PracticaAirBnbAntoniSbertCañellas

February 17, 2021

1 Práctica final AIRBNB

Antoni Sbert Cañellas

Link repositorio github: https://github.com/mangunillos/practivaAirBNB

Una vez que se han visualizado los datos del csv y vemos que tiene header, se procede a cargar estos en un dataframe de pandas para poder analizar los datos

[1]:
```
%matplotlib inline
import csv
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_style('darkgrid')
sns.set_context('notebook')

airbnbdata = []
with open('airbnb.csv', encoding='utf-8', newline='') as csvDataFile:
    reader = csv.reader(csvDataFile)
    for row in reader:
        airbnbdata.append(row)

df = pd.read_csv('airbnb.csv')

sns.set_style('darkgrid')
sns.set_context('notebook')
```

Mostramos la informació del dataset para ver que datos són interessantes para la solución que queremos proponer

[2]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17608 entries, 0 to 17607
Data columns (total 74 columns):
id                                              17608 non-null int64
listing_url                                     17608 non-null object
scrape_id                                       17608 non-null int64
```

```
last_scraped                                  17608 non-null object
name                                          17607 non-null object
description                                   17393 non-null object
neighborhood_overview                         8213 non-null object
picture_url                                   17608 non-null object
host_id                                       17608 non-null int64
host_url                                      17608 non-null object
host_name                                     17606 non-null object
host_since                                    17606 non-null object
host_location                                 17572 non-null object
host_about                                    11696 non-null object
host_response_time                            15862 non-null object
host_response_rate                            15862 non-null object
host_acceptance_rate                          16098 non-null object
host_is_superhost                             17606 non-null object
host_thumbnail_url                            17606 non-null object
host_picture_url                              17606 non-null object
host_neighbourhood                            364 non-null object
host_listings_count                           17606 non-null float64
host_total_listings_count                     17606 non-null float64
host_verifications                            17608 non-null object
host_has_profile_pic                          17606 non-null object
host_identity_verified                        17606 non-null object
neighbourhood                                 8213 non-null object
neighbourhood_cleansed                        17608 non-null object
neighbourhood_group_cleansed                  0 non-null float64
latitude                                      17608 non-null float64
longitude                                     17608 non-null float64
property_type                                 17608 non-null object
room_type                                     17608 non-null object
accommodates                                  17608 non-null int64
bathrooms                                     0 non-null float64
bathrooms_text                                17600 non-null object
bedrooms                                      17333 non-null float64
beds                                          17511 non-null float64
amenities                                     17608 non-null object
price                                         17608 non-null object
minimum_nights                                17608 non-null int64
maximum_nights                                17608 non-null int64
minimum_minimum_nights                        17608 non-null int64
maximum_minimum_nights                        17608 non-null int64
minimum_maximum_nights                        17608 non-null int64
maximum_maximum_nights                        17608 non-null int64
minimum_nights_avg_ntm                        17608 non-null float64
maximum_nights_avg_ntm                        17608 non-null float64
calendar_updated                              0 non-null float64
has_availability                              17608 non-null object
availability_30                               17608 non-null int64
```

```
availability_60                                 17608 non-null int64
availability_90                                 17608 non-null int64
availability_365                                17608 non-null int64
calendar_last_scraped                           17608 non-null object
number_of_reviews                               17608 non-null int64
number_of_reviews_ltm                           17608 non-null int64
number_of_reviews_l30d                          17608 non-null int64
first_review                                    11173 non-null object
last_review                                     11173 non-null object
review_scores_rating                            10957 non-null float64
review_scores_accuracy                          10951 non-null float64
review_scores_cleanliness                       10953 non-null float64
review_scores_checkin                           10949 non-null float64
review_scores_communication                     10951 non-null float64
review_scores_location                          10950 non-null float64
review_scores_value                             10949 non-null float64
license                                         11431 non-null object
instant_bookable                                17608 non-null object
calculated_host_listings_count                  17608 non-null int64
calculated_host_listings_count_entire_homes     17608 non-null int64
calculated_host_listings_count_private_rooms    17608 non-null int64
calculated_host_listings_count_shared_rooms     17608 non-null int64
reviews_per_month                               11173 non-null float64
dtypes: float64(19), int64(21), object(34)
memory usage: 9.9+ MB
```

A primera vista vemos que hay una gran cantidad de datos sobre las diferentes viviendas. Aún asi vemos que segurament hay algunos datos que no podremos aprovechar como por ejemplo la url de la foto.

2 Seleccion de datos relevantes

Empezamos con el análisis de los datos de scraping, estos parametros una vez cogidos y visualizados de esta manera, nos encontramos que para el objetivo final, predecir el precio, no són relevantes para nosotros.

[3]:
```
df[['id',␣
 ↪'listing_url','scrape_id','last_scraped','name','description','neighborhood_overview','pict
```

[3]:

	id	listing_url	scrape_id	\
0	11547	https://www.airbnb.com/rooms/11547	20200919153121	
1	100831	https://www.airbnb.com/rooms/100831	20200919153121	
2	105891	https://www.airbnb.com/rooms/105891	20200919153121	
3	106833	https://www.airbnb.com/rooms/106833	20200919153121	
4	130669	https://www.airbnb.com/rooms/130669	20200919153121	
…	…	…	…	
17603	45489412	https://www.airbnb.com/rooms/45489412	20200919153121	
17604	45489550	https://www.airbnb.com/rooms/45489550	20200919153121	

```
17605  45493152  https://www.airbnb.com/rooms/45493152  20200919153121
17606  45496032  https://www.airbnb.com/rooms/45496032  20200919153121
17607  45499210  https://www.airbnb.com/rooms/45499210  20200919153121

      last_scraped                                              name  \
0       2020-09-21                                My home at the beach
1       2020-09-21                     HOUSE IN MALLORCA - WiFi(ET-3045)
2       2020-09-20           VILLAGE HOUSE WITH POOL: IDEAL FOR FAMILIES
3       2020-09-20                     Villa with a big pool in Mallorca
4       2020-09-20                                Room great apartment
…              …                                                    …
17603   2020-09-20                                       Villa Mariona
17604   2020-09-20  Holiday Home El Clavet with Mountain View, Wi-…
17605   2020-09-20                  Es Molinet Selva villa Mallorca 197
17606   2020-09-21  Hab. con terraza y baño privado,piscina compar…
17607   2020-09-20                       Bonita Casa Adosada con Jardín

