Housing based on venue preferences and cost of land

Saul Martinez

Introduction

Problematic

Buying a house is one of the big financial decision one person does in his life, in this decision is very difficult to be impartial due to the emotional implications persons have at the moment to choose a neighborhood and a house/apartment.

To have a more logistic approximation to the decision of where to buy a house, in this study I want to propose an initial approximation that takes into account the venue's preferences of a person, and the average land prices of each neighborhood in a city.

Scope

In this evaluation, the objective will be to produce personalized results based on preferences of a given person that is looking where to live in Madrid, this means accomplish a set of factors to determine the best locations, and the price per square foot on each neighborhood.

Data Requirements

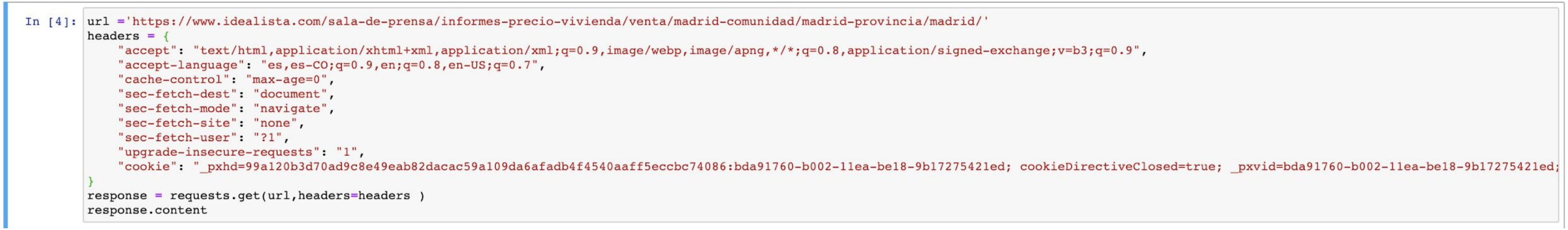
To create the model the information will be divided in five different datasets:

- Prices of the square foot on each neighborhood.
- Geographical location data of each neighborhood.
- Geographical information to establish boundaries between each neighborhood
- Venue's information of each neighborhood
- User preferences

