Wepscraping house prices in Madrid

Habib Slim
habib.slim@grenoble-inp.org

National School of Computer Science and Applied Mathematics, Grenoble

1 Introduction

As part of the proposed webscraping challenge, we were asked to build a database with information about dwelling prices for houses and flats in a big city in the European Union, using webscraping techniques.

Madrid had a population of approximately 3.2 millions as of 2019[1], and thus qualifies for this challenge.

Only the websites providing services exclusively in the real estate category were used, and websites specialized in luxury real estate were excluded.

In order to determine which websites are more appropriate to gather information about real estate prices in Madrid, multiple metrics could have been considered to rank the most relevant sources.

The following were chosen:

- Number of estimated monthly visits (using SimilarWeb[2] as a reference)
- Number of currently available offers for houses and flats located in the Madrid capital[3].

Website name	**Monthly visits (millions)**	**Number of offers (houses)**	**Number of offers (flats)**
Idealista.com	33.50	$\approx$ 2.000	$\approx$ 23.600
Fotocasa.es	14.79	$\approx$ 1.000	$\approx$ 27.000
Habitaclia.com	9.88	$\approx$ 1.000	$\approx$ 14.000
Pisos.com	6.20	$\approx$ 1.000	$\approx$ 14.000
Yaencontre.com	3.05	$\approx$ 500	$\approx$ 6.000
Tucasa.com	2.17	$\approx$ 2.000	$\approx$ 10.000

Table 1: Basic comparison of top spanish real estate websites.

Idealista and Fotocasa were ideal candidates regarding the volume of data, however their Terms of Services explicitly forbid the use of automated scrapers on their services[4].

For this preliminary challenge, Habitaclia, Pisos and Tucasa were chosen as targets for the scraping (highlighted in gray in the previous table). The solution that I provide uses Python, BeautifulSoup and pandas - and will take at most 20 seconds to scrape 15.000 entries.

A short user manual is given in the GitHub repository for this project[5].

[1] Source: Municipal Register of Spain 2019 - National Statistics Institute

[2] See for example: `http://similarweb.com/website/pisos.com`.

[3] Those figures are directly provided by real estate websites when using their search tool.

[4] In the case of Idealista, the contents of the website can still potentially be fetched by making a formal request to use their search API.

[5] `https://github.com/HabibSlim/MadridImmo`

2 Produced dataset

2.1 Description

A short description of the produced dataset (file `"madrid_immo.csv"` in the `dataset` folder) will be given here.

Below is the list of fields provided for every entry:

- `url`: The link to the description of the offer on the realtor website.
- `address`: The address of the property to be sold.
- `loc`: The localization of the property to be sold.
- `price`: The price in euros.
- `m2`: The floor area in square meters.
- `type`: The type of dwelling ("house" or "flat").

Since a precise address for the properties is rarely disclosed by the chosen target websites, the "`address`" field is most of the time quite sparse[6]. However, a district name is always provided, which enables us to localize properties quite accurately: in order to store this information, the "`loc`" field was added.

Website name	**Number of entries (flats)**	**Number of entries (houses)**	**Total**
Habitaclia.com	13.245	1.110	14.355
Pisos.com	3.000	655	3.655
Tucasa.com	10.000	2.055	12.055
Total	26.245	3.820	**30.065**

Table 2: Number of entries scraped for each target website.

For Pisos.com, there is a fixed limit on the maximum number of pages that can be visited through a unique search (searches here are defined by the set of criteria specified by the user to select through the property database).

This could be bypassed by using multiple searches with different criteria (*example:* searching for flats with prices ranging from 0€ to 100.00€, then from 100.00€ to 200.00€, etc.) - however this potential solution to this problem has not been explored yet.

Below are some rows extracted from the dataset as an illustration:

URL	**Address**	**Localization**	**Price**	m^2	**Type**
[..]/comprar/casa-puente_de[..]	calle de mejorana, near [..]	Entrevías	128000	76.0	house
[..]/comprar/chalet_adosado[..]	hispanoamerica	Hispanoamérica	990000	287.0	house
[..]/comprar/chalet-nino_je[..]	niño jesús	Niño Jesús	1318000	263.0	house
[..]/comprar/chalet-puente_[..]	calle de membezar, near [..]	Entrevías	139000	69.0	house
[..]/comprar/chalet-chamart[..]	chamartin	Hispanoamérica	1390000	370.0	house

Table 3: Some entries from the `madrid_immo.csv` file.

