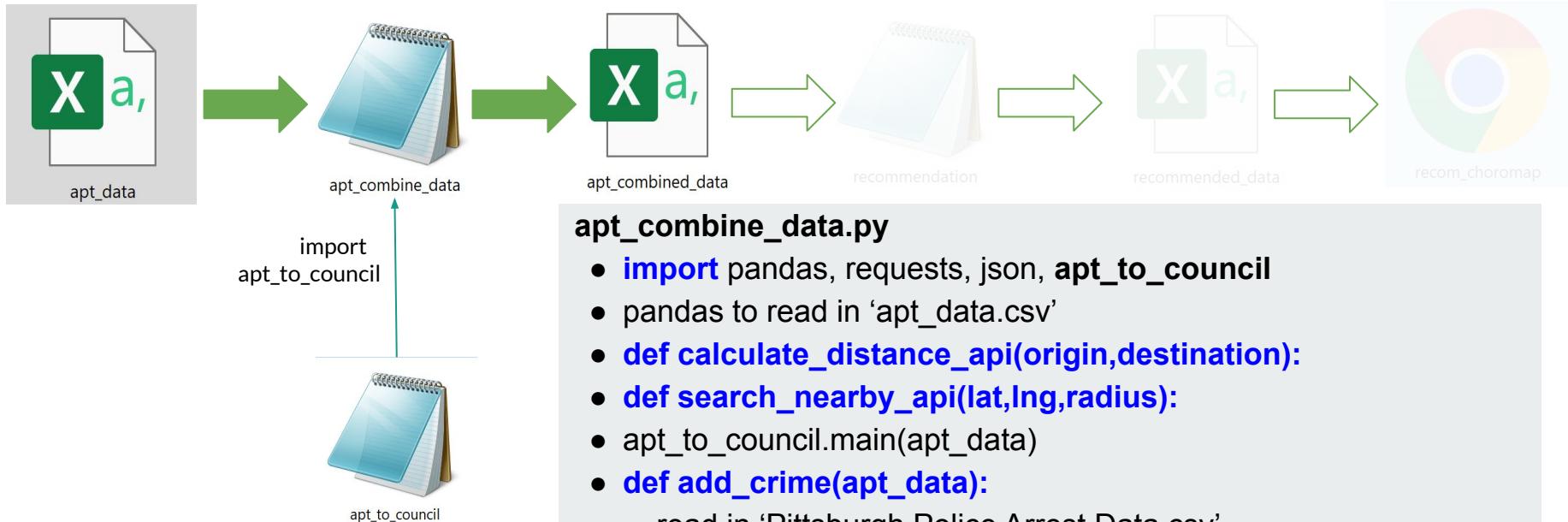
Use case 2: We offer personalized apartment recommendation!

Realization methods:



Use case 2: We offer personalized apartment recommendation!

Realization methods:



apt_combine_data.py

- **import** pandas, requests, json, **apt_to_council**
- pandas to read in 'apt_data.csv'
- **def calculate_distance_api(origin,destination):**
- **def search_nearby_api(lat,lng,radius):**
- **apt_to_council.main(apt_data)**
- **def add_crime(apt_data):**
 - read in 'Pittsburgh Police Arrest Data.csv'
 - group_by Council ID, add safety metric column to apartment
- Output apt_combined_data.csv

Use case 2: We offer personalized apartment recommendation!

Magical API!



Enabled APIs Last 30 days

API	Requests
Geocoding API	1,647
Directions API	735
Places API	621
Geolocation API	0
Maps Elevation API	0

Billing Last 3 months

Current bill \$0.00
March 01 - March 05 Total usage: \$35.84



Use case 2: We offer personalized apartment recommendation!