Methodology

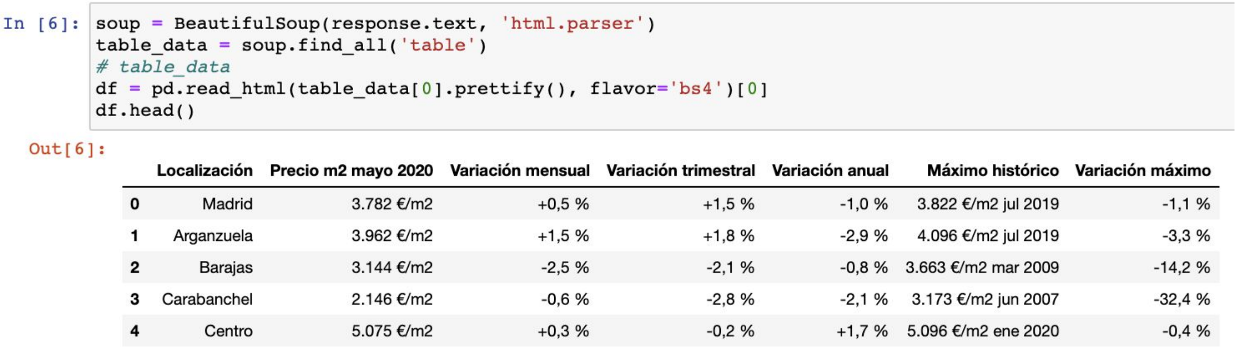
Data Acquisition

Request to Idealista.com webpage



```
In [4]: url ='https://www.idealista.com/sala-de-prensa/informes-precio-vivienda/venta/madrid-comunidad/madrid-provincia/madrid/'
        headers = {
            "accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9",
            "accept-language": "es,es-CO;q=0.9,en;q=0.8,en-US;q=0.7",
            "cache-control": "max-age=0",
            "sec-fetch-dest": "document",
            "sec-fetch-mode": "navigate",
            "sec-fetch-site": "none",
            "sec-fetch-user": "?1",
            "upgrade-insecure-requests": "1",
            "cookie": "_pxhd=99a120b3d70ad9c8e49eab82dacac59a109da6afadb4f4540aaff5eccbc74086:bda91760-b002-11ea-be18-9b17275421ed; cookieDirectiveClosed=true; _pxvid=bda91760-b002-11ea-be18-9b17275421ed;
        }
        response = requests.get(url,headers=headers )
        response.content
```

Scrape with bs4 to get the data into dataset.



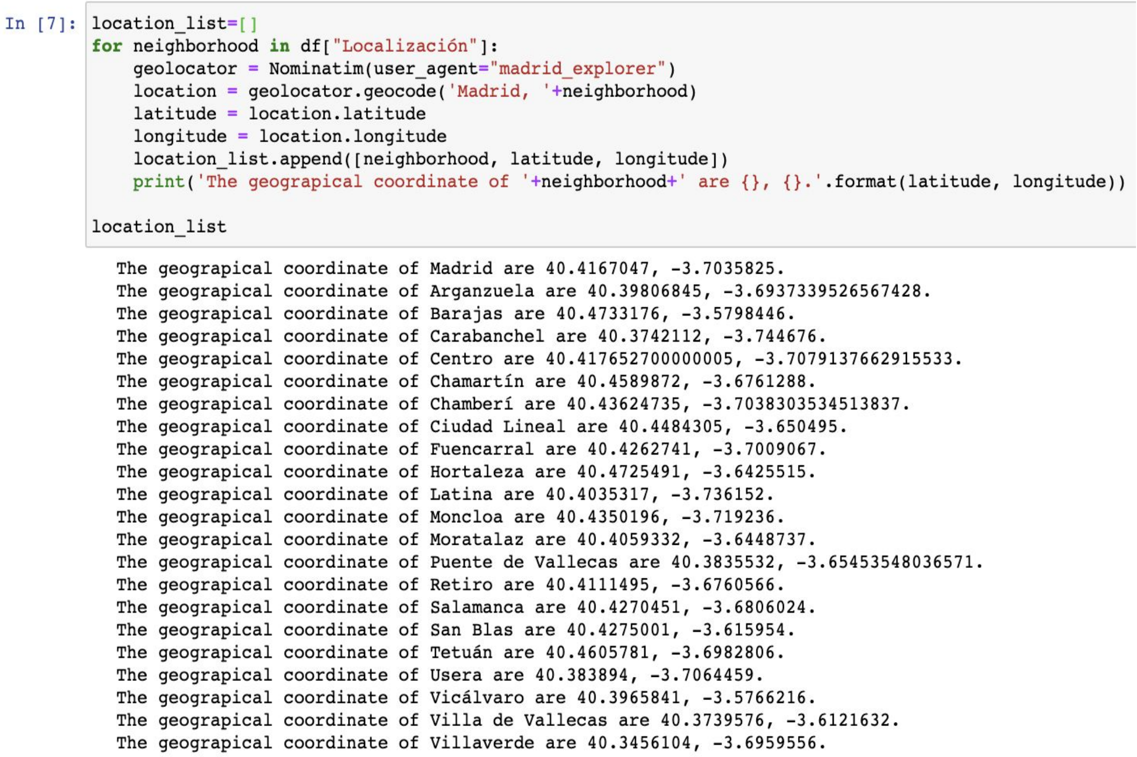
```
In [6]: soup = BeautifulSoup(response.text, 'html.parser')
        table_data = soup.find_all('table')
        # table_data
        df = pd.read_html(table_data[0].prettify(), flavor='bs4')[0]
        df.head()
```

Out[6]:

