Low Level Design

Airbnb Data Analysis

Written By	Ajit Kumar
Document Version	1
Last Revised Date	

Document Version Control

Date Issued	Version	Description	Author
8 th july 2022	1.0	First Version of Complete LLD	Ajit Kumar

Contents

1. Introduction

1.1 What is Low-Level design document?

The goal of the LDD or Low-level design document (LLDD) is to give the internal logic design of the actual program code for the Airbnb Booking Data . LDD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. The process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

2. Architecture Description

2.1. Data Description

The dataset had information regarding the reviews with respect to listing id.
This data had all the information regarding the listings. It had Host name, location, neighbourhood, price, review score and number of review, latitude, longitude ,room type. etc..

The features in the dataset can be described as follows:

1. room id - This is the identity number of the property listed by a particular host.

2.survey id - This is the identity number of survey.

3. name - It stands for the name of the property listed by the host.

4. Host id - It is the identity number of the hosts who have registered on Airbnb website.

5. room type - This represent the various types of room listed by host.

6. Country – name of the country where the survey has conducted.

7. City - name of the city where the survey has conducted.

8. neighbourhood- These are the names of the neighbourhood or locations present in the city.

9. latitude - These represent the coordinates of latitude of the property listed.

10. longitude - These represent the coordinates of longitude of the property listed.

11. price - This is the rent of the property listed in euro.

12. ministay - This represent the minimum number of nights customer rented the property.

13. reviews - This represent the number of customers reviewed the property.

14. overall_satisfaction – customars has given a rating to a places in 0 to 5.

15. Location – it has given a code of the locations.

16. Bedrooms – no. of bedrooms present In the property.

17. Bathrooms – no. of bathrooms present in the property.

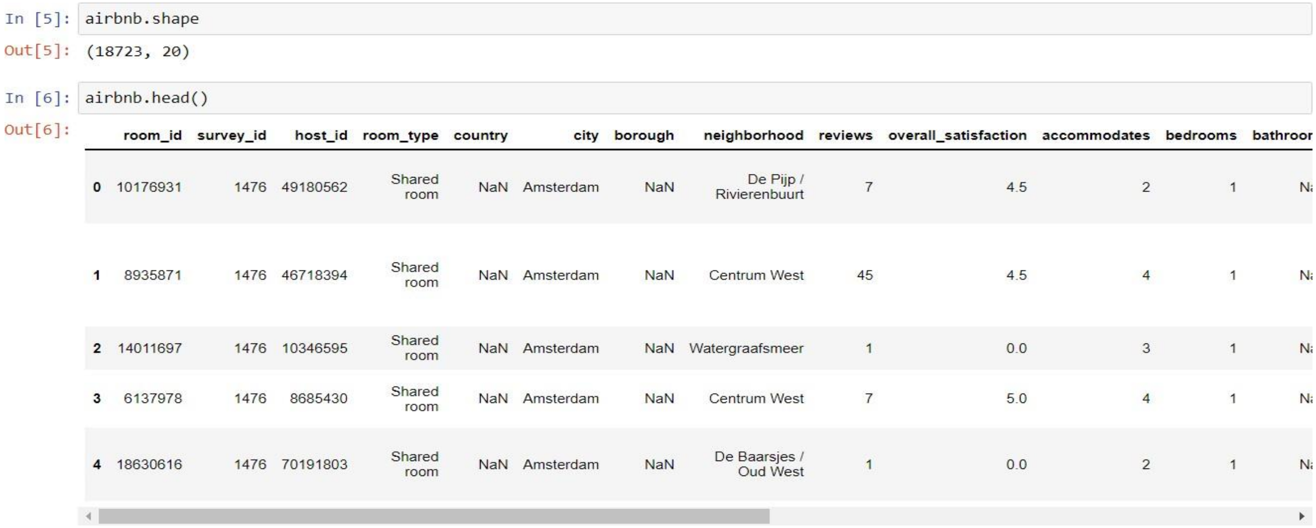
18. Last modification - This represent the date when the property was last reviewed.

2.2. Data Transformation

In the Transformation Process, we will Transform our original datasets excel fil into jupyter notebook for data Exploration and performing Exploratory Data Analysis using python programming language.

2.3. Data Exploration

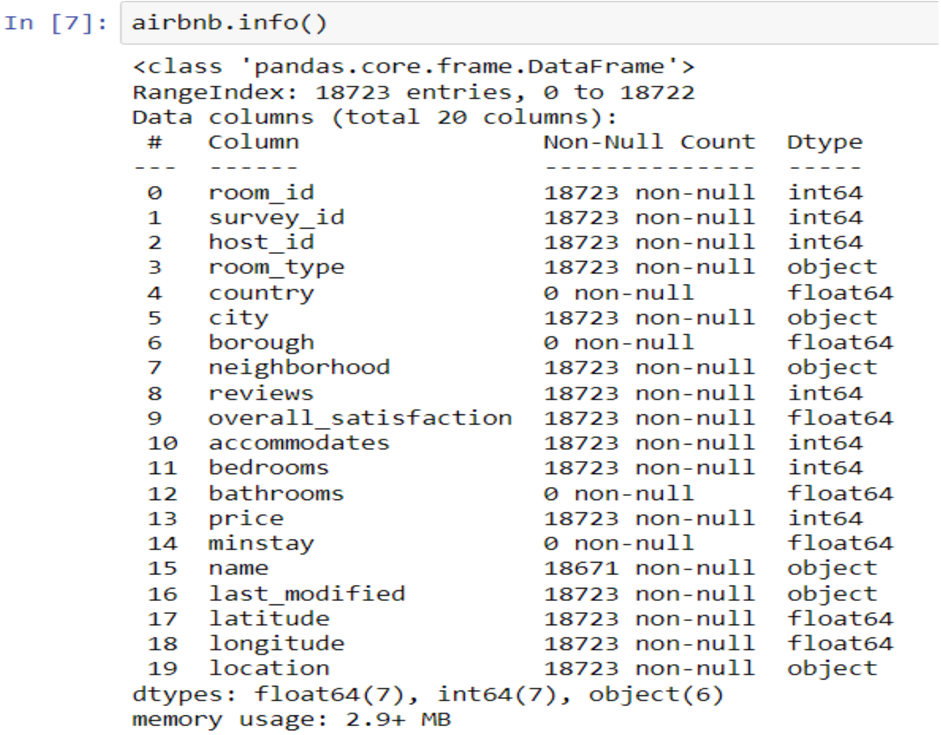
Checking the first 5 rows of the dataset and the dataset consist of 18723 observations (rows) and 20 features (columns).



```
In [5]: airbnb.shape
Out[5]: (18723, 20)

In [6]: airbnb.head()
Out[6]:
```