                                              description  \
0      Sun, joy, relax, quality, beach & peace.<br />…
1      <b>The space</b><br />House situated in a quie…
2      The house is a street on the outskirts of the …
3      <b>The space</b><br />This is a restored old b…
4      Located in a residential neighbourhood and 10m…
…                                                    …
17603  Modern and comfortable villa for 6 people comp…
17604  Located between Pollença and Port de Pollença,…
17605  Magnificent finca with pool, garden and great …
17606                                                NaN
17607                                                NaN

                                  neighborhood_overview  \
0                                                    NaN
1                                                    NaN
2      The village's population does not reach two th…
3                                                    NaN
4      Located in the center of the city, within minu…
…                                                    …
17603                                                NaN
17604                                                NaN
17605                                                NaN
17606  Habitación doble a la entrada del puerto muy c…
17607                                                NaN

                                              picture_url
0      https://a0.muscache.com/pictures/494126/8c151b…
1      https://a0.muscache.com/pictures/675527/72b329…
2      https://a0.muscache.com/pictures/1036816/f36ce…
```

```
3      https://a0.muscache.com/pictures/710218/98134c…
4      https://a0.muscache.com/pictures/866653/58dc48…
…                                                    …
17603  https://a0.muscache.com/pictures/105faa9b-5315…
17604  https://a0.muscache.com/pictures/f7e347f8-894c…
17605  https://a0.muscache.com/pictures/2473f78b-359e…
17606  https://a0.muscache.com/pictures/2dd9c712-7b01…
17607  https://a0.muscache.com/pictures/c2f63ec4-fce6…

[17608 rows x 8 columns]
```

2.1 Datos del host

Una vez visualizados los datos del host, nos encontramos que estos parámetros no los podemos usar tampoco ya que no nos sirve para orientar el precio de la vivienda por sus características, quizas estos datos podrían llegar a ser relevantes para otro tipo de situaciones donde cobrarian más importancia

[4]:
```
df[['host_id','host_url','host_name','host_since','host_location','host_about','host_response_
 ↪'host_acceptance_rate','host_is_superhost','host_thumbnail_url', ␣
 ↪'host_picture_url','host_neighbourhood', 'host_listings_count',␣
 ↪'host_total_listings_count','host_verifications','host_has_profile_pic',␣
 ↪'host_identity_verified']]
```

[4]:
```
          host_id                                      host_url  \
0           42942      https://www.airbnb.com/users/show/42942
1          529151     https://www.airbnb.com/users/show/529151
2          549192     https://www.airbnb.com/users/show/549192
3          551974     https://www.airbnb.com/users/show/551974
4          643065     https://www.airbnb.com/users/show/643065
…             …                                             …
17603  131594018  https://www.airbnb.com/users/show/131594018
17604  285670200  https://www.airbnb.com/users/show/285670200
17605    3893191    https://www.airbnb.com/users/show/3893191
17606  350176105  https://www.airbnb.com/users/show/350176105
17607   67725432   https://www.airbnb.com/users/show/67725432

                                 host_name  host_since  \
0                                   Daniel  2009-10-02
1                                   Miguel  2011-04-23
2                                 Bartomeu  2011-05-01
3                                    Xisco  2011-05-02
4                                     Nick  2011-05-30
…                                        …           …
17603          Mallorca Holiday Properties  2017-05-23
17604                              Bookiply  2019-08-14
17605  Antoni Mallorca Charme Home Rentals  2012-10-16
```

```
17606                              Gabriele  2020-06-15
17607                                  Juan  2016-04-18

                                          host_location  \
0                                Balearic Islands, Spain
1                                                Mallorca
2                         Ariany, Balearic Islands, Spain
3               Palma de Mallorca, Balearic Islands, Spain
4               Palma de Mallorca, Balearic Islands, Spain
…                                                      …
17603  Av. Príncipes de España 4A,  Alcudia, 07400 Is…
17604                                                 DE
17605                           Muro, Illes Balears, Spain
17606                Port d'Andratx, Illes Balears, Spain
17607                                                 ES

                                             host_about  host_response_time  \
0                                                     .      within an hour
1      Somos una pareja con los mismos gustos e inter…                 NaN
2      Hola!. Resido en una casa de campo de un puebl…  within a few hours
3               I'm Xisco. I love Mallorcan way of life.        within a day
4                                                   NaN        within a day
…                                                     …                  …
17603  Mallorca Holiday Properties está formado por u…      within an hour
17604  Bookiply is a German agency offering vacation …  within a few hours
17605                                               NaN        within a day
17606                                               NaN                 NaN
17607                                               NaN      within an hour

      host_response_rate host_acceptance_rate host_is_superhost  \
0                   100%                  96%                 f
1                    NaN                 100%                 f
2                   100%                  83%                 t
3                   100%                  NaN                 f
4                   100%                  NaN                 f
…                      …                    …                 …
17603               100%                 100%                 f
17604                87%                 100%                 f
17605                94%                  97%                 f
17606                NaN                  NaN                 f
17607               100%                  NaN                 f

                                    host_thumbnail_url  \
0      https://a0.muscache.com/im/users/42942/profile…
1      https://a0.muscache.com/im/users/529151/profil…
2      https://a0.muscache.com/im/users/549192/profil…
3      https://a0.muscache.com/im/users/551974/profil…
```

```
4      https://a0.muscache.com/im/users/643065/profil…
…                                                    …
17603  https://a0.muscache.com/im/pictures/user/dce1d…
17604  https://a0.muscache.com/im/pictures/user/8cfab…
17605  https://a0.muscache.com/im/pictures/user/7dca3…
17606  https://a0.muscache.com/im/pictures/user/88dc0…
17607  https://a0.muscache.com/im/pictures/user/40053…

                                    host_picture_url host_neighbourhood  \
0      https://a0.muscache.com/im/users/42942/profile…                NaN
1      https://a0.muscache.com/im/users/529151/profil…                NaN
2      https://a0.muscache.com/im/users/549192/profil…                NaN
3      https://a0.muscache.com/im/users/551974/profil…                NaN
4      https://a0.muscache.com/im/users/643065/profil…                NaN
…                                                    …                …
17603  https://a0.muscache.com/im/pictures/user/dce1d…                NaN
17604  https://a0.muscache.com/im/pictures/user/8cfab…                NaN
17605  https://a0.muscache.com/im/pictures/user/7dca3…                NaN
17606  https://a0.muscache.com/im/pictures/user/88dc0…                NaN
17607  https://a0.muscache.com/im/pictures/user/40053…                NaN

       host_listings_count  host_total_listings_count  \
0                      0.0                        0.0
1                      1.0                        1.0
2                      2.0                        2.0
3                      1.0                        1.0
4                      3.0                        3.0
…                      …                          …
17603                 56.0                       56.0
17604                243.0                      243.0
17605                 98.0                       98.0
17606                  1.0                        1.0
17607                  2.0                        2.0