	Localización	Precio m2 mayo 2020	Variación mensual	Variación trimestral	Variación anual	Máximo histórico	Variación máximo
0	Madrid	3.782 €/m2	+0,5 %	+1,5 %	-1,0 %	3.822 €/m2 jul 2019	-1,1 %
1	Arganzuela	3.962 €/m2	+1,5 %	+1,8 %	-2,9 %	4.096 €/m2 jul 2019	-3,3 %
2	Barajas	3.144 €/m2	-2,5 %	-2,1 %	-0,8 %	3.663 €/m2 mar 2009	-14,2 %
3	Carabanchel	2.146 €/m2	-0,6 %	-2,8 %	-2,1 %	3.173 €/m2 jun 2007	-32,4 %
4	Centro	5.075 €/m2	+0,3 %	-0,2 %	+1,7 %	5.096 €/m2 ene 2020	-0,4 %

Data Acquisition

Geographical data to get information about each Neighborhood.

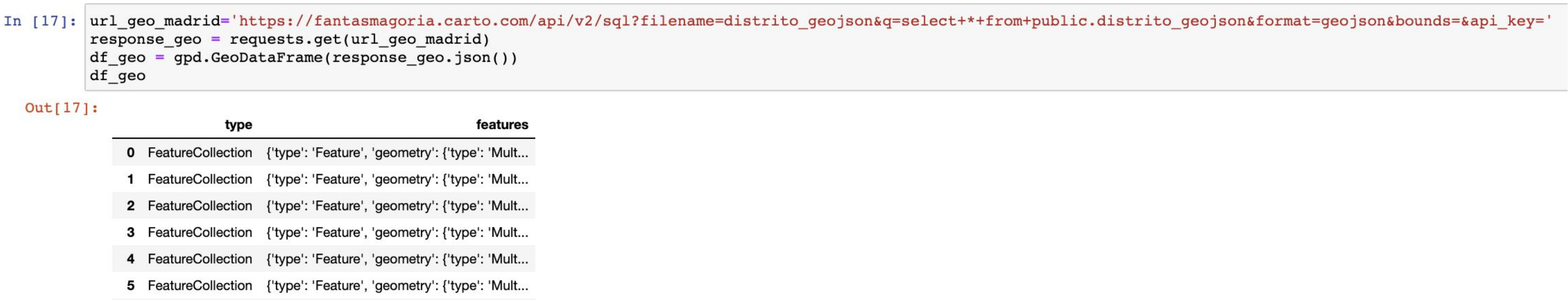


```
In [7]: location_list=[]
        for neighborhood in df["Localización"]:
            geolocator = Nominatim(user_agent="madrid_explorer")
            location = geolocator.geocode('Madrid, '+neighborhood)
            latitude = location.latitude
            longitude = location.longitude
            location_list.append([neighborhood, latitude, longitude])
            print('The geographical coordinate of '+neighborhood+' are {}, {}.'.format(latitude, longitude))

        location_list
```

```
The geographical coordinate of Madrid are 40.4167047, -3.7035825.
The geographical coordinate of Arganzuela are 40.39806845, -3.6937339526567428.
The geographical coordinate of Barajas are 40.4733176, -3.5798446.
The geographical coordinate of Carabanchel are 40.3742112, -3.744676.
The geographical coordinate of Centro are 40.417652700000005, -3.7079137662915533.
The geographical coordinate of Chamartín are 40.4589872, -3.6761288.
The geographical coordinate of Chamberí are 40.43624735, -3.7038303534513837.
The geographical coordinate of Ciudad Lineal are 40.4484305, -3.650495.
The geographical coordinate of Fuencarral are 40.4262741, -3.7009067.
The geographical coordinate of Hortaleza are 40.4725491, -3.6425515.
The geographical coordinate of Latina are 40.4035317, -3.736152.
The geographical coordinate of Moncloa are 40.4350196, -3.719236.
The geographical coordinate of Moratalaz are 40.4059332, -3.6448737.
The geographical coordinate of Puente de Vallecas are 40.3835532, -3.65453548036571.
The geographical coordinate of Retiro are 40.4111495, -3.6760566.
The geographical coordinate of Salamanca are 40.4270451, -3.6806024.
The geographical coordinate of San Blas are 40.4275001, -3.615954.
The geographical coordinate of Tetuán are 40.4605781, -3.6982806.
The geographical coordinate of Usera are 40.383894, -3.7064459.
The geographical coordinate of Vicálvaro are 40.3965841, -3.5766216.
The geographical coordinate of Villa de Vallecas are 40.3739576, -3.6121632.
The geographical coordinate of Villaverde are 40.3456104, -3.6959556.
```