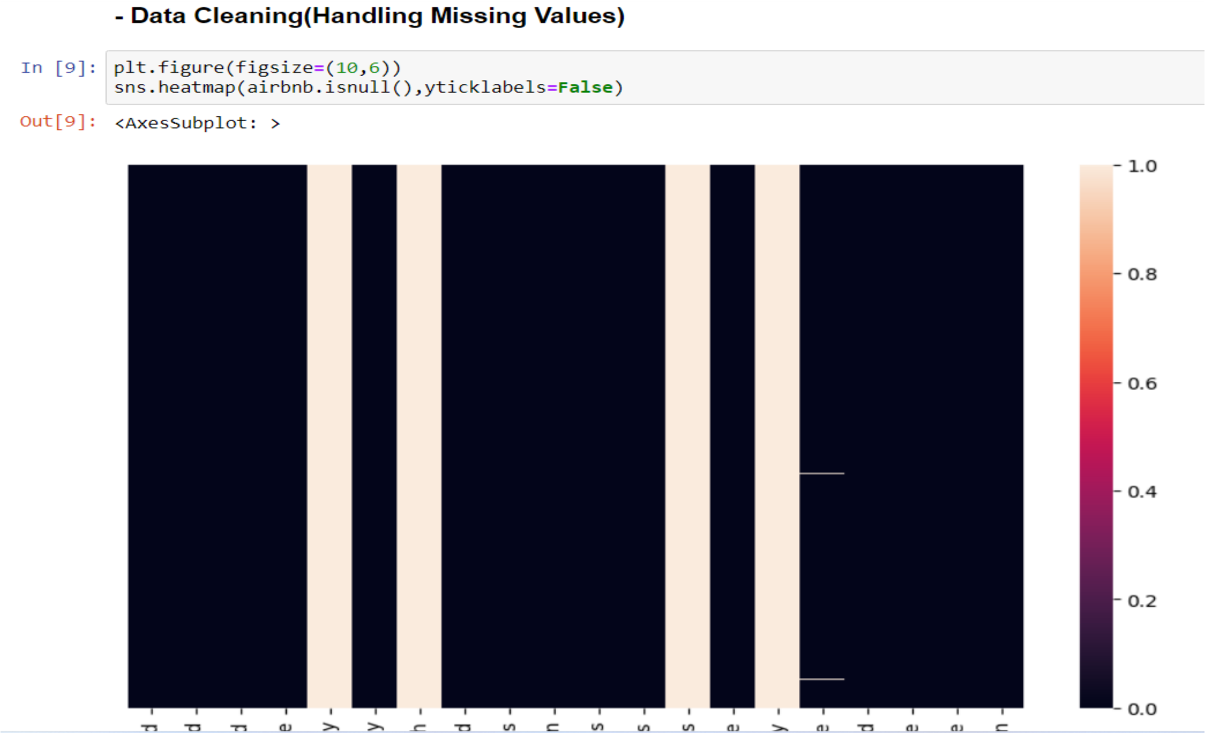
	room_id	survey_id	host_id	room_type	country	city	borough	neighborhood	reviews	overall_satisfaction	accommodates	bedrooms	bathroor
0	10176931	1476	49180562	Shared room	NaN	Amsterdam	NaN	De Pijp / Rivierenbuurt	7	4.5	2	1	Na
1	8935871	1476	46718394	Shared room	NaN	Amsterdam	NaN	Centrum West	45	4.5	4	1	Na
2	14011697	1476	10346595	Shared room	NaN	Amsterdam	NaN	Watergraafsmeer	1	0.0	3	1	Na
3	6137978	1476	8685430	Shared room	NaN	Amsterdam	NaN	Centrum West	7	5.0	4	1	Na
4	18630616	1476	70191803	Shared room	NaN	Amsterdam	NaN	De Baarsjes / Oud West	1	0.0	2	1	Na

Checking out the 20 features: categorical columns and non-categorical columns in the dataset.



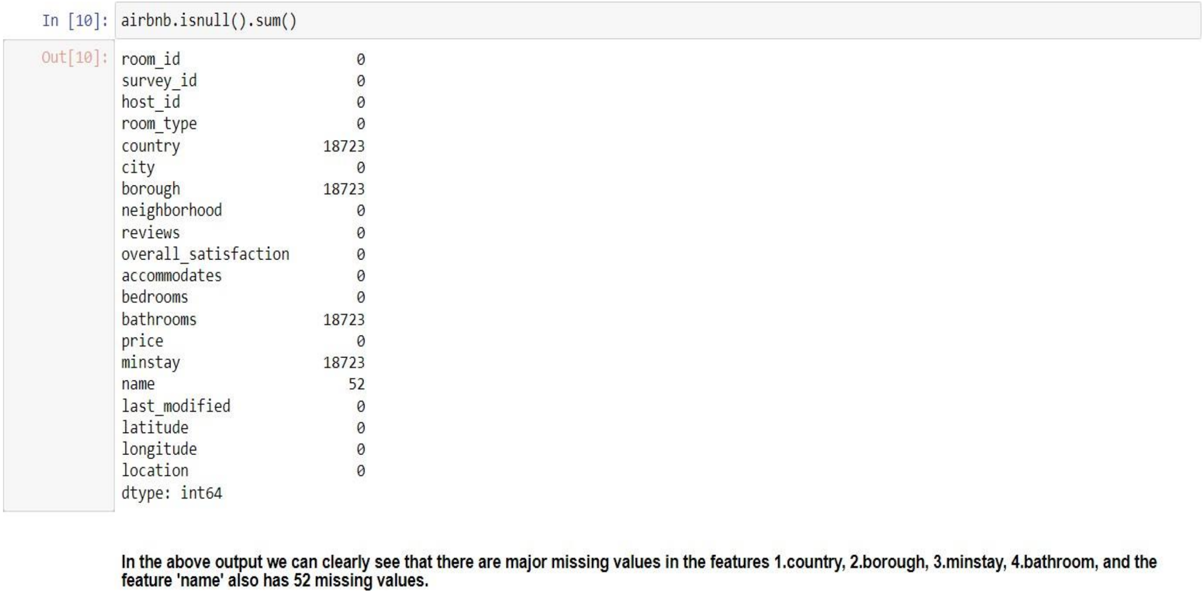
```
In [7]: airbnb.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18723 entries, 0 to 18722
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   room_id               18723 non-null  int64
 1   survey_id             18723 non-null  int64
 2   host_id               18723 non-null  int64
 3   room_type             18723 non-null  object
 4   country               0 non-null      float64
 5   city                  18723 non-null  object
 6   borough               0 non-null      float64
 7   neighborhood          18723 non-null  object
 8   reviews               18723 non-null  int64
 9   overall_satisfaction  18723 non-null  float64
 10  accommodates          18723 non-null  int64
 11  bedrooms              18723 non-null  int64
 12  bathrooms             0 non-null      float64
 13  price                 18723 non-null  int64
 14  minstay               0 non-null      float64
 15  name                  18671 non-null  object
 16  last_modified         18723 non-null  object
 17  latitude              18723 non-null  float64
 18  longitude             18723 non-null  float64
 19  location              18723 non-null  object
dtypes: float64(7), int64(7), object(6)
memory usage: 2.9+ MB
```

Checking for null values: The columns like country, borough, bathroom, minstay have largest number of null values. The columns like name contain 52 null values.



- Data Cleaning(Handling Missing Values)

```
In [9]: plt.figure(figsize=(10,6))
        sns.heatmap(airbnb.isnull(),yticklabels=False)
Out[9]: <AxesSubplot: >
```

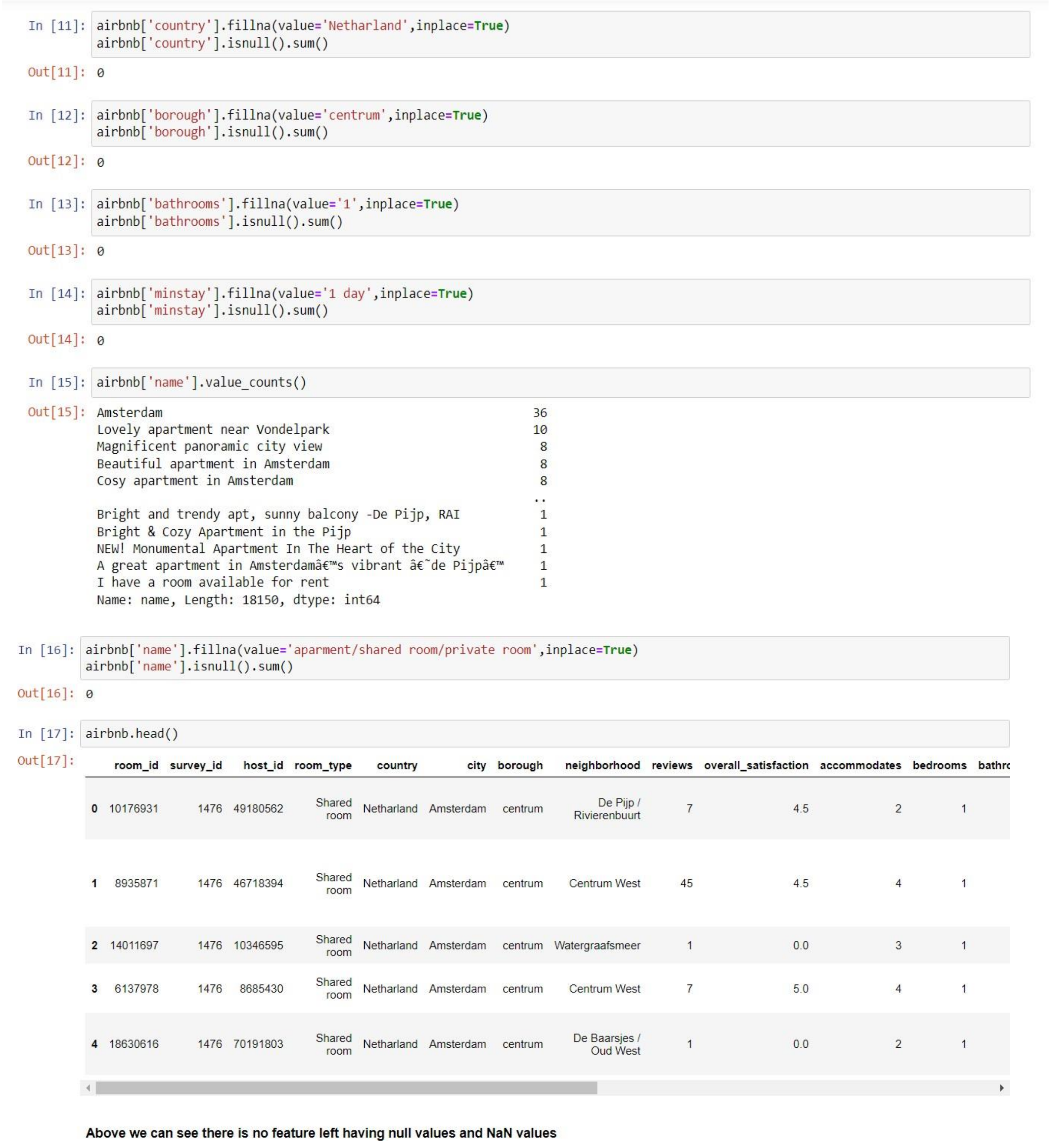


```
In [10]: airbnb.isnull().sum()
Out[10]: room_id                    0
         survey_id                  0
         host_id                    0
         room_type                  0
         country                18723
         city                       0
         borough                18723
         neighborhood               0
         reviews                    0
         overall_satisfaction       0
         accommodates               0
         bedrooms                   0
         bathrooms              18723
         price                      0
         minstay                18723
         name                      52
         last_modified              0
         latitude                   0
         longitude                  0
         location                   0
         dtype: int64
```