                                   host_verifications host_has_profile_pic  \
0      ['email', 'phone', 'reviews', 'jumio', 'govern…                    t
1      ['email', 'phone', 'facebook', 'reviews', 'jum…                    t
2      ['email', 'phone', 'facebook', 'reviews', 'jum…                    t
3      ['email', 'phone', 'reviews', 'jumio', 'offlin…                    t
4                          ['email', 'phone', 'reviews']                    t
…                                                    …                …
17603  ['email', 'phone', 'jumio', 'offline_governmen…                    t
17604  ['email', 'phone', 'jumio', 'offline_governmen…                    t
17605  ['email', 'phone', 'reviews', 'offline_governm…                    t
17606                                ['email', 'phone']                    t
17607                                ['email', 'phone']                    t
```

```
      host_identity_verified
0                          t
1                          t
2                          t
3                          t
4                          f
…                        …
17603                      t
17604                      t
17605                      t
17606                      f
17607                      f

[17608 rows x 18 columns]
```

2.2 Datos del vecindario.

Nos encontramos con más datos que no podemos usar para indicar cual es el precio del los apartamentos de manera objetiva y no podemos aprovechar esta información para obtener los precios directamente. Además que muchos campos de las columns al ser Nan nos faltaria información

[5]:
```
df[['neighbourhood','neighbourhood_cleansed','neighbourhood_group_cleansed']]
```

[5]:
```
                                  neighbourhood      neighbourhood_cleansed  \
0                                           NaN                      Calvià
1                                           NaN           Santa Margalida
2      Maria de la Salut, Balearic Islands, Spain         Maria de la Salut
3                                           NaN  Sant Llorenç des Cardassar
4                   Palma de Mallorca, PM, Spain         Palma de Mallorca
…                                             …                         …
17603                                       NaN                      Muro
17604                                       NaN                   Pollença
17605                                       NaN                     Selva
17606              Andratx, Illes Balears, Spain                   Andratx
17607                                       NaN                  Llucmajor

       neighbourhood_group_cleansed
0                               NaN
1                               NaN
2                               NaN
3                               NaN
4                               NaN
…                               …
17603                           NaN
17604                           NaN
17605                           NaN
17606                           NaN
```

```
17607                          NaN

[17608 rows x 3 columns]
```

2.3 Datos sobre las características de las casas

Estos datos parece que podrían servir para definir los precios de las casas aunque por ejemplo la columna bathrooms no se pordra usar y se tendria quer modificar la de bathroom_text para poder sacar valor. En estos campos la columna amenities tiene muchas información que podría llegar a ser un mini dataset y no lo usaremos para definir el precio final

[6]:
```
df[['latitude', 'longitude', 'property_type', 'room_type', 'accommodates',␣
 ↪'bathrooms', 'bathrooms_text', 'bedrooms', 'beds', 'amenities', 'price']]
```

[6]:
```
       latitude  longitude              property_type        room_type  \
0      39.51888    2.48182           Entire apartment  Entire home/apt
1      39.76347    3.16255               Entire house  Entire home/apt
2      39.66044    3.07165           Entire townhouse  Entire home/apt
3      39.61600    3.30121               Entire villa  Entire home/apt
4      39.56478    2.60333  Private room in apartment     Private room
…         …        …                          …              …
17603  39.76505    3.12689               Entire villa  Entire home/apt
17604  39.89835    3.03647               Entire villa  Entire home/apt
17605  39.75437    2.90504               Entire villa  Entire home/apt
17606  39.54550    2.39348  Private room in apartment     Private room
17607  39.44457    2.75415           Entire apartment  Entire home/apt

       accommodates  bathrooms bathrooms_text  bedrooms  beds  \
0                 2        NaN         1 bath       1.0   1.0
1                 8        NaN        3 baths       4.0   7.0
2                 6        NaN        2 baths       3.0   4.0
3                 4        NaN         1 bath       2.0   4.0
4                 2        NaN         1 bath       1.0   2.0
…               …        …             …       …   …
17603             6        NaN      3.5 baths       3.0   4.0
17604             9        NaN        3 baths       5.0   8.0
17605             6        NaN        2 baths       3.0   4.0
17606             2        NaN         1 bath       1.0   1.0
17607             6        NaN        2 baths       2.0   4.0

                                                amenities    price
0      ["Oven", "Wifi", "Coffee maker", "Dishes and s…   $89.00
1      ["First aid kit", "Hair dryer", "Iron", "Washe…  $175.00
2      ["Smoke alarm", "Oven", "Wifi", "Garden or bac…  $140.00
3      ["Pool", "Free parking on premises", "Air cond…  $200.00
4      ["Pool", "Washer", "Air conditioning", "Kitche…  $110.00
…                                                    …      …
```

```
17603  ["Oven", "Wifi", "Garden or backyard", "Coffee…  $195.00
17604  ["Hair dryer", "Washer", "Free parking on prem…  $110.00
17605  ["Oven", "Wifi", "Garden or backyard", "Coffee…  $179.00
17606  ["Oven", "Bread maker", "Extra pillows and bla…   $42.00
17607  ["First aid kit", "Hair dryer", "Iron", "Washe…  $100.00

[17608 rows x 11 columns]
```

2.4 Datos de las noches que se pueden reservar las casa

Estos datos no los podemos usar porque no nos indican directamente el precio según las características de las casa

[7]:
```
df[['minimum_nights', 'maximum_nights', 'minimum_minimum_nights',␣
 ↪'maximum_minimum_nights','minimum_maximum_nights','maximum_maximum_nights','minimum_nights_
```

[7]:
```
       minimum_nights  maximum_nights  minimum_minimum_nights  \
0                   5              60                       5
1                   7             365                       7
2                   6             365                       6
3                   5             365                       5
4                   2             365                       2
…                 …               …                       …
17603               1             365                       6
17604               1            1125                       1
17605               1             365                       1
17606               1               7                       1
17607               5            1120                       5

       maximum_minimum_nights  minimum_maximum_nights  maximum_maximum_nights  \
0                           5                      60                      60
1                           7                    1125                    1125
2                           6                     365                     365
3                           5                     365                     365
4                           2                     365                     365
…                         …                       …                       …
17603                       7                     365                     365
17604                       1                      28                    1125
17605                       7                     365                     365
17606                       1                    1125                    1125
17607                       5                    1120                    1120

       minimum_nights_avg_ntm  maximum_nights_avg_ntm
0                         5.0                    60.0
1                         7.0                  1125.0
2                         6.0                   365.0
3                         5.0                   365.0
```