Data Acquisition

Geographical boundaries of each neighborhood using polygons and geopandas



In [17]:

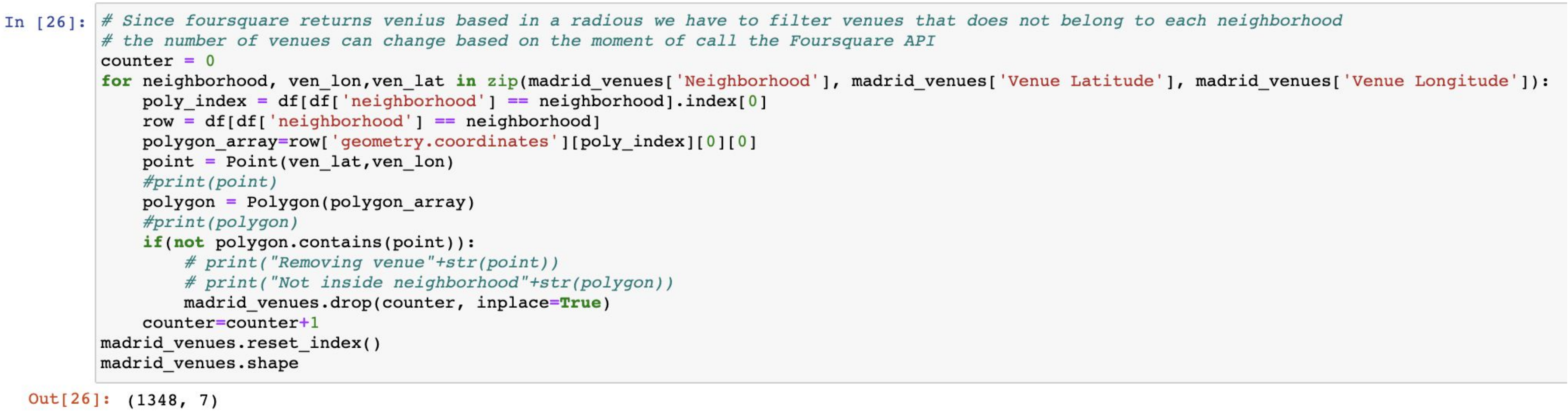
```
url_geo_madrid='https://fantasmagoria.carto.com/api/v2/sql?filename=distrito_geojson&q=select+*+from+public.distrito_geojson&format=geojson&bounds=&api_key='
response_geo = requests.get(url_geo_madrid)
df_geo = gpd.GeoDataFrame(response_geo.json())
df_geo
```

Out[17]:

	type	features
0	FeatureCollection	{'type': 'Feature', 'geometry': {'type': 'Mult...
1	FeatureCollection	{'type': 'Feature', 'geometry': {'type': 'Mult...
2	FeatureCollection	{'type': 'Feature', 'geometry': {'type': 'Mult...
3	FeatureCollection	{'type': 'Feature', 'geometry': {'type': 'Mult...
4	FeatureCollection	{'type': 'Feature', 'geometry': {'type': 'Mult...
5	FeatureCollection	{'type': 'Feature', 'geometry': {'type': 'Mult...