In the above output we can clearly see that there are major missing values in the features 1.country, 2.borough, 3.minstay, 4.bathroom, and the feature 'name' also has 52 missing values.

2.4 Data Cleaning

Fixing the null values: We have filled the null values i.e for country-Netharland, Borough-centrum, bathroom-1, mainstay-1day, name-apartment/shared/private room.

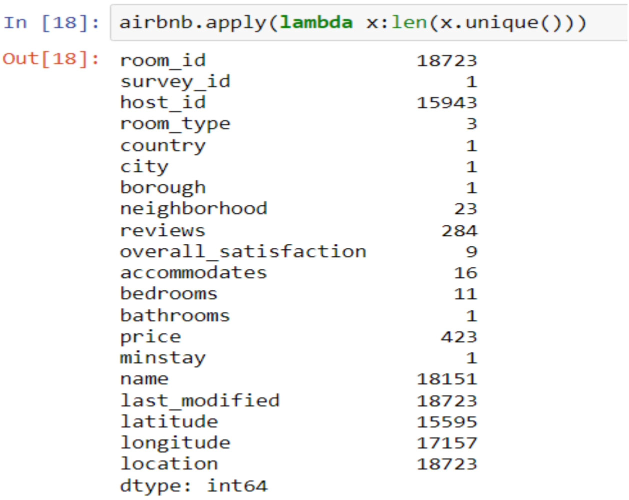


```
In [11]: airbnb['country'].fillna(value='Netharland',inplace=True)
         airbnb['country'].isnull().sum()

Out[11]: 0

In [12]: airbnb['borough'].fillna(value='centrum',inplace=True)
         airbnb['borough'].isnull().sum()

Out[12]: 0

In [13]: airbnb['bathrooms'].fillna(value='1',inplace=True)
         airbnb['bathrooms'].isnull().sum()

Out[13]: 0

In [14]: airbnb['minstay'].fillna(value='1 day',inplace=True)
         airbnb['minstay'].isnull().sum()

Out[14]: 0

In [15]: airbnb['name'].value_counts()

Out[15]: Amsterdam                                                     36
         Lovely apartment near Vondelpark                              10
         Magnificent panoramic city view                                8
         Beautiful apartment in Amsterdam                               8
         Cosy apartment in Amsterdam                                    8
                                                                       ..
         Bright and trendy apt, sunny balcony -De Pijp, RAI             1
         Bright & Cozy Apartment in the Pijp                            1
         NEW! Monumental Apartment In The Heart of the City             1
         A great apartment in Amsterdamâ€™s vibrant â€˜de Pijpâ€™        1
         I have a room available for rent                               1
         Name: name, Length: 18150, dtype: int64

In [16]: airbnb['name'].fillna(value='aparment/shared room/private room',inplace=True)
         airbnb['name'].isnull().sum()

Out[16]: 0

In [17]: airbnb.head()

Out[17]:
```

	room_id	survey_id	host_id	room_type	country	city	borough	neighborhood	reviews	overall_satisfaction	accommodates	bedrooms	bathr
0	10176931	1476	49180562	Shared room	Netharland	Amsterdam	centrum	De Pijp / Rivierenbuurt	7	4.5	2	1	
1	8935871	1476	46718394	Shared room	Netharland	Amsterdam	centrum	Centrum West	45	4.5	4	1	
2	14011697	1476	10346595	Shared room	Netharland	Amsterdam	centrum	Watergraafsmeer	1	0.0	3	1	
3	6137978	1476	8685430	Shared room	Netharland	Amsterdam	centrum	Centrum West	7	5.0	4	1	
4	18630616	1476	70191803	Shared room	Netharland	Amsterdam	centrum	De Baarsjes / Oud West	1	0.0	2	1	

Above we can see there is no feature left having null values and NaN values

Checking for unique values present in each feature of the dataset.

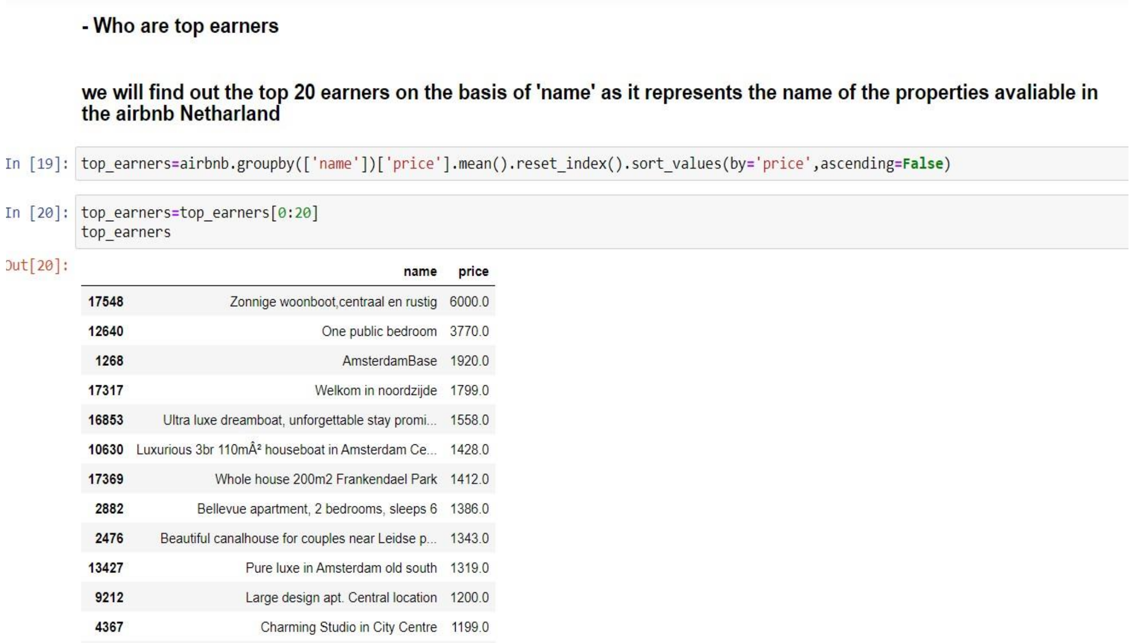


```
In [18]: airbnb.apply(lambda x:len(x.unique()))

Out[18]: room_id                 18723
         survey_id                   1
         host_id                 15943
         room_type                   3
         country                     1
         city                        1
         borough                     1
         neighborhood               23
         reviews                   284
         overall_satisfaction        9
         accommodates               16
         bedrooms                   11
         bathrooms                   1
         price                     423
         minstay                     1
         name                    18151
         last_modified           18723
         latitude                15595
         longitude               17157
         location                18723
         dtype: int64
```

2.5 Data Analysis and Data Visualization

1. Who are top earners:

we will find out the top 20 earners on the basis of 'name' as it represents the name of the properties available in the airbnb Netharland,