[6] In the case of the entries extracted from Habitaclia.com, addresses are not disclosed at all.

2.2 Preliminary exploration

We can now do some preliminary exploration of the data scraped. In the "`mapbuild`" notebook (located in the "explore" directory of the repository) is a toy exercise with some simple manipulations on the scraped data, in order to compute the average price per square meter for every district in the city.

After some filtering, here are some of the computed values for the following districts[7]:

District	**Price per m^2 (euros)**	**Margin of error (euros, $\alpha = 0.95$)**
Arganzuela	3994.92	$\pm$ 49.50
Barajas	3067.08	$\pm$ 70.91
Carabanchel	2219.89	$\pm$ 27.01
Centro	4948.70	$\pm$ 66.58
Chamartín	5391.62	$\pm$ 77.38
Chamberí	5409.88	$\pm$ 67.29
...		
All districts	3818.42	$\pm$ 24.28

Table 4: Average price per square meter in some Madrid districts

From those values, we build the following map of average prices per square meter for every district in Madrid:

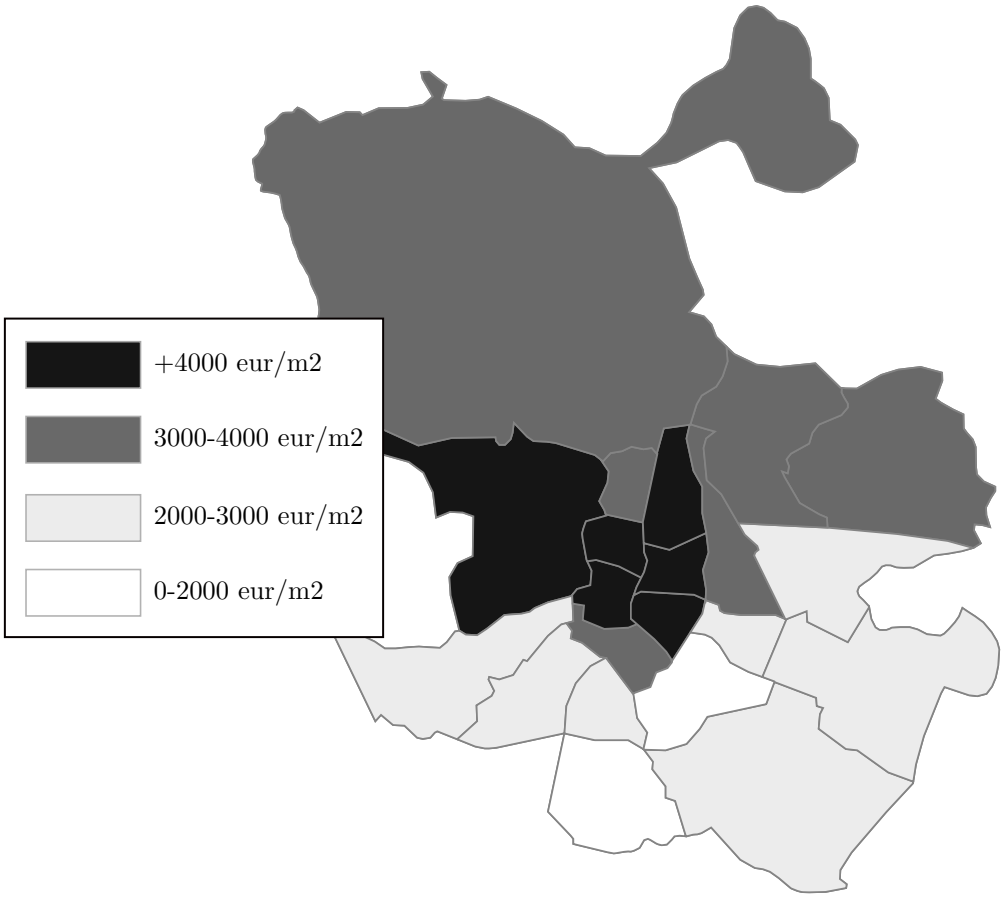

Figure 1: Average price per square meters for each district in Madrid.

[7]All of the remaining values can be accessed from the provided Jupyter notebook.

3 Conclusion

The provided dataset covers most of the currently available offers for houses and flats in Madrid provided by the three websites targeted for this project (Habitaclia, Pisos and Tucasa), for a total of around 30.000 entries.

This dataset can be practically used as shown in section 2, given some filtering of the values is realized. It can also be extended using additional websites/aggregators, and APIs.