Data Acquisition

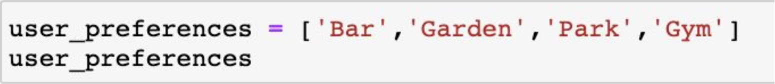
Using Foursquare API get the venues per neighborhood



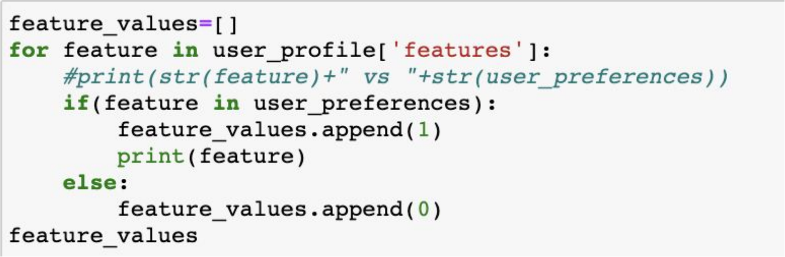
```
In [26]: # Since foursquare returns venius based in a radious we have to filter venues that does not belong to each neighborhood
         # the number of venues can change based on the moment of call the Foursquare API
         counter = 0
         for neighborhood, ven_lon,ven_lat in zip(madrid_venues['Neighborhood'], madrid_venues['Venue Latitude'], madrid_venues['Venue Longitude']):
             poly_index = df[df['neighborhood'] == neighborhood].index[0]
             row = df[df['neighborhood'] == neighborhood]
             polygon_array=row['geometry.coordinates'][poly_index][0][0]
             point = Point(ven_lat,ven_lon)
             #print(point)
             polygon = Polygon(polygon_array)
             #print(polygon)
             if(not polygon.contains(point)):
                 # print("Removing venue"+str(point))
                 # print("Not inside neighborhood"+str(polygon))
                 madrid_venues.drop(counter, inplace=True)
             counter=counter+1
         madrid_venues.reset_index()
         madrid_venues.shape

Out[26]: (1348, 7)
```

Data Acquisition

Create an user profile



```
user_preferences = ['Bar','Garden','Park','Gym']
user_preferences
```

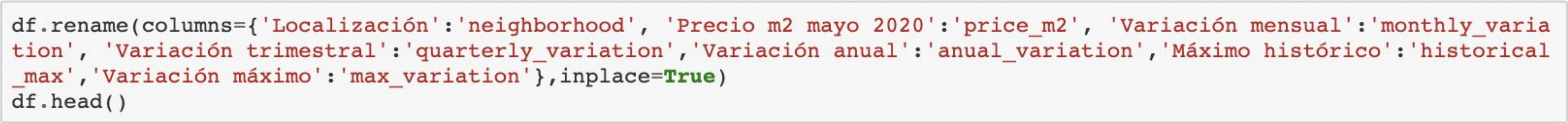


```
feature_values=[]
for feature in user_profile['features']:
    #print(str(feature)+" vs "+str(user_preferences))
    if(feature in user_preferences):
        feature_values.append(1)
        print(feature)
    else:
        feature_values.append(0)
feature_values
```

Data understanding and preparation

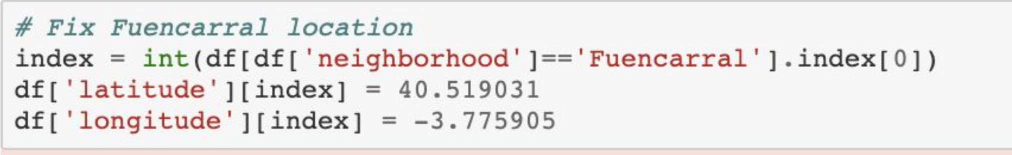
While the data was collected understanding, prepare and merge the data was essential to create the required information for the recommendation system.