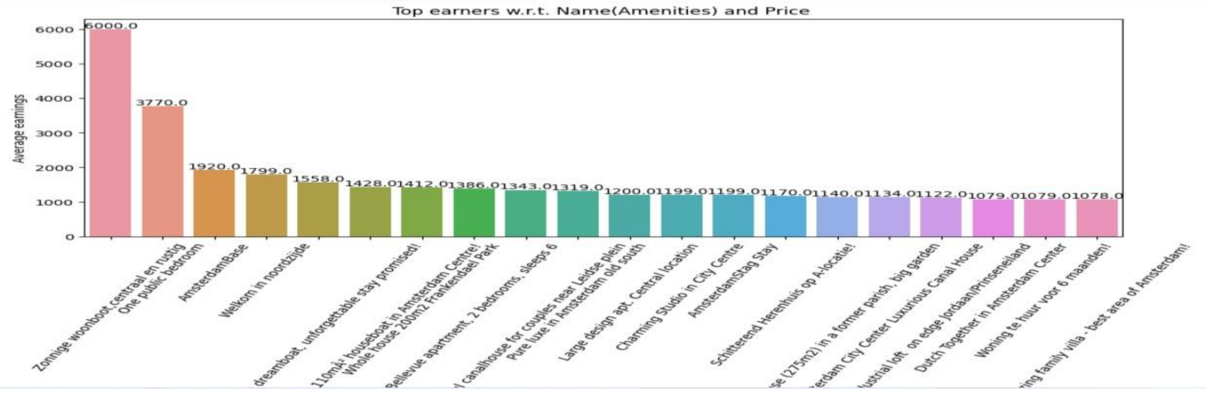
- Who are top earners

we will find out the top 20 earners on the basis of 'name' as it represents the name of the properties avaliable in the airbnb Netharland

```
In [19]: top_earners=airbnb.groupby(['name'])['price'].mean().reset_index().sort_values(by='price',ascending=False)

In [20]: top_earners=top_earners[0:20]
         top_earners
```

Out[20]:

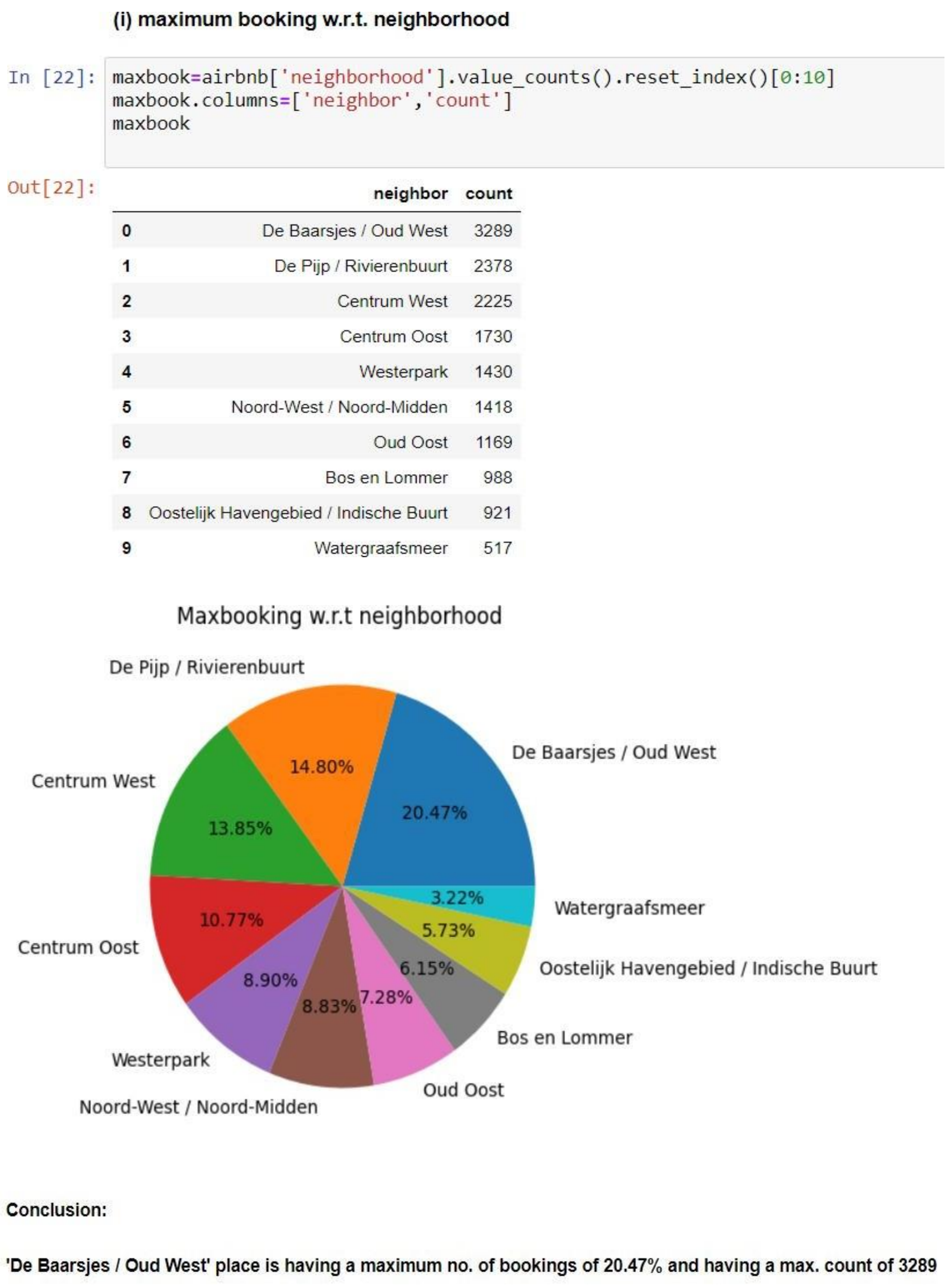
	name	price
17548	Zonnige woonboot,centraal en rustig	6000.0
12640	One public bedroom	3770.0
1268	AmsterdamBase	1920.0
17317	Welkom in noordzijde	1799.0
16853	Ultra luxe dreamboat, unforgettable stay promi...	1558.0
10630	Luxurious 3br 110mÂ² houseboat in Amsterdam Ce...	1428.0
17369	Whole house 200m2 Frankendael Park	1412.0
2882	Bellevue apartment, 2 bedrooms, sleeps 6	1386.0
2476	Beautiful canalhouse for couples near Leidse p...	1343.0
13427	Pure luxe in Amsterdam old south	1319.0
9212	Large design apt. Central location	1200.0
4367	Charming Studio in City Centre	1199.0



We can clearly see that top 3 earners are:
-'Zonnige woonboot,centraal en rustig' with the highest earning of $6000.
-'One public bedroom' with the second highest earning of $3770.
-'AmsterdamBase' with the third highest earning of $1920.

2. Any particular location getting maximum number of bookings:

(i) maximum booking w.r.t. neighborhood



(i) maximum booking w.r.t. neighborhood

```
In [22]: maxbook=airbnb['neighborhood'].value_counts().reset_index()[0:10]
         maxbook.columns=['neighbor','count']
         maxbook
```

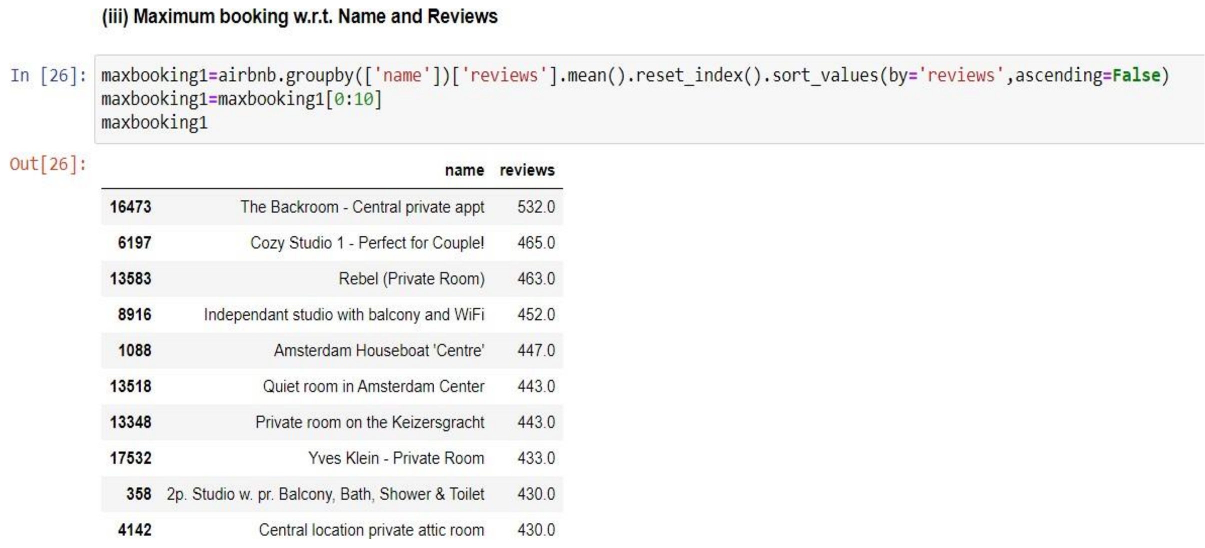
Out[22]:

	neighbor	count
0	De Baarsjes / Oud West	3289
1	De Pijp / Rivierenbuurt	2378
2	Centrum West	2225
3	Centrum Oost	1730
4	Westerpark	1430
5	Noord-West / Noord-Midden	1418
6	Oud Oost	1169
7	Bos en Lommer	988
8	Oostelijk Havengebied / Indische Buurt	921
9	Watergraafsmeer	517

Conclusion:

'De Baarsjes / Oud West' place is having a maximum no. of bookings of 20.47% and having a max. count of 3289

(ii) Maximum booking w.r.t. Name and Reviews

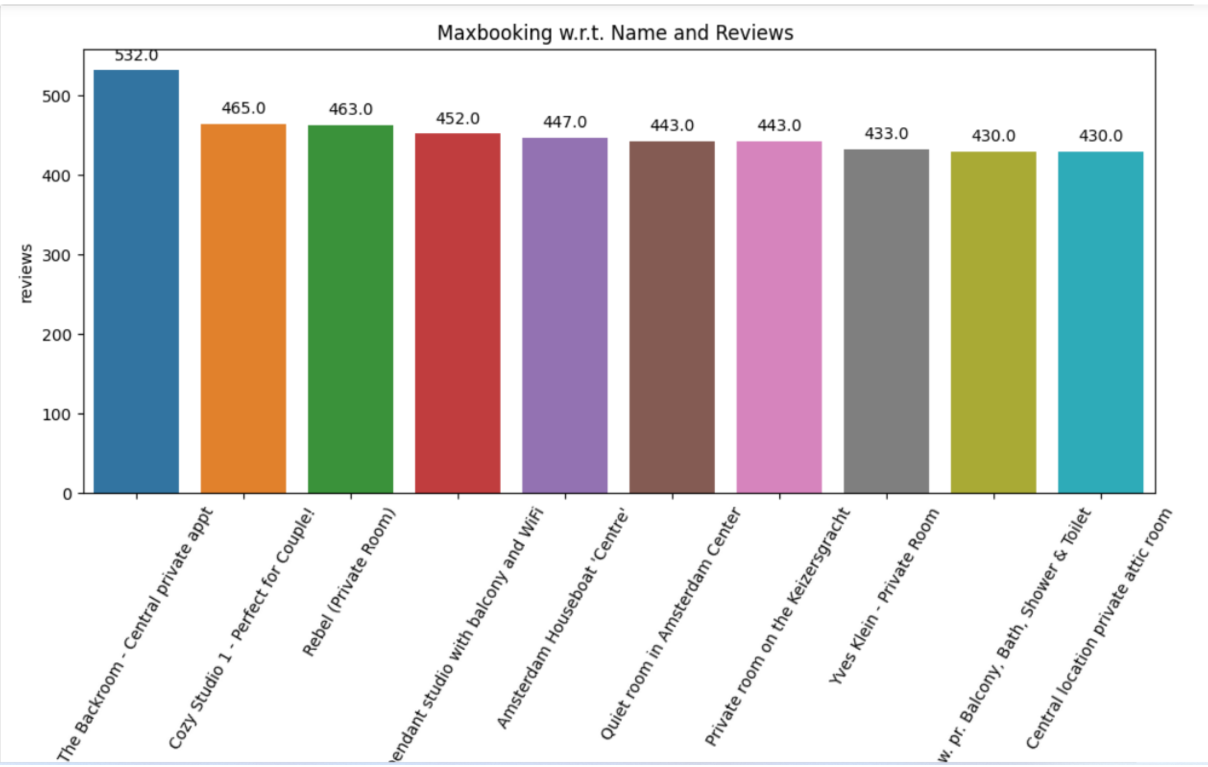


(iii) Maximum booking w.r.t. Name and Reviews

```
In [26]: maxbooking1=airbnb.groupby(['name'])['reviews'].mean().reset_index().sort_values(by='reviews',ascending=False)
         maxbooking1=maxbooking1[0:10]
         maxbooking1
```

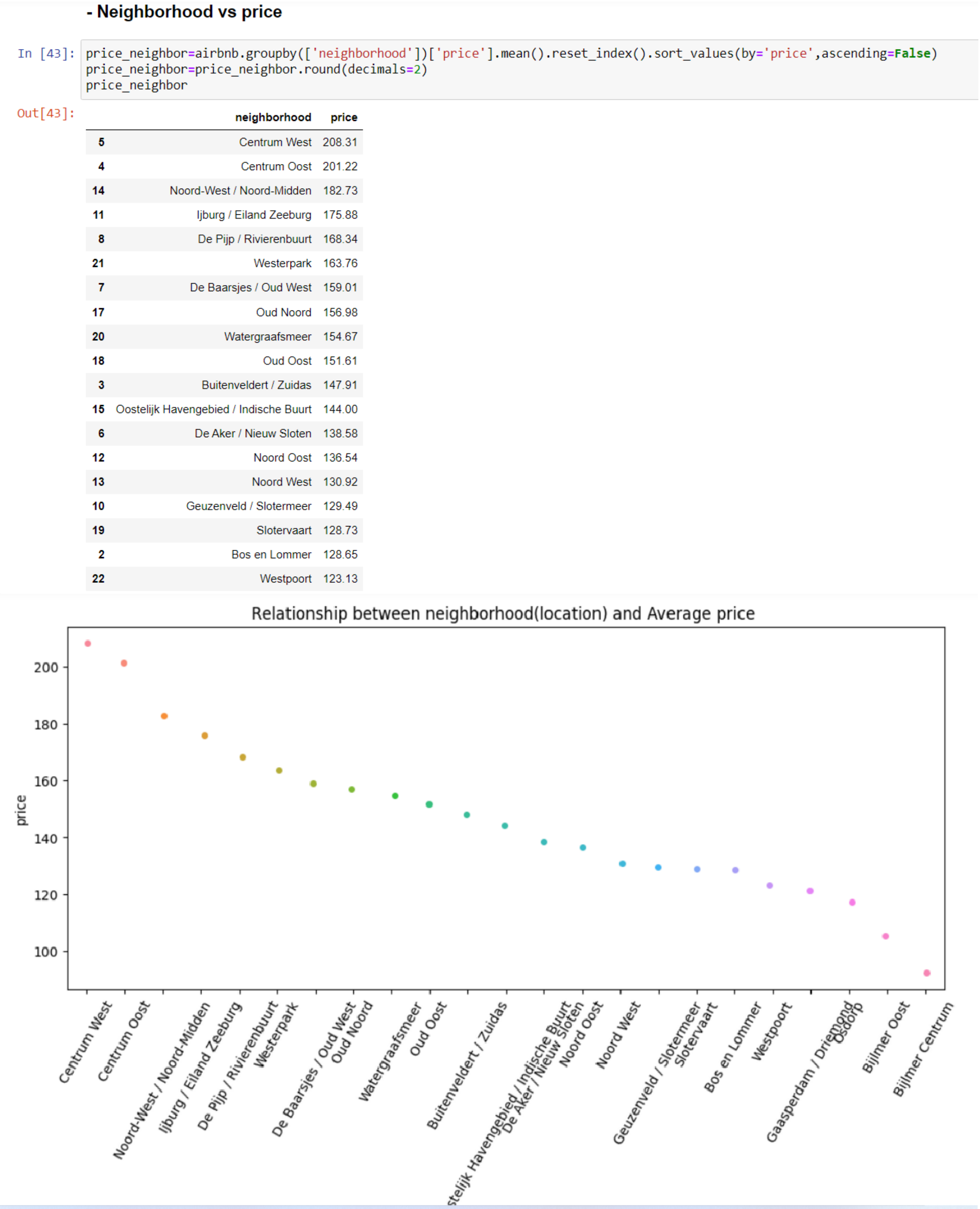
Out[26]:

	name	reviews
16473	The Backroom - Central private appt	532.0
6197	Cozy Studio 1 - Perfect for Couple!	465.0
13583	Rebel (Private Room)	463.0
8916	Independant studio with balcony and WiFi	452.0
1088	Amsterdam Houseboat 'Centre'	447.0
13518	Quiet room in Amsterdam Center	443.0
13348	Private room on the Keizersgracht	443.0
17532	Yves Klein - Private Room	433.0
358	2p. Studio w. pr. Balcony, Bath, Shower & Toilet	430.0
4142	Central location private attic room	430.0



