ÇANKAYA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CENG 474

Introduction to Data Science

Machine Learning with Airbnb Istanbul Data

Merve KARAKAYA

Machine Learning with Airbnb Istanbul Data

Overview

In this project, machine learning operations were wanted to be done. Accordingly, exploratory data analysis and visualization were performed first. The preliminary stages for model development were made in order. After this stage, the model will be developed using the cleaned data. Classification models will be used for this.

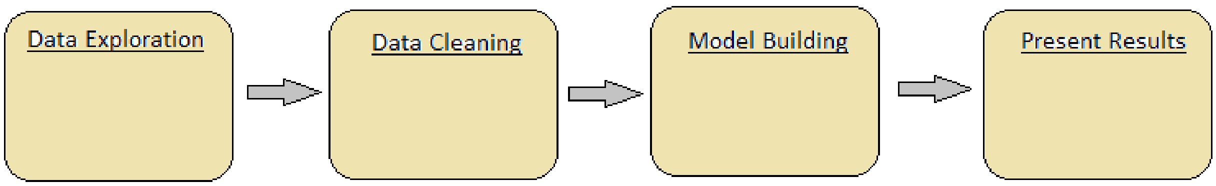

Figure 1: Machine learning process

Group members

- Merve Karakaya:

  Data Exploration
  Data Cleaning
  Model Building
  Present Results

Project Data Set

The data required for the project is from kaggle.com. This data is Airbnb Istanbul.[2]

Literature

Airbnb is a platform for organizing or offering accommodation. It is American vacation rental online marketplace company based in San Francisco, California, United States. Airbnb offers an opportunity to stay for those who prefer home to a hotel. Due to Airbnb's unique nature, hosting pricing strategies are very different from the traditional hospitality industry. For example, price determination criteria in hotels may not have a sufficient effect on Airbnb prices. Thus, there are quite different criteria in determining the prices in Airbnb compared to traditional accommodation types.

With this project, I wanted to examine the Airbnb pricing determinants. Therefore, I chose Airbnb Istanbul data which is has most lists in Turkey. Before working on this data, I did a literature review. The study I examined as a result of this research is briefly described below.[1]

The dataset is taken from kaggle.com. This dataset contains 16251 rows and 16 columns. Some rows contain missing values. Therefore, these missing values must be corrected or cleared. For example, there is no data available in the *neighbourhood_group* column. This column needs to be removed. Additionally, 8484 missing values appear in the *last_review* and *reviews_per_month* columns. Thus, it will be sufficient to fill in these columns with a value of zero.

The libraries used in this study are as follows: Pandas library was used for data analysis. Seaborn and Matplotlib libraries were used for visualization. Folium library was used to show the visualization on the map. Shortly, in this study, exploratory data analysis and visualization were performed. Firstly, analyzes were made about the data set. Later, the missing values in the data set were corrected. Visualizations were used while doing these operations. According to these, the results of the study are as follows:[1]

- The neighborhoods of the most frequently found lists are from Beyoglu, Sisli, Fatih, Kadiköy and Besiktas.
- The most listed room type is private room with number of 8565. The Entire home/apt and shared room follow with 7191 and 495 numbers.
- The host with id "21907588" has the most listings with 77 number of listings.
- Average price is highest in Kücükcekmece neighborhood. The average daily price for Kücükcekmece is 1263 TL.
- Pendik has the lowest average price with 153 TL per day.
- The most expensive advertisement is the "3 Rooms 1 Salon - Grand Holiday Istanbul" special room in Kücükcekmece with 59561 TL a day.
- According to the map, 90% of the 10 most expensive ads are located on the European side of Istanbul.
- The most reviewed list is the special room "Atatürk Airport 5 minutes" prepared by Melik Fırat.
- The daily average list price is 207.8 TL.

Goals

The aim of this project is to examine the effects of airbnb pricing criteria. Machine learning classification models will be used to see which criteria have an effect on price. The expected results are as follows:

The most effective features on the price are the neighborhoods and the types of rooms. Pricing is expected to be high in the most preferred luxury districts of Istanbul, which is related to the number of lists belonging to that neighborhood. Private room types are also expected to have the highest price.