Giving structure to the collected data



```
df.rename(columns={'Localización':'neighborhood', 'Precio m2 mayo 2020':'price_m2', 'Variación mensual':'monthly_variation', 'Variación trimestral':'quarterly_variation','Variación anual':'anual_variation','Máximo histórico':'historical_max','Variación máximo':'max_variation'},inplace=True)
df.head()
```

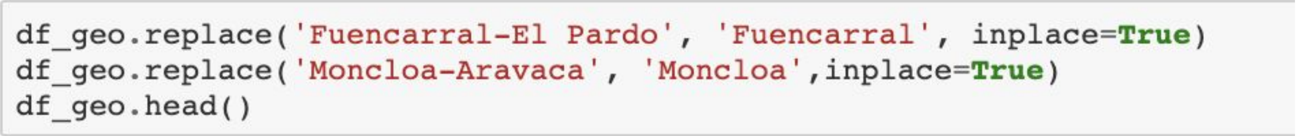
Correct information



```
# Fix Fuencarral location
index = int(df[df['neighborhood']=='Fuencarral'].index[0])
df['latitude'][index] = 40.519031
df['longitude'][index] = -3.775905
```

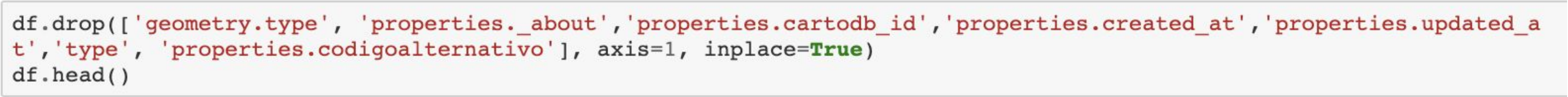
Data understanding and preparation

Align the data into the same features



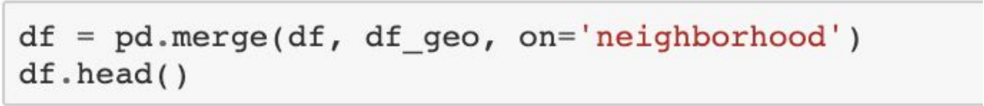
```
df_geo.replace('Fuencarral-El Pardo', 'Fuencarral', inplace=True)
df_geo.replace('Moncloa-Aravaca', 'Moncloa',inplace=True)
df_geo.head()
```

Delete Unnecessary data



```
df.drop(['geometry.type', 'properties._about','properties.cartodb_id','properties.created_at','properties.updated_a
t','type', 'properties.codigoalternativo'], axis=1, inplace=True)
df.head()
```

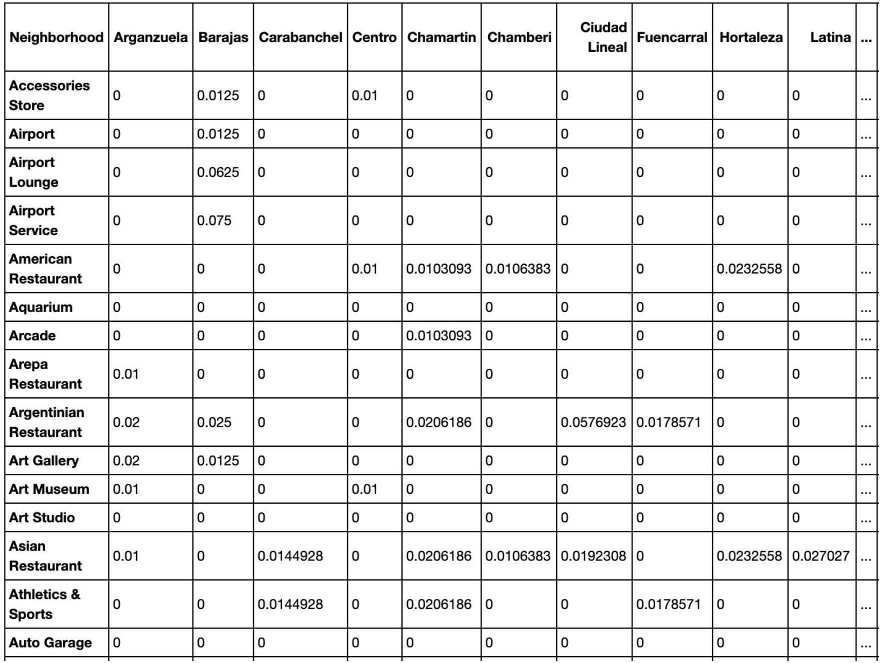
Merging the data into a unique dataset



```
df = pd.merge(df, df_geo, on='neighborhood')
df.head()
```