This are the top 5 name of property types having maximum bookings-
-The Backroom - Central private appt
-Cozy Studio 1 - Perfect for Couple! Rebel (Private Room)
-Independant studio with balcony and WiFi
-Amsterdam Houseboat 'Centre'

3. Neighborhood vs price:



- Neighborhood vs price

```
In [43]: price_neighbor=airbnb.groupby(['neighborhood'])['price'].mean().reset_index().sort_values(by='price',ascending=False)
         price_neighbor=price_neighbor.round(decimals=2)
         price_neighbor
```

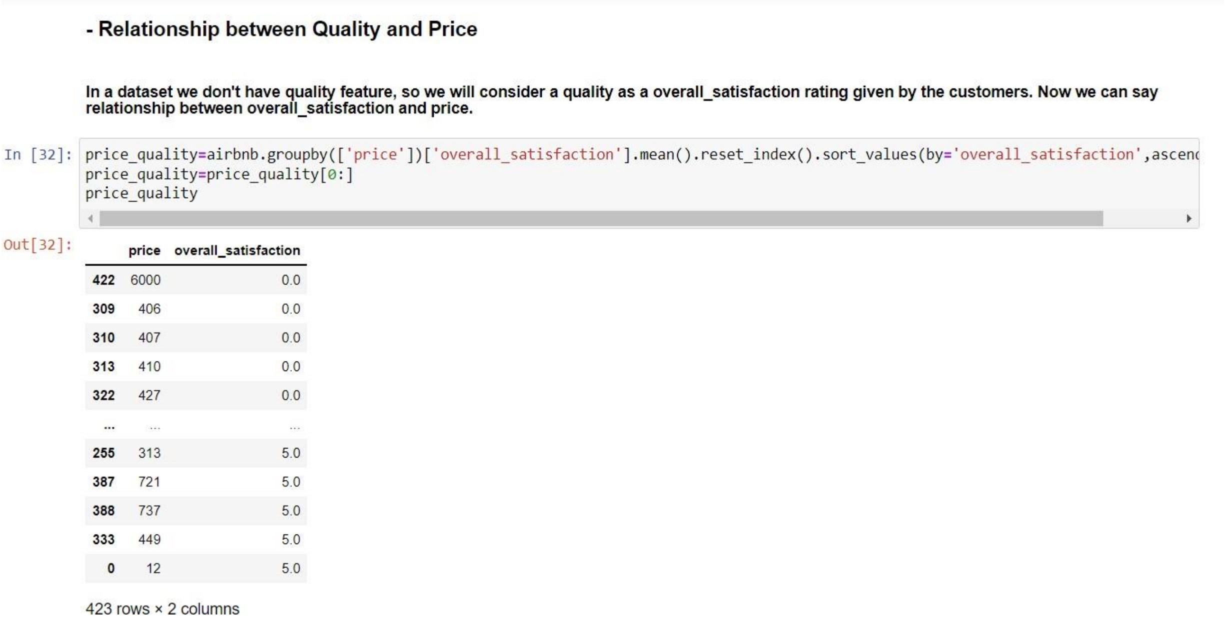
Out[43]:

	neighborhood	price
5	Centrum West	208.31
4	Centrum Oost	201.22
14	Noord-West / Noord-Midden	182.73
11	Ijburg / Eiland Zeeburg	175.88
8	De Pijp / Rivierenbuurt	168.34
21	Westerpark	163.76
7	De Baarsjes / Oud West	159.01
17	Oud Noord	156.98
20	Watergraafsmeer	154.67
18	Oud Oost	151.61
3	Buitenveldert / Zuidas	147.91
15	Oostelijk Havengebied / Indische Buurt	144.00
6	De Aker / Nieuw Sloten	138.58
12	Noord Oost	136.54
13	Noord West	130.92
10	Geuzenveld / Slotermeer	129.49
19	Slotervaart	128.73
2	Bos en Lommer	128.65
22	Westpoort	123.13

Centrum West has a highest average price 208.31
Centrum Oost has a 2nd highest average price 201.22
Noord-West / Noord-Midden has a 3rd highest average price 182.73

4. Relationship between Quality and Price:

In a dataset we don't have quality feature, so we will consider a quality as a overall_satisfaction rating given by the customers. Now we can say relationship between overall_satisfaction and price.



- Relationship between Quality and Price

In a dataset we don't have quality feature, so we will consider a quality as a overall_satisfaction rating given by the customers. Now we can say relationship between overall_satisfaction and price.

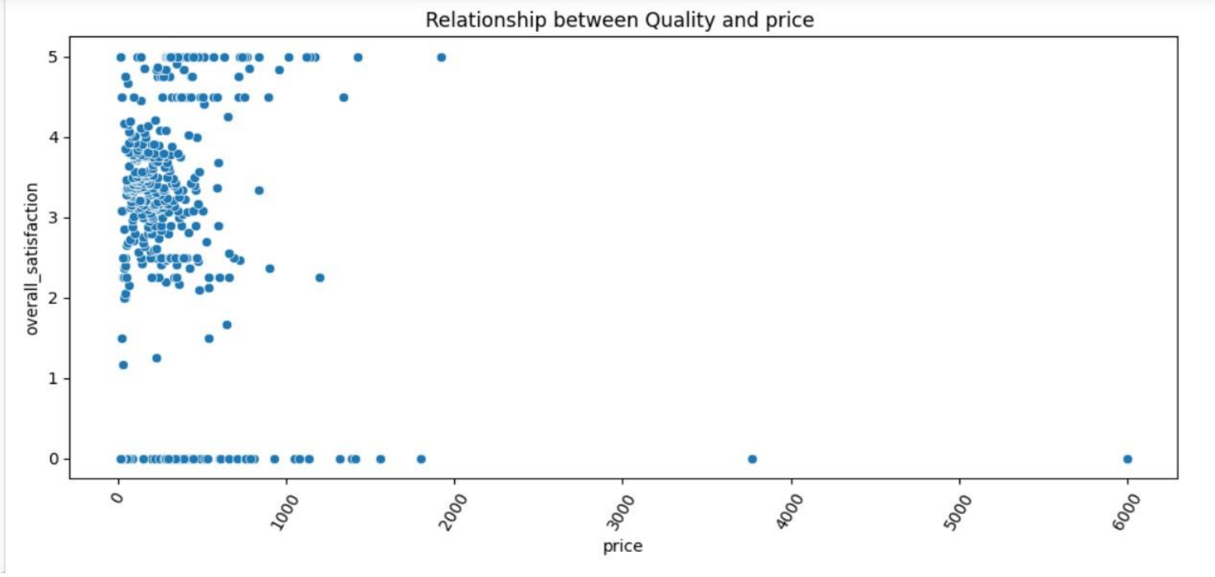
In [32]:

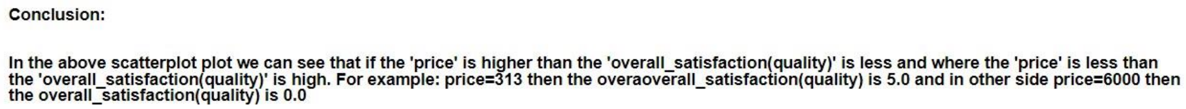
```
price_quality=airbnb.groupby(['price'])['overall_satisfaction'].mean().reset_index().sort_values(by='overall_satisfaction',ascen
price_quality=price_quality[0:]
price_quality
```

Out[32]:

	price	overall_satisfaction
422	6000	0.0
309	406	0.0
310	407	0.0
313	410	0.0
322	427	0.0
...	...	...
255	313	5.0
387	721	5.0
388	737	5.0
333	449	5.0
0	12	5.0

