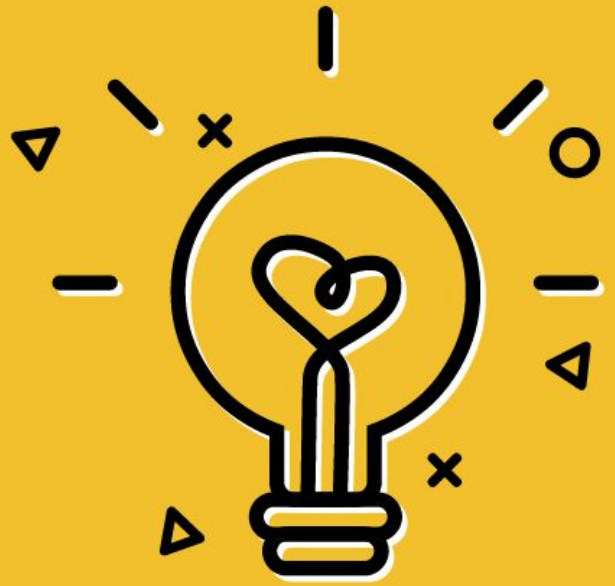


Web Scraper and Data Analysis

Python | Selenium | Chrome

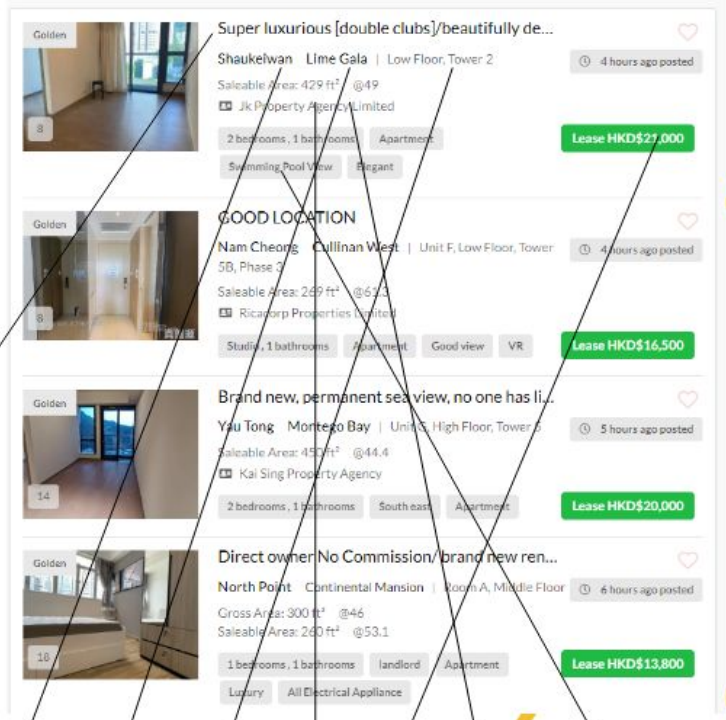
Utilizing Selenium to Scrape Real-Estate Data

Scraping Data

Selenium with a Chrome webdriver

We use Selenium with a Chrome driver to navigate to https://www.28hse.com/en/rent/residential where we scrape as many pages as we want and save it in csv format. The data saved includes:

-Description
-Link to listing page
-District
-Address line 1
-Address line 2
-Saleable and gross area
-Lease price
-Agency name
-Description tags



	A	B	C	D	E	F	G	H
1	Super luxurious [double clubs]/beaut	https://www.28hse.com/e	Shaukeiwan	Lime Gala	Low Floor, Tower 2	Saleable Area	Lease HKD$21,000	Jk Property Agen	['2 bedrooms , 1 bathrooms', 'Apartment', 'Swimming Pool View', 'El
2	GOOD LOCATION	https://www.28hse.com/e	Nam Cheong	Cullinan West	Unit F, Low Floor, Tow	Saleable Area	Lease HKD$16,500	Ricacorp Proper	['Studio , 1 bathrooms', 'Apartment', 'Good view', 'VR']
3	Brand new, permanent sea view, no	https://www.28hse.com/e	Yau Tong	Montego Bay	Unit G, High Floor, To	Saleable Area	Lease HKD$20,000	Kai Sing Property	['2 bedrooms , 1 bathrooms', 'South east', 'Apartment']
4	Direct owner No Commission/ brand	https://www.28hse.com/e	North Point		Room A, Middle Floor	Gross Area: 3 Saleable Area	Lease HKD$13,800		['1 bedrooms , 1 bathrooms', 'landlord', 'Apartment', 'Luxury', 'All El

Scraping Data

We scrape 1000 pages at once (~45 mins):
15 listings per page x 1000 pages scraped = 15000 listings.

We also do some data cleaning.

```
[3]: df = pd.read_csv('./example_data_1000 pages_scraped.csv', names = custom_columns)
```

```
[4]: df.tail(2)
```

[4]:

	description	link	district	address_1	address_2	
14998	High-rise 2-bedroom mountain view	https://www.28hse.com/en/rent/residential/prop...	Shek Tong Tsui	Lun Fung Court	Unit D, Mid Floor, Middle Floor	
14999	Convenient transportation and comfortable envi...	https://www.28hse.com/en/rent/residential/prop...	Tai Wai	Festival City	Unit Nb, High Floor, Tower 5, Phase Iii	@2

```
# Find listings with misplaced data that was pushed to blank_1 and blank_2

non_nan_mask = df['blank_1'].notnull()
df[non_nan_mask]
```

```
# Listings with only link column populated incorrectly
listings_to_fix = [807, 973, 4736, 6406, 10247, 14482]

# Set 'Link' value to None
for listing in listings_to_fix:
    df.iloc[listing]['link'] = None

    # Shift column data 1 spot to the left
    df.iloc[listing, 1:] = df.iloc[listing, 1:].shift(-1)
```

Replacing NaN values in 'tags' column with empty lists

```
df['tags'] = df['tags'].apply(lambda x: [] if pd.isna(x) else x)
```

Scraping Data

Some more cleaning – “size” column values are not uniform; some have ‘Gross’, some have ‘Saleable’, some have both!

address_2	size	price
Unit D, Mid Floor, Middle Floor	Saleable Area: 369 ft² @44.7	Lease HKD$16,500
Unit Nb, High Floor, Tower 5, Phase Iii	Gross Area: 1,273 ft² @27.5\nSaleable Area: 96...	Lease HKD$35,000

```
# Extract numerical values from df['size'], append the saleable value to s_values
# and gross values to g_values

s_values = []
g_values = []
for k, v in df['size'].items():
    if 'Gross' in v and 'Saleable' in v:
        s_values.append(int(v.split(' ')[6].replace(",", "")))
        g_values.append(v.split(' ')[2].replace(",", ""))

    elif 'Gross' not in v:
        s_values.append(int(v.split(' ')[2].replace(",", "")))
        g_values.append('')

    elif 'Saleable' not in v:
        g_values.append(int(v.split(' ')[2].replace(",", "")))
        s_values.append('')
```

```
df['saleable_sqft'] = s_values
```

```
df['gross_sqft'] = g_values
```

size	price	agency	tags	saleable_sqft	gross_sqft
Saleable Area: 369 ft² @44.7	16500	Joe Asia Property Agency	['Stand-alone Building', 'Mountain view']	369	
Gross Area: 1,273 ft² @27.5\nSaleable Area: 96...	35000	Centaline Property Agency Limited	['Apartment']	961	1273

EDA

After some more preprocessing, we can do some EDA.

See more in the Data Analysis.ipynb