Creating the model

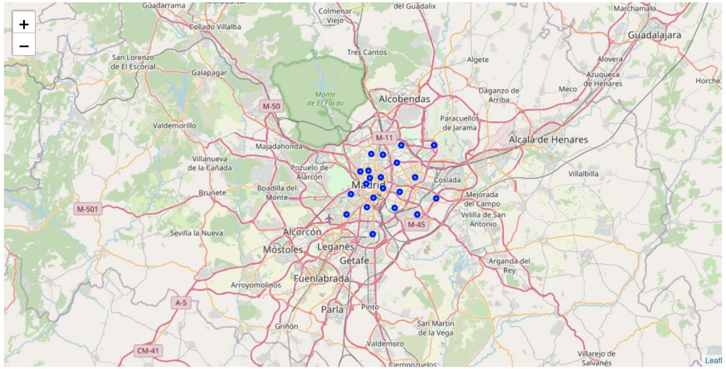
Creating the recommendation using the collected data and obtain a result to the recommendation per neighborhood based on the user preferences



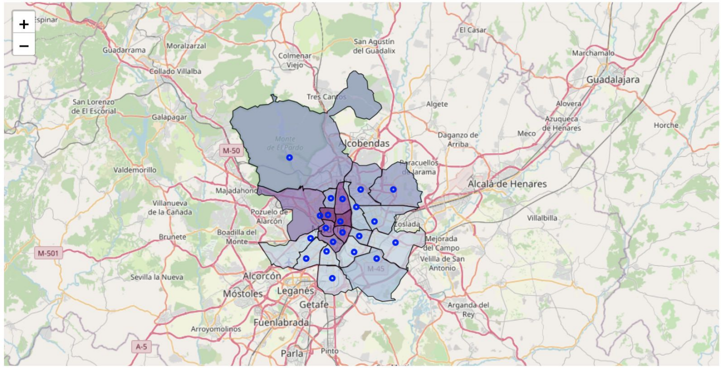
Neighborhood	Arganzuela	Barajas	Carabanchel	Centro	Chamartin	Chamberi	Ciudad Lineal	Fuencarral	Hortaleza	Latina	...
Accessories Store	0	0.0125	0	0.01	0	0	0	0	0	0	...
Airport	0	0.0125	0	0	0	0	0	0	0	0	...
Airport Lounge	0	0.0625	0	0	0	0	0	0	0	0	...
Airport Service	0	0.075	0	0	0	0	0	0	0	0	...
American Restaurant	0	0	0	0.01	0.0103093	0.0106383	0	0	0.0232558	0	...
Aquarium	0	0	0	0	0	0	0	0	0	0	...
Arcade	0	0	0	0	0.0103093	0	0	0	0	0	...
Arepa Restaurant	0.01	0	0	0	0	0	0	0	0	0	...
Argentinian Restaurant	0.02	0.025	0	0	0.0206186	0	0.0576923	0.0178571	0	0	...
Art Gallery	0.02	0.0125	0	0	0	0	0	0	0	0	...
Art Museum	0.01	0	0	0.01	0	0	0	0	0	0	...
Art Studio	0	0	0	0	0	0	0	0	0	0	...
Asian Restaurant	0.01	0	0.0144928	0	0.0206186	0.0106383	0.0192308	0	0.0232558	0.027027	...
Athletics & Sports	0	0	0.0144928	0	0.0206186	0	0	0.0178571	0	0	...
Auto Garage	0	0	0	0	0	0	0	0	0	0	...