```
4                        2.0                   365.0
…                        …                     …
17603                    6.4                   365.0
17604                    1.0                   590.4
17605                    3.3                   365.0
17606                    1.0                  1125.0
17607                    5.0                  1120.0

[17608 rows x 8 columns]
```

2.5 Datos de la disponibilidad

Según los dias y reviews, en este caso tampoco podemos usar los datos para poder determinar el precio de los immuebles

[8]:
```
df[['has_availability',
'availability_30',
'availability_60',
'availability_90',
'availability_365',
'calendar_last_scraped',
'number_of_reviews',
'number_of_reviews_ltm',
'number_of_reviews_l30d']]
```

[8]:
```
      has_availability  availability_30  availability_60  availability_90  \
0                    t               13               43               73
1                    t                0                0                0
2                    t               23               53               83
3                    t               20               50               80
4                    t               30               60               90
…                  …                 …                …                …
17603                t               10               40               70
17604                t               30               60               90
17605                t               20               20               20
17606                t               24               54               84
17607                t               30               60               90

       availability_365 calendar_last_scraped  number_of_reviews  \
0                   311            2020-09-21                103
1                     0            2020-09-21                 30
2                   325            2020-09-20                 14
3                   355            2020-09-20                  9
4                   365            2020-09-20                  0
…                 …                        …                …
17603               303            2020-09-20                  0
17604               365            2020-09-20                  0
```

```
17605                113             2020-09-20                   0
17606                174             2020-09-21                   0
17607                365             2020-09-20                   0

       number_of_reviews_ltm  number_of_reviews_l30d
0                          8                       2
1                         13                       0
2                          0                       0
3                          0                       0
4                          0                       0
…                        …                       …
17603                      0                       0
17604                      0                       0
17605                      0                       0
17606                      0                       0
17607                      0                       0

[17608 rows x 9 columns]
```

2.6 Número de reviews

Según las reviews, no podemos estimar el precio de la vivienda estos datos podrían se mejores para poder definir si hay una mayor satisfación del cliente más que del precio de la vivienda

[9]:
```
df[['number_of_reviews_l30d',
'first_review',
'last_review',
'review_scores_rating',
'review_scores_accuracy',
'review_scores_cleanliness',
'review_scores_checkin',
'review_scores_communication',
'review_scores_location',
'review_scores_value',
'license',
'instant_bookable',
'calculated_host_listings_count',
'calculated_host_listings_count_entire_homes',
'calculated_host_listings_count_private_rooms',
'calculated_host_listings_count_shared_rooms',
'reviews_per_month']]
```

[9]:
```
       number_of_reviews_l30d first_review last_review  review_scores_rating  \
0                           2   2011-08-23  2020-09-06                 96.0
1                           0   2019-01-18  2020-01-25                100.0
2                           0   2012-06-19  2019-09-03                 97.0
3                           0   2012-06-05  2018-08-13                 98.0
```

```
4                           0          NaN         NaN                    NaN
…                         …          …         …                      …
17603                       0          NaN         NaN                    NaN
17604                       0          NaN         NaN                    NaN
17605                       0          NaN         NaN                    NaN
17606                       0          NaN         NaN                    NaN
17607                       0          NaN         NaN                    NaN

       review_scores_accuracy  review_scores_cleanliness  \
0                        10.0                        9.0
1                        10.0                       10.0
2                        10.0                       10.0
3                        10.0                       10.0
4                         NaN                        NaN
…                         …                          …
17603                     NaN                        NaN
17604                     NaN                        NaN
17605                     NaN                        NaN
17606                     NaN                        NaN
17607                     NaN                        NaN

       review_scores_checkin  review_scores_communication  \
0                       10.0                        10.0
1                       10.0                        10.0
2                       10.0                        10.0
3                       10.0                        10.0
4                        NaN                         NaN
…                        …                           …
17603                    NaN                         NaN
17604                    NaN                         NaN
17605                    NaN                         NaN
17606                    NaN                         NaN
17607                    NaN                         NaN

       review_scores_location  review_scores_value      license  \
0                        10.0                 10.0          NaN
1                        10.0                 10.0     ETV-3045
2                         9.0                 10.0     ETV/6127
3                         9.0                  9.0      ET/1961
4                         NaN                  NaN          NaN
…                         …                    …          …
17603                     NaN                  NaN  LIZE84/2017
17604                     NaN                  NaN          NaN
17605                     NaN                  NaN     00551ETV
17606                     NaN                  NaN          NaN
17607                     NaN                  NaN          NaN
```

```
      instant_bookable  calculated_host_listings_count  \
0                    f                               1
1                    t                               1
2                    t                               2
3                    f                               1
4                    t                               2
…                  …                               …
17603                t                              49
17604                t                              90
17605                t                             133
17606                t                               1
17607                f                               1

       calculated_host_listings_count_entire_homes  \
0                                                1
1                                                1
2                                                2
3                                                1
4                                                0
…                                              …
17603                                           49
17604                                           90
17605                                          133
17606                                            0
17607                                            1

       calculated_host_listings_count_private_rooms  \
0                                                 0
1                                                 0
2                                                 0
3                                                 0
4                                                 2
…                                               …
17603                                             0
17604                                             0
17605                                             0
17606                                             1
17607                                             0

       calculated_host_listings_count_shared_rooms  reviews_per_month
0                                                0               0.93
1                                                0               1.47
2                                                0               0.14
3                                                0               0.09
4                                                0                NaN
…                                              …                 …
17603                                            0                NaN
```

```
17604                                                 0                NaN
17605                                                 0                NaN
17606                                                 0                NaN
17607                                                 0                NaN