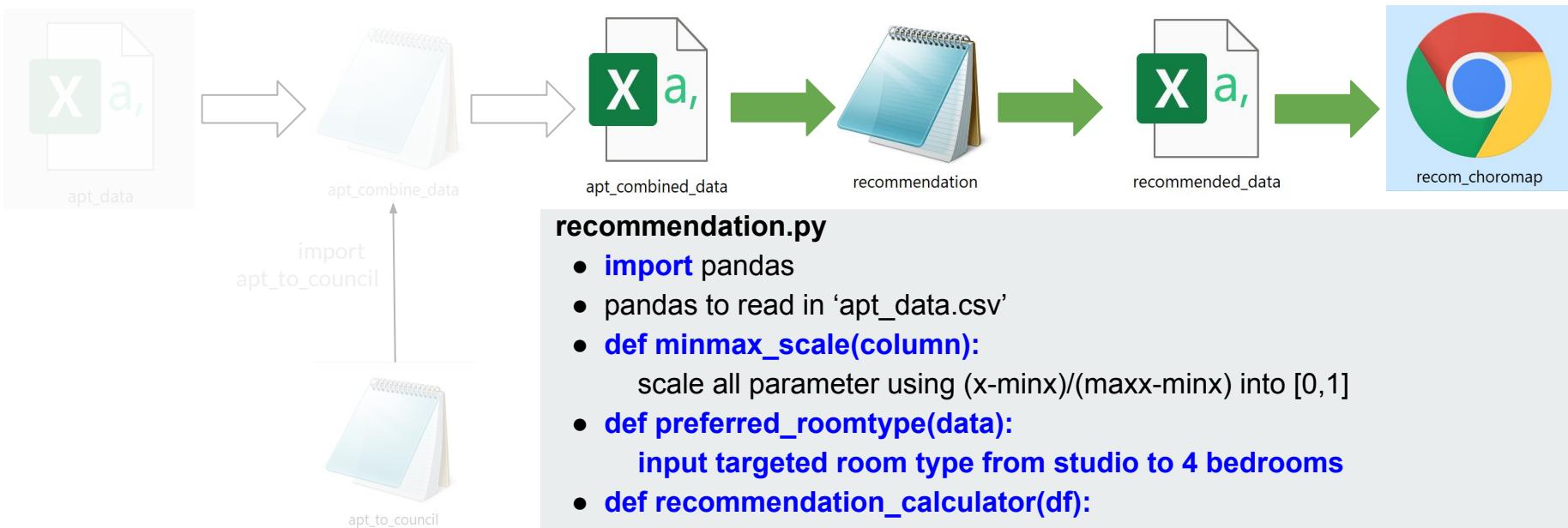
Have a look of what we found for restaurants!



Index	Type	Size	
0	dict	2	{'name':'Subway', 'price_level':1}
1	dict	2	{'name':'Lucca', 'price_level':2}
2	dict	2	{'name':'Ali Baba Restaurant', 'price_level':2}
3	dict	2	{'name':'Union Grill Oakland', 'price_level':2}
4	dict	2	{'name':'Eatunique', 'price_level':2}
5	dict	2	{'name':'Lulu's Noodles', 'price_level':1}
6	dict	2	{'name':'Chipotle Mexican Grill', 'price_level':1}
7	dict	2	{'name':'Sushi Fuku', 'price_level':NoneType}
8	dict	2	{'name':'Schatz Dining Room', 'price_level':NoneType}
9	dict	2	{'name':'The Cafe Carnegie', 'price_level':2}
10	dict	2	{'name':'El Gallo de Oro', 'price_level':NoneType}

Use case 2: We offer personalized apartment recommendation!

Realization methods:



recommendation.py

- **import** pandas
- pandas to read in 'apt_data.csv'
- **def minmax_scale(column):**
scale all parameter using $(x-\text{minx})/(\text{maxx}-\text{minx})$ into [0,1]
- **def preferred_roomtype(data):**
input targeted room type from studio to 4 bedrooms
- **def recommendation_calculator(df):**
input scores from 1-5 for preference of price, safety, distance to Heinz, restaurant, transit convenience, calculate the final score
return top 10 recommended apartments
- Output recommended_apartment html

“We aim to optimize your residential experience! ”

Thank you!