Visualization of the result

Creating visualization on each step of the process using a choropleth map and adding layer to show the different data.

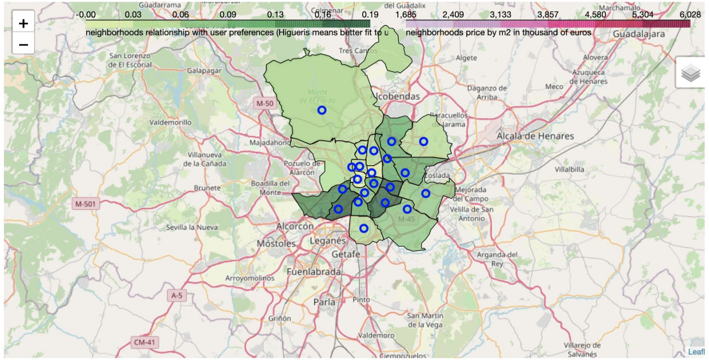


Neighborhoods



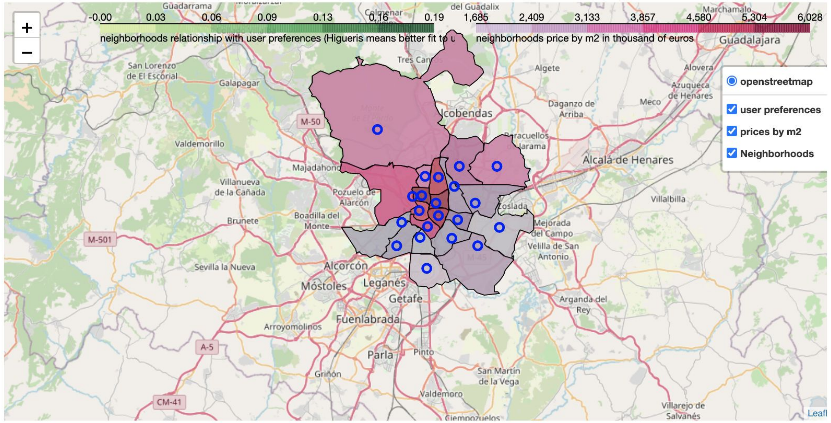
Boundaries and price per m2

Visualization of the result



-0.00 0.03 0.06 0.09 0.13 0.16 0.19
neighborhoods relationship with user preferences (Higueris means better fit to u
1,685 2,409 3,133 3,857 4,580 5,304 6,028
neighborhoods price by m2 in thousand of euros

Recommendation system result



-0.00 0.03 0.06 0.09 0.13 0.16 0.19
neighborhoods relationship with user preferences (Higueris means better fit to u
1,685 2,409 3,133 3,857 4,580 5,304 6,028
neighborhoods price by m2 in thousand of euros

● openstreetmap
☑ user preferences
☑ prices by m2
☑ Neighborhoods

Final result

Results

Result

1. The price of the square meter is higher in the center and north parts of the city.
2. This type of recommendation system produces a broad result so it helps the user to make a more rational decision, but there are still factors that can affect the user decisions.
3. In this case, some of the neighborhoods that cover most of the user preferences are not the most expensive in the city, so the preferences of the user will affect how much a user should pay for a place to live.
4. In this case Mortaraz, Latina, Carabanchel, puente de Vallecas cover most of the preferences of the user at lower prices by square meter.
5. Mortaraz is the neighborhood that covers all the user preferences.

Conclusions

Is very important always have in mind what is your goal from the start of the analysis and focus on getting the necessary data to answer the initial question because is very easy to get distracted with the additional result the data can produce and loss the focus on what was really the initial objective. This project gives a structured vision of how a problem can be approached solving step by step and how to obtain the necessary data to respond to the proposed question. The next steps for this analysis will be add new important perspectives at the moment to look for a house and try to obtain additional information of the type of property the user is looking for. I hope this analysis will be interest for other persons and I be open to any commentary.