[17608 rows x 17 columns]
```

3 Datos que finalmente procesaremos

Una vez que hemos analizado los datos del dataframe nos quedamos con estas columnas.

[10]:
```
df_transform = df[['latitude',␣
↪'longitude','room_type','accommodates','bathrooms',␣
↪'bedrooms','beds','price']]
```

[11]:
```
df_transform
```

[11]:
```
        latitude  longitude        room_type  accommodates  bathrooms  \
0       39.51888    2.48182  Entire home/apt             2        NaN
1       39.76347    3.16255  Entire home/apt             8        NaN
2       39.66044    3.07165  Entire home/apt             6        NaN
3       39.61600    3.30121  Entire home/apt             4        NaN
4       39.56478    2.60333     Private room             2        NaN
…          …          …                …            …         …
17603   39.76505    3.12689  Entire home/apt             6        NaN
17604   39.89835    3.03647  Entire home/apt             9        NaN
17605   39.75437    2.90504  Entire home/apt             6        NaN
17606   39.54550    2.39348     Private room             2        NaN
17607   39.44457    2.75415  Entire home/apt             6        NaN

        bedrooms  beds    price
0            1.0   1.0   $89.00
1            4.0   7.0  $175.00
2            3.0   4.0  $140.00
3            2.0   4.0  $200.00
4            1.0   2.0  $110.00
…          …   …      …
17603        3.0   4.0  $195.00
17604        5.0   8.0  $110.00
17605        3.0   4.0  $179.00
17606        1.0   1.0   $42.00
17607        2.0   4.0  $100.00

[17608 rows x 8 columns]
```

Nos encontramos que hay unas columnas que necesitan una modificación para poder ser usadas como el roomtype y modificaremos la columna bathrroms a partir de la de texto que contiene la informción

[12]:
```
#Modificamos la columna precios para que sea float
price_columns = df.price.str.split("$", expand=True)
split_price = price_columns[1].str.split(",", expand=False)
df_transform['price'] = split_price.str.join("")
df_transform['price'] = df_transform.price.astype(float)
```

```
C:\Users\Mangu\Anaconda3\lib\site-packages\ipykernel_launcher.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  after removing the cwd from sys.path.
C:\Users\Mangu\Anaconda3\lib\site-packages\ipykernel_launcher.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
```

[13]:
```
#Modificamos la información de los baños para que sea númerica, tomamos una␣
↪decisisión arbitraria de que los compartidos y medio baños se transforman en␣
↪0.5, también podríamos modificar
df_transform['bathrooms'] = df.bathrooms_text.str.split(" ", expand=True)
df_transform['bathrooms'] = df_transform['bathrooms'].replace(['Half-bath', 0.
↪5])
df_transform['bathrooms'] = df_transform['bathrooms'].replace(['Shared', 0.5])
df_transform['bathrooms'] = df_transform['bathrooms'].replace(['Private', 1])
df_transform['bathrooms'] = df_transform.bathrooms.astype(float)
```

```
C:\Users\Mangu\Anaconda3\lib\site-packages\ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

C:\Users\Mangu\Anaconda3\lib\site-packages\ipykernel_launcher.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

```
  This is separate from the ipykernel package so we can avoid doing imports
until
C:\Users\Mangu\Anaconda3\lib\site-packages\ipykernel_launcher.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  after removing the cwd from sys.path.
C:\Users\Mangu\Anaconda3\lib\site-packages\ipykernel_launcher.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  """
C:\Users\Mangu\Anaconda3\lib\site-packages\ipykernel_launcher.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[14]:
```
df_transform
```

[14]:

	latitude	longitude	room_type	accommodates	bathrooms	bedrooms	beds	price
0	39.51888	2.48182	Entire home/apt	2	1.0	1.0	1.0	89.0
1	39.76347	3.16255	Entire home/apt	8	3.0	4.0	7.0	175.0
2	39.66044	3.07165	Entire home/apt	6	2.0	3.0	4.0	140.0
3	39.61600	3.30121	Entire home/apt	4	1.0	2.0	4.0	200.0
4	39.56478	2.60333	Private room	2	1.0	1.0	1.0	79.0
…	…	…	…	…	…	…	…	…
17603	39.76505	3.12689	Entire home/apt	6	3.5	3.0	3.0	150.0
17604	39.89835	3.03647	Entire home/apt	9	3.0	4.0	5.0	150.0
17605	39.75437	2.90504	Entire home/apt	6	2.0	3.0	3.0	150.0
17606	39.54550	2.39348	Private room	2	1.0	1.0	1.0	79.0
17607	39.44457	2.75415	Entire home/apt	6	2.0	3.0	3.0	150.0

```
4           1.0   2.0  110.0
…           …     …    …
17603       3.0   4.0  195.0
17604       5.0   8.0  110.0
17605       3.0   4.0  179.0
17606       1.0   1.0   42.0
17607       2.0   4.0  100.0

[17608 rows x 8 columns]
```

Quitamos la columna de room_types para poder hacer un pairplot y ver la correlación de las variables elegidas previamentes

[15]:
```
dataFrameToPlot = pd.DataFrame(df_transform,columns=␣
↪['latitude','longitude','accommodates','bathrooms', 'bedrooms','beds',␣
↪'price'])
```

[16]:
```
dataFrameToPlot.describe()
```

[16]:
```
           latitude     longitude  accommodates     bathrooms      bedrooms  \
count  17608.000000  17608.000000  17608.000000  17600.000000  17333.000000
mean      39.657597      2.994115      5.857962      2.291477      2.952980
std        0.165577      0.246129      2.727737      1.362038      1.520433
min       39.301970      2.346500      0.000000      0.000000      1.000000
25%       39.543433      2.796892      4.000000      1.000000      2.000000
50%       39.670285      3.047310      6.000000      2.000000      3.000000
75%       39.797413      3.145215      8.000000      3.000000      4.000000
max       39.930650      3.475520     16.000000     32.000000     40.000000