423 rows × 2 columns



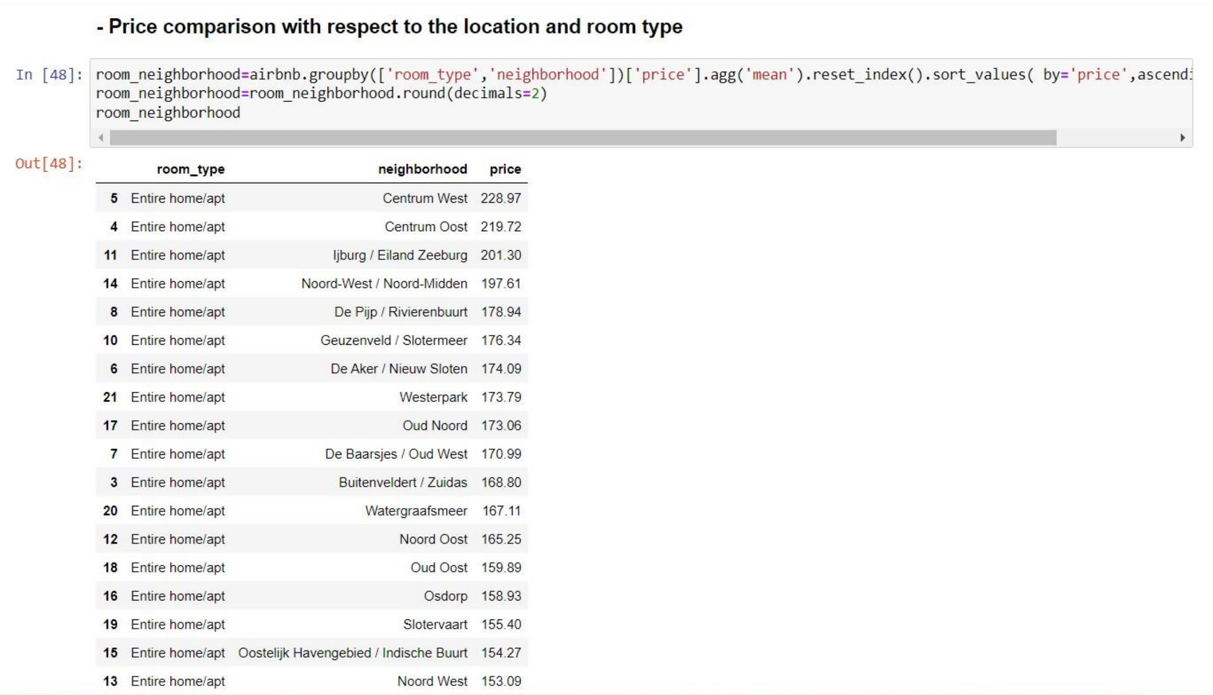


Conclusion:

In the above scatterplot plot we can see that if the 'price' is higher than the 'overall_satisfaction(quality)' is less and where the 'price' is less than the 'overall_satisfaction(quality)' is high. For example: price=313 then the overaoverall_satisfaction(quality) is 5.0 and in other side price=6000 then the overall_satisfaction(quality) is 0.0

In the above scatterplot plot we can see that if the 'price' is higher than the 'overall_satisfaction(quality)' is less and where the 'price' is less than the 'overall_satisfaction(quality)' is high. For example: price=313 then the overaoverall_satisfaction(quality) is 5.0 and in other side price=6000 then the overall_satisfaction(quality) is 0.0

5. Price comparison with respect to the location and room type:

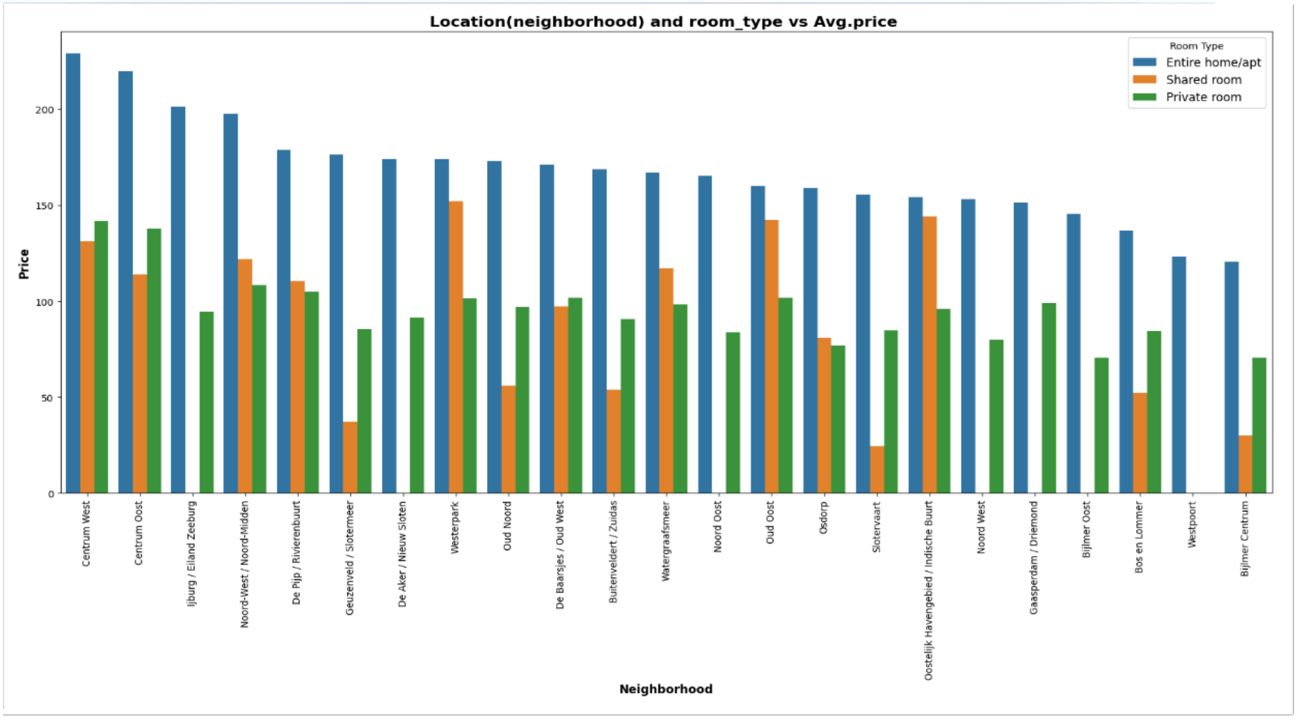


- Price comparison with respect to the location and room type

In [48]:
```
room_neighborhood=airbnb.groupby(['room_type','neighborhood'])['price'].agg('mean').reset_index().sort_values( by='price',ascendi
room_neighborhood=room_neighborhood.round(decimals=2)
room_neighborhood
```

