«Data Mining course»

Project 1 - AirBnb

Dr.Farahani , Dr.Kheradpisheh

Ashkan Safavi Sohi

98422096

About dataset

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present more unique, personalized way of experiencing the world. This dataset describes the listing activity and metrics in NYC, NY for 2019.This data file includes all needed information to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions.

Analyze and Questions

1. What can we learn about different hosts and areas?

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	numbe
0	2539	Clean & quiet apt home by the park	2787	John	Brooklyn	Kensington	40.64749	-73.97237	Private room	149	1	
1	2595	Skylit Midtown Castle	2845	Jennifer	Manhattan	Midtown	40.75362	-73.98377	Entire home/apt	225	1	
2	3647	THE VILLAGE OF HARLEM....NEW YORK !	4632	Elisabeth	Manhattan	Harlem	40.80902	-73.94190	Private room	150	3	
3	3831	Cozy Entire Floor of Brownstone	4869	LisaRoxanne	Brooklyn	Clinton Hill	40.68514	-73.95976	Entire home/apt	89	1	
4	5022	Entire Apt: Spacious Studio/Loft by central park	7192	Laura	Manhattan	East Harlem	40.79851	-73.94399	Entire home/apt	80	10	
...	...	...	...	...	...	...	...	...	...	...	...	
48890	36484665	Charming one bedroom - newly renovated rowhouse	8232441	Sabrina	Brooklyn	Bedford-Stuyvesant	40.67853	-73.94995	Private room	70	2	
48891	36485057	Affordable room in Bushwick/East Williamsburg	6570630	Marisol	Brooklyn	Bushwick	40.70184	-73.93317	Private room	40	4	
48892	36485431	Sunny Studio at Historical Neighborhood	23492952	Ilgar & Aysel	Manhattan	Harlem	40.81475	-73.94867	Entire home/apt	115	10	
48893	36485609	43rd St. Time Square-cozy single bed	30985759	Taz	Manhattan	Hell's Kitchen	40.75751	-73.99112	Shared room	55	1	
48894	36487245	Trendy duplex in the very heart of Hell's Kitchen	68119814	Christophe	Manhattan	Hell's Kitchen	40.76404	-73.98933	Private room	90	7	

48895 rows × 16 columns

In the first step, we realize that our database consists of 16 columns and 48895 rows.

```
In [5]: list(ds.columns)

Out[5]: ['id',
         'name',
         'host_id',
         'host_name',
         'neighbourhood_group',
         'neighbourhood',
         'latitude',
         'longitude',
         'room_type',
         'price',
         'minimum_nights',
         'number_of_reviews',
         'last_review',
         'reviews_per_month',
         'calculated_host_listings_count',
         'availability_365']
```

The information of the columns that are dataset feature can be seen in the list above.

```
In [6]: ds.isnull().sum()

Out[6]: id                                    0
        name                                 16
        host_id                               0
        host_name                            21
        neighbourhood_group                   0
        neighbourhood                         0
        latitude                              0
        longitude                             0
        room_type                             0
        price                                 0
        minimum_nights                        0
        number_of_reviews                     0
        last_review                       10052
        reviews_per_month                 10052
        calculated_host_listings_count        0
        availability_365                      0
        dtype: int64
```

In the list above, it can be seen that this database also contains null data that can be deleted or replaced by the desired number if needed.

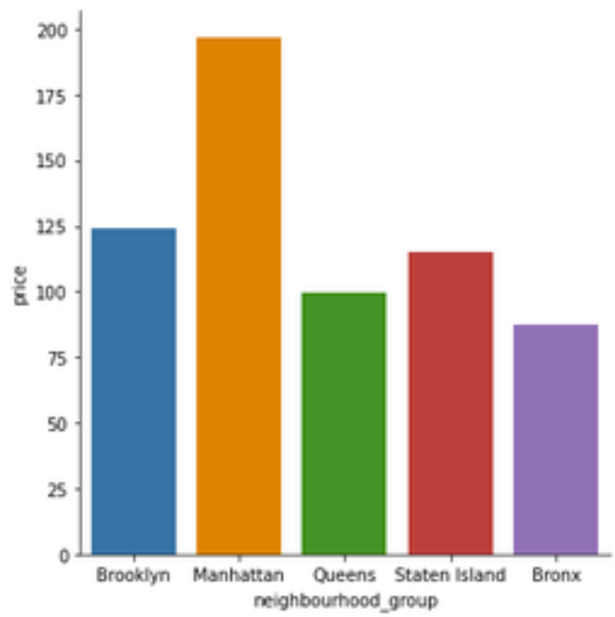
```
In [8]: ds.dtypes

Out[8]: id                                   int64
        name                                object
        host_id                              int64
        host_name                           object
        neighbourhood_group                 object
        neighbourhood                       object
        latitude                           float64
        longitude                          float64
        room_type                           object
        price                                int64
        minimum_nights                       int64
        number_of_reviews                    int64
        last_review                         object
        reviews_per_month                  float64
        calculated_host_listings_count       int64
        availability_365                     int64
        dtype: object
```

This database includes 10 numerical feature and 6 catgorical feature.