                beds         price
count  17511.000000  17608.000000
mean       4.357147    244.383561
std        2.561838    409.958169
min        0.000000      0.000000
25%        3.000000    110.000000
50%        4.000000    179.000000
75%        6.000000    275.000000
max       50.000000  20736.000000
```

A primera vista podemos ver que hay algunos de los datos que no estan completos y que la desviación de los valores que de precio, existe una presencia de outliers igual que los minimos y los máximos de cantidades altas y abajs en comparación de la media.

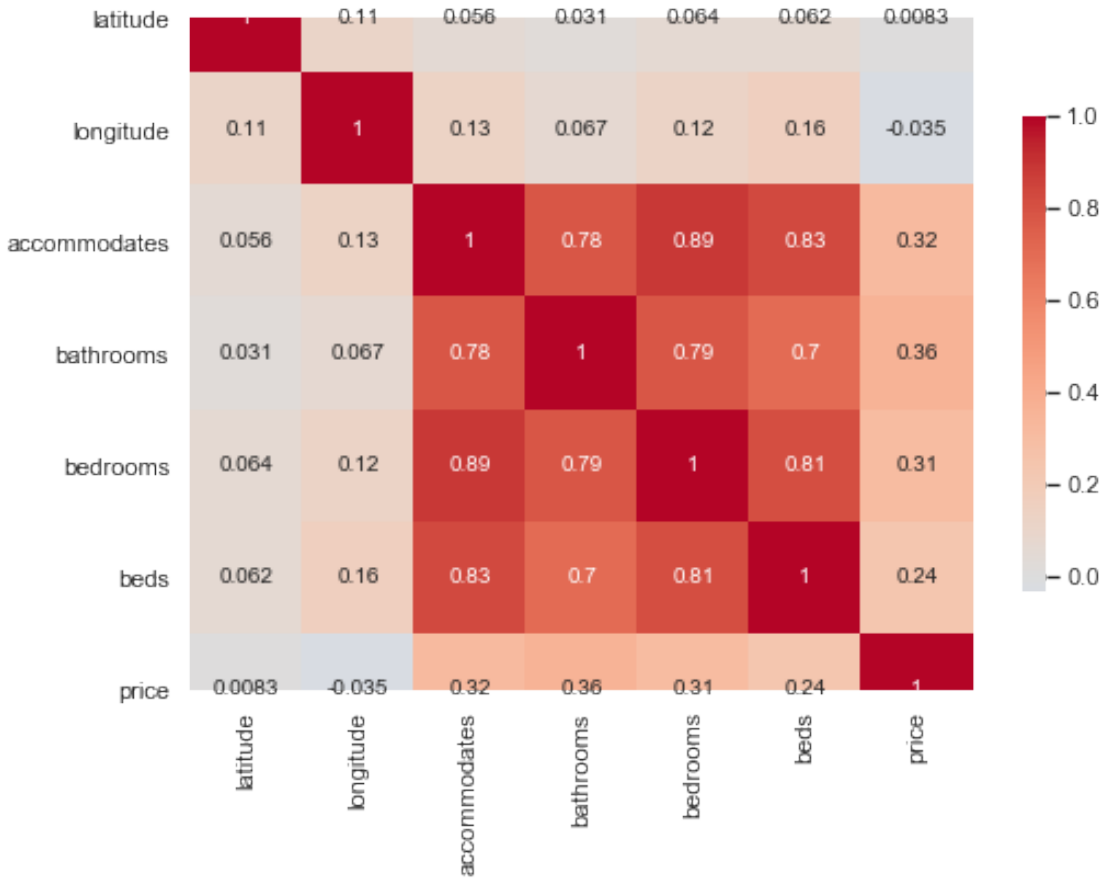
[17]:
```
dataFrameToPlot.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17608 entries, 0 to 17607
Data columns (total 7 columns):
latitude        17608 non-null float64
```

```
longitude       17608 non-null float64
accommodates    17608 non-null int64
bathrooms       17600 non-null float64
bedrooms        17333 non-null float64
beds            17511 non-null float64
price           17608 non-null float64
dtypes: float64(6), int64(1)
memory usage: 963.1 KB
```

[18]:
```
# Computa la matriz de correlación
corr = dataFrameToPlot.corr()

# Crea una figura de tamaño 12, 12
f, ax = plt.subplots(figsize=(8, 8))

# Draw the heatmap with the mask and correct aspect ratio
sns.heatmap(corr, cmap='coolwarm', center=0, square=True, cbar_kws={"shrink": .
↪5}, annot=True, ax=ax)
```

[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1a8ccf144c8>

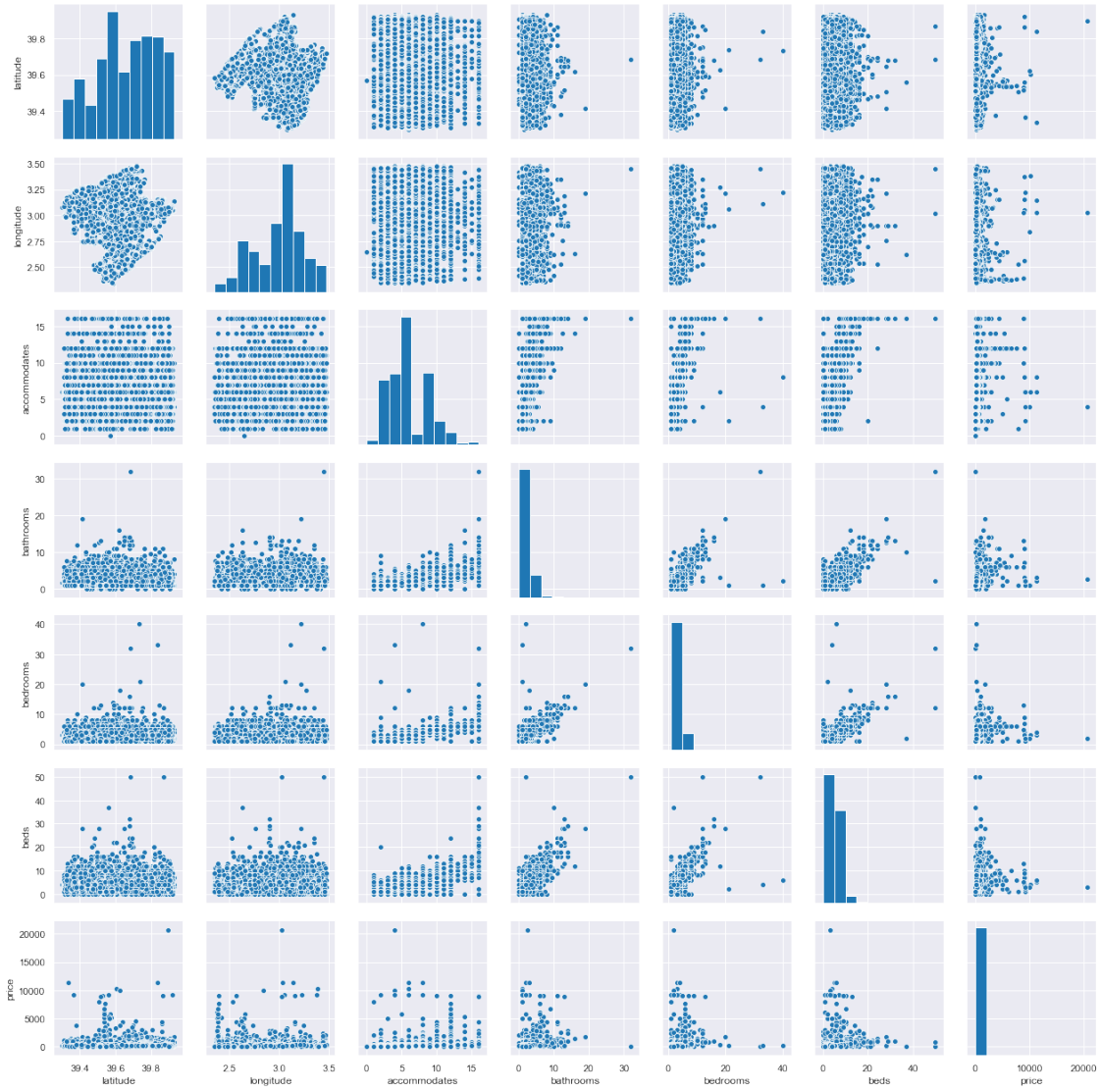


Una vez hemos hecho el plot de la matriz de correlación vemos que hay variables que tiene una fuerte correlación entre ellas. Por ejemplo los baños y el número de habitaciones. Lo interessante de esta matriz es que hay una fuerte correlación del dato de precio con las habitaciones, los baños y las camas que permiten ver que seguramente estos datos són bastante buenos para poder predcir el precio. Por otra parte los datos de longitud y latitud tienen una menor correlación entre ellos y de esta manera se puede incluso plantear si descartarlos seria bueno si el computo es lento o el resultado no es el esperado

[19]:
```
sns.pairplot(dataFrameToPlot)
```