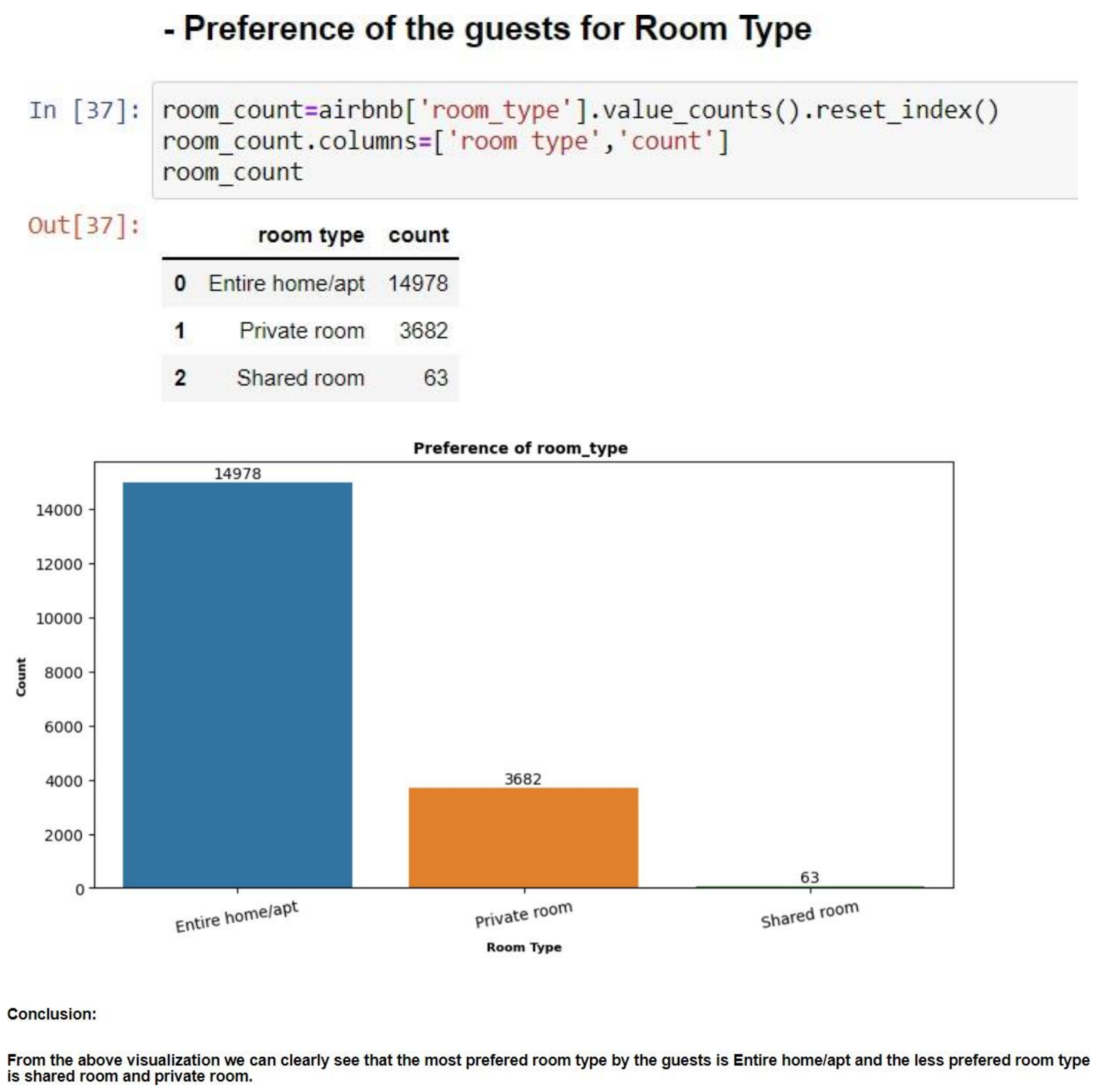
Out[48]:

	room_type	neighborhood	price
5	Entire home/apt	Centrum West	228.97
4	Entire home/apt	Centrum Oost	219.72
11	Entire home/apt	Ijburg / Eiland Zeeburg	201.30
14	Entire home/apt	Noord-West / Noord-Midden	197.61
8	Entire home/apt	De Pijp / Rivierenbuurt	178.94
10	Entire home/apt	Geuzenveld / Slotermeer	176.34
6	Entire home/apt	De Aker / Nieuw Sloten	174.09
21	Entire home/apt	Westerpark	173.79
17	Entire home/apt	Oud Noord	173.06
7	Entire home/apt	De Baarsjes / Oud West	170.99
3	Entire home/apt	Buitenveldert / Zuidas	168.80
20	Entire home/apt	Watergraafsmeer	167.11
12	Entire home/apt	Noord Oost	165.25
18	Entire home/apt	Oud Oost	159.89
16	Entire home/apt	Osdorp	158.93
19	Entire home/apt	Slotervaart	155.40
15	Entire home/apt	Oostelijk Havengebied / Indische Buurt	154.27
13	Entire home/apt	Noord West	153.09



From the above barplot we can see that 'Entire home/apt' room type at a 'Centrum West' has a highest price 228.97 and the 'Shared room' at the 'Slotervaar' has a lowest price 24.33 and also the location 'Centrum West' has a highes average price for all the three room types.

6. Preference of the guests for Room Type:



- Preference of the guests for Room Type

In [37]:
```
room_count=airbnb['room_type'].value_counts().reset_index()
room_count.columns=['room type','count']
room_count
```

Out[37]:

	room type	count
0	Entire home/apt	14978
1	Private room	3682
2	Shared room	63

Conclusion:

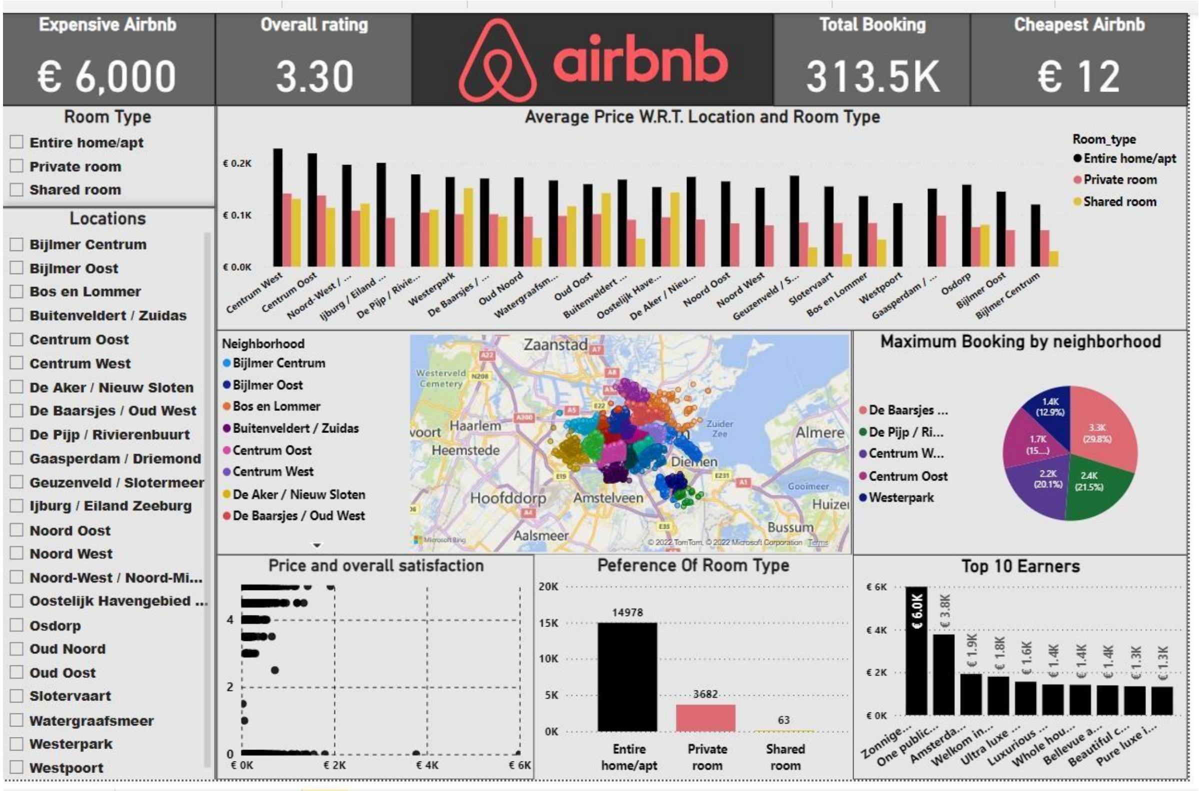
From the above visualization we can clearly see that the most prefered room type by the guests is Entire home/apt and the less prefered room type is shared room and private room.

From the above visualization we can clearly see that the most prefered room type by the guests is Entire home/apt and the less prefered room type is shared room and private room.

3. Business Intelligence Tool

- **Power BI :**

Creating power bi report for better understanding of dataset and for the stakeholders to understand the data in a better way and solving the business problems and taking a right decisions for increasing the profitability.



Expensive Airbnb
€ 6,000

Overall rating
3.30

Total Booking
313.5K

Cheapest Airbnb
€ 12

Room Type
- □ Entire home/apt
- □ Private room
- □ Shared room

Locations
- □ Bijlmer Centrum
- □ Bijlmer Oost
- □ Bos en Lommer
- □ Buitenveldert / Zuidas
- □ Centrum Oost
- □ Centrum West
- □ De Aker / Nieuw Sloten
- □ De Baarsjes / Oud West
- □ De Pijp / Rivierenbuurt
- □ Gaasperdam / Driemond
- □ Geuzenveld / Slotermeer
- □ Ijburg / Eiland Zeeburg
- □ Noord Oost
- □ Noord West
- □ Noord-West / Noord-Mi...
- □ Oostelijk Havengebied ...
- □ Osdorp
- □ Oud Noord
- □ Oud Oost
- □ Slotervaart
- □ Watergraafsmeer
- □ Westerpark
- □ Westpoort