2. What can we learn from predictions? (ex: locations, prices, reviews, etc)

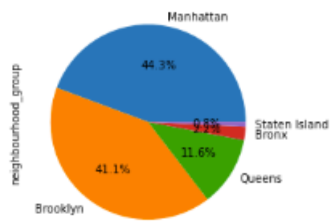
```
In [18]: sns.catplot(x="neighbourhood_group",y='price' ,kind="bar", data=ds,ci=None)
Out[18]: <seaborn.axisgrid.FacetGrid at 0x7fa5afe29220>
```



Manhattan has more expensive hotels than anywhere else

3. Which hosts are the busiest and why?

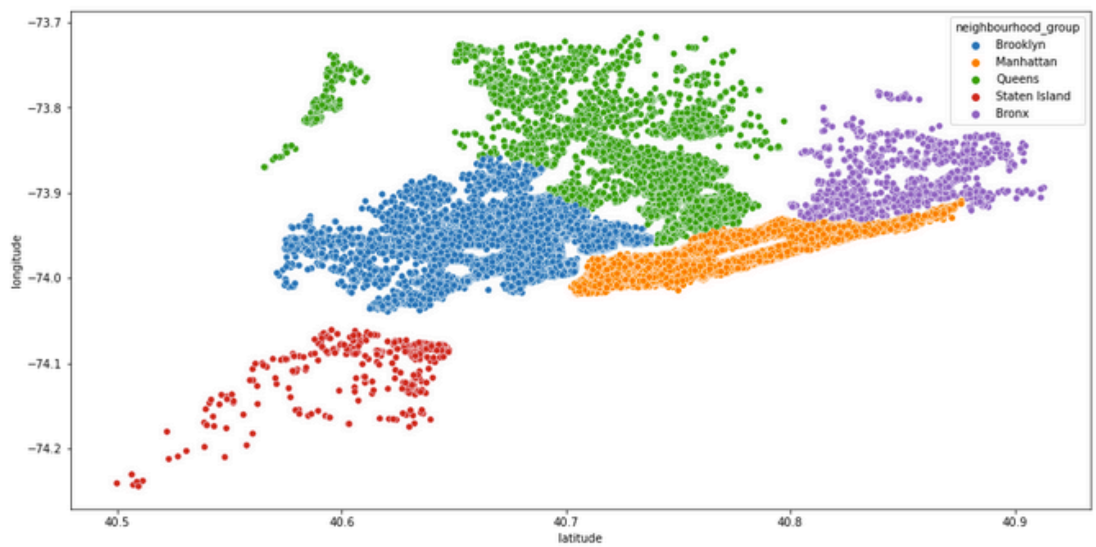
```
In [14]: ds['neighbourhood_group'].value_counts().plot.pie(explode=[0,0,0,0,0],autopct='%1.1f%%',shadow=False)
Out[14]: <AxesSubplot:ylabel='neighbourhood_group'>
```



First Manhattan and then brooklyn have a larger share in the number of hotels

4. Is there any noticeable difference of traffic among different areas and what could be the reason for it?

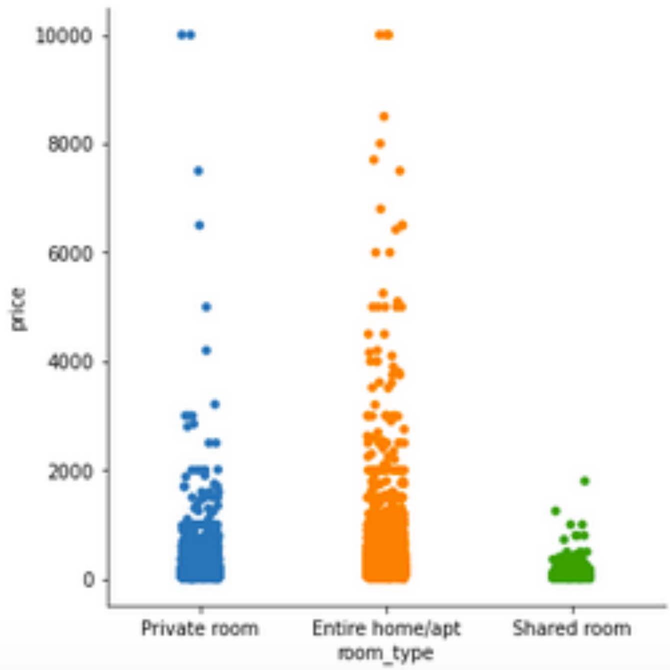
```
In [23]: f , _map = plt.subplots(figsize=(16,8))
         _map = sns.scatterplot(x=ds.latitude,y=ds.longitude,hue=ds.neighbourhood_group,palette='tab10')
```



The chart above shows the distribution of bookers by geographical location , Population density in the range of -74 to -79.3 longitude is quite clear .

5. Price dispersion based on types of places

```
In [27]: sns.catplot(x="room_type",y='price' , data=ds,)
Out[27]: <seaborn.axisgrid.FacetGrid at 0x7fa5b053dcd0>
```



As can be seen from the diagram above, the rooms have various prices.

6. What are the most popular neighborhoods?

```
In [46]: review = ds.sort_values('number_of_reviews',ascending=False)
         top_v = review.loc[:,['neighbourhood','number_of_reviews']][:20]
         top_v = top_v.groupby('neighbourhood').mean().sort_values('number_of_reviews',ascending=False).reset_index()
         sns.catplot(x=top_v["neighbourhood"],y=top_v['number_of_reviews'].values ,kind="bar", data=ds,ci=None)
Out[46]: <seaborn.axisgrid.FacetGrid at 0x7fa5b21ab550>
```