```
C:\Users\Mangu\Anaconda3\lib\site-packages\numpy\lib\histograms.py:824:
RuntimeWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\Mangu\Anaconda3\lib\site-packages\numpy\lib\histograms.py:825:
RuntimeWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

[19]:
```
<seaborn.axisgrid.PairGrid at 0x1a8cd25d248>
```



Dividimos entre variables dependientes e independientes

[20]:
```
X = df_transform[['latitude',␣
 ↪'longitude','room_type','accommodates','bathrooms', 'bedrooms','beds']]
y = df_transform['price']
```

3.1 Defínción de la estrategia a seguir

Se va a afrontar el problema de manera que intentaremos predecir el problema de manera

Vamos a transformar la variable categorica roomType a numerico para poder crear pipelines

[21]:
```
from sklearn.base import TransformerMixin
from sklearn.pipeline import Pipeline
```

```
### aux functions

class SelectColumns(TransformerMixin):
    def __init__(self, columns: list) -> pd.DataFrame:
        if not isinstance(columns, list):
            raise ValueError('Specify the columns into a list')
        self.columns = columns
    def fit(self, X, y=None): # we do not need to specify the target in the␣
 ↪transformer. We leave it as optional arg for consistency
        return self
    def transform(self, X):
        return X[self.columns]

class DropColumns(TransformerMixin):
    def __init__(self, columns: list) -> pd.DataFrame:
        if not isinstance(columns, list):
            raise ValueError('Specify the columns into a list')
        self.columns = columns
    def fit(self, X, y=None):
        return self
    def transform(self, X):
        return X.drop(self.columns, axis=1)
```

Vamos a crear el step de modificar la variable categorica para poder tratarla como númerica y poder usarla como datos de entreno

[22]:
```
from sklearn.preprocessing import OneHotEncoder

select_col_step = ('select', SelectColumns(['room_type']))

one_hot_step = ('room_type_one_hot', OneHotEncoder(sparse=False))

cat_pipe_steps = [select_col_step, one_hot_step]

cat_pipe = Pipeline(cat_pipe_steps)
```

[23]:
```
df_transform_final = cat_pipe.fit_transform(df_transform)
```

[24]:
```
pd.DataFrame(df_transform_final)
```

[24]:

	0	1	2	3
0	1.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0
2	1.0	0.0	0.0	0.0
3	1.0	0.0	0.0	0.0
4	0.0	0.0	1.0	0.0
…	…	…	…	…
17603	1.0	0.0	0.0	0.0

```
17604  1.0  0.0  0.0  0.0
17605  1.0  0.0  0.0  0.0
17606  0.0  0.0  1.0  0.0
17607  1.0  0.0  0.0  0.0

[17608 rows x 4 columns]
```

[25]:
```
cat_pipe['room_type_one_hot'].categories_
```

[25]:
```
[array(['Entire home/apt', 'Hotel room', 'Private room', 'Shared room'],
       dtype=object)]
```

Creamos el step para poder hacer el escalado de los datos para poder modificar las entradas que sean NaN por valores medios, y metemos el sistema para no tener colinearidad

[26]:
```
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import PolynomialFeatures
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

drop_column_step = ('drop_column', DropColumns(['room_type']))

imputation_step = ('imputer', SimpleImputer(strategy='mean'))

scaler_step = ('scaler', MinMaxScaler())

pca_step = ('pca', PCA())

num_pipe_steps = [drop_column_step, imputation_step,  scaler_step, pca_step]

num_pipe = Pipeline(num_pipe_steps)
```

Creamos una full pipe que une las 2 anteriores

[27]:
```
from sklearn.pipeline import FeatureUnion
from sklearn.ensemble import RandomForestRegressor

transformer_list = [('num_pipe', num_pipe),('cat_pipe', cat_pipe)]

full_pipe = FeatureUnion(transformer_list=transformer_list)

full_pipe
```

[27]:
```
FeatureUnion(n_jobs=None,
             transformer_list=[('num_pipe',
                                Pipeline(memory=None,
                                         steps=[('drop_column',
```

```
                                                 <__main__.DropColumns object at
0x000001A8D4B6C048>),
                                                ('imputer',
SimpleImputer(add_indicator=False,
                                                             copy=True,
                                                             fill_value=None,
missing_values=nan,
                                                             strategy='mean',
                                                             verbose=0)),
                                                ('scaler',
                                                 MinMaxScaler(copy=True,
                                                              feature_range=(0,
1))),
                                                ('pca',
                                                 PCA(copy=True,
                                                     iterated…
                                                     whiten=False))],
                                         verbose=False)),
                                ('cat_pipe',
                                 Pipeline(memory=None,
                                         steps=[('select',
                                                 <__main__.SelectColumns object
at 0x000001A8D0606B88>),
                                                ('room_type_one_hot',
OneHotEncoder(categorical_features=None,
                                                             categories=None,
                                                             drop=None,
                                                             dtype=<class
'numpy.float64'>,
handle_unknown='error',
                                                             n_values=None,
                                                             sparse=False))],
                                         verbose=False))],
             transformer_weights=None, verbose=False)
```

3.2 Pipe para resolver el problema de regression

Vamos a decidir el mejor modelo para hacer la regression teniamos 2 opciones de poder Una regression lineal normal o un arbol de decision. Se ha decidido usar un random forest regresor debido a las características de esto ya que mejoras a los arboles de decision normales.

Dividimos los datos en test y train con la función train_test_split

[28]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(df_transform.drop('price',␣
 ↪axis=1), df_transform.price, test_size=0.2)
```

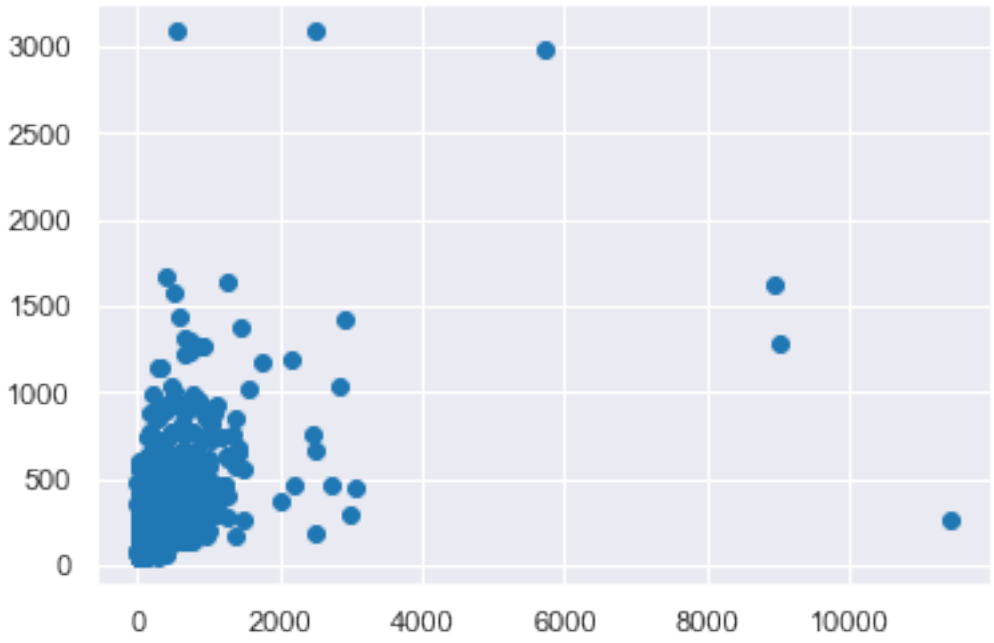
[35]:
```
from sklearn.model_selection import GridSearchCV

regressor_step = ('model', RandomForestRegressor())

full_pipe_step = ('def_pipeline', full_pipe)
pipe_steps = [full_pipe_step, regressor_step]
pipe = Pipeline(pipe_steps)

param_grid = {'def_pipeline__num_pipe__imputer__strategy': ['mean', 'median',␣
 ↪'most_frequent', 'constant'],
              'model__max_depth': [3, 5, 8, 10, 12 ,15, 17, 20, 25, 30],
              'model__min_samples_leaf': [2, 5, 8, 10, 15, 17, 20, 25, 30]}

## usamos el pipe como estimador en el gridsearch
rg_gs = GridSearchCV(pipe, cv=5, n_jobs=-1, param_grid=param_grid, verbose=1)

rg_gs.fit(X_train, y_train)

print('Best params: ', rg_gs.best_params_)
print('Best score: ', rg_gs.best_score_)

y_predict = rg_gs.predict(X_test)
```

```
Fitting 5 folds for each of 360 candidates, totalling 1800 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:    9.3s
[Parallel(n_jobs=-1)]: Done 192 tasks      | elapsed:   41.9s
[Parallel(n_jobs=-1)]: Done 442 tasks      | elapsed:  1.9min
[Parallel(n_jobs=-1)]: Done 792 tasks      | elapsed:  3.3min
[Parallel(n_jobs=-1)]: Done 1242 tasks      | elapsed:  5.3min
[Parallel(n_jobs=-1)]: Done 1792 tasks      | elapsed:  7.7min
[Parallel(n_jobs=-1)]: Done 1800 out of 1800 | elapsed:  7.7min finished
C:\Users\Mangu\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245:
FutureWarning: The default value of n_estimators will change from 10 in version
0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)

Best params:  {'def_pipeline__num_pipe__imputer__strategy': 'mean',
'model__max_depth': 17, 'model__min_samples_leaf': 10}
Best score:  0.21799793871683396
```

Una vez hecho el grid search vemos que la mejor estrategia para los Nan es la de constant, cosa que sorprende que vaya mejor esta que la media aritmética. Además aunque se haya hecho una cross validation en el grid serach el mejor resultado sea tan bajo.

[34]:
```
plt.scatter(y_test, y_predict)
```

[34]:
```
<matplotlib.collections.PathCollection at 0x1a8d4e36cc8>
```



En esta gráfica podemos ver el resultado de las predicciones donde podemos ver que las predicciones no són buenas, sobretodo por los outliers aunque el error igualmente hay una gran cantidad de errores en las predicciones. Estos errores pueden deberse a muchos factores. Primero podría ser que los datos seleccionados al principio no eran los correctos por ejemplo los ammenities quizas influencian en el resultado final si hubieran sido elegidas en el model. También decir que muchos de los datos que se visualizan en la gràfica estan muy concentrados los valores que aciertan mejor. Por eso se tendrian que analizar las características de las casas para poder modificar los inputs de la regression.

[32]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

def MAPE(y_true, y_pred):
    """
    Returns de MAPE of two vectors.
    """
    return np.mean(np.abs((y_true - y_pred)/y_true))

print("MAE:", mean_absolute_error(y_test, y_predict))
print("MSE:", mean_squared_error(y_test, y_predict))
print("MAPE:", MAPE(y_test, y_predict))
print("r2_score:", r2_score(y_test, y_predict))
```

```
MAE: 107.5690577628779
MQE: 105997.06117980478
MAPE: 0.5912127586943915
r2_score: 0.24588574235007288
```

Finalmente podemos ver que las metricas nos indican que hay un error muy grande. Aunque a primera vista se puede pensar que el error medio es solo de aproximadamente de 110€ esto puede ser una cifra que podría afectar mucho a las ventas de airbnb. Lo que podemos ver de las metricas es que el modelo da mucha importancia a valores grande cosa que el tener casa de valor alto y predicciones con error grande, por tant se tendria que estudiar estos valores en concreto. Por último el r2 indica que las predicciones del modelo no són buenas y se deberia revisar si la estrategia usada es incorrecta, porque o si el problema viene de los datos que se han utilitzado como input y porque funciona así.