Specifications

Exploratory Data Analysis and Visualizations

First of all, it is necessary to examine the story and structure of the data set for exploratory data analysis. Therefore, the data set should be read and stored in a data frame. Accordingly, the first and last five data observation outputs are as follows:

```
# Reading data set
airbnb = pd.read_csv("AirbnbIstanbul.csv")
#First 5 observation displays
print(airbnb.head(5))
```

```
      id                              name  host_id host_name  \
0   4826                         The Place     6603      Kaan
1  20815    The Bosphorus from The Comfy Hill    78838    Gülder
2  25436  House for vacation rental furnutare   105823     Yesim
3  27271       LOVELY APT. IN PERFECT LOCATION   117026     Mutlu
4  28277        Duplex Apartment with Terrace   121607      Alen

   neighbourhood_group neighbourhood  latitude  longitude        room_type  \
0                  NaN       Uskudar  41.05650   29.05367  Entire home/apt
1                  NaN      Besiktas  41.06984   29.04545  Entire home/apt
2                  NaN      Besiktas  41.07731   29.03891  Entire home/apt
3                  NaN       Beyoglu  41.03220   28.98216  Entire home/apt
4                  NaN         Sisli  41.04471   28.98567  Entire home/apt

   price  minimum_nights  number_of_reviews last_review  reviews_per_month  \
0    554               1                  1  2009-06-01               0.01
1    100              30                 41  2018-11-07               0.38
2    211              21                  0         NaN                NaN
3    237               5                  2  2018-05-04               0.04
4    591               3                  0         NaN                NaN

   calculated_host_listings_count  availability_365
0                               1               365
1                               2                49
2                               1                83
3                               1               228
4                              13               356
```

Figure 2: First five data observation

```
#last 5 observation displays
airbnb.tail()
```

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_c
16246	32452512	Best place of town	29568076	Antonio	NaN	Sisli	41.04775	28.99283	Entire home/apt	248	1	
16247	32453285	luxury flat in city center atiye str nisantası	29568076	Antonio	NaN	Sisli	41.04775	28.99283	Entire home/apt	248	1	
16248	32453323	Double Room	228430419	Saladin	NaN	Fatih	41.00435	28.97692	Private room	237	2	
16249	32455952	Cozy room in charming home at the heart of Bey...	108703005	Pelin	NaN	Beyoglu	41.03118	28.97837	Private room	53	3	
16250	32457561	Perfect view with comfortable room	25991676	Uğur	NaN	Kadikoy	40.99467	29.05423	Private room	100	1	

Figure 3: Last five data observation

Then, the structure of this data set should be examined in detail. Accordingly, the data set consists of 16251 rows and 16 columns. The list of properties in these columns is as follows:

```
#Data set has 16251 samples(rows) and 16 features(columns)
airbnb.shape
```

```
(16251, 16)
```

```
#displaying these columns
airbnb.columns
```

```
Index(['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'],
      dtype='object')
```

Figure 4: Data set structure

The types of data included in these features and the number of empty data in them are very important for data analysis. For this, the information of the data set should be examined in detail. According to the information in the figure below, *name*, *host_name* , *neighborhood_group*, *last_review*, *reviews_per_month* columns contain missing values. In addition, the types of data included in these columns are also listed.

```
airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16251 entries, 0 to 16250
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              16251 non-null  int64  
 1   name                            16160 non-null  object 
 2   host_id                         16251 non-null  int64  
 3   host_name                       16244 non-null  object 
 4   neighbourhood_group             0 non-null      float64
 5   neighbourhood                   16251 non-null  object 
 6   latitude                        16251 non-null  float64
 7   longitude                       16251 non-null  float64
 8   room_type                       16251 non-null  object 
 9   price                           16251 non-null  int64  
 10  minimum_nights                  16251 non-null  int64  
 11  number_of_reviews               16251 non-null  int64  
 12  last_review                     7767 non-null   object 
 13  reviews_per_month               7767 non-null   float64
 14  calculated_host_listings_count  16251 non-null  int64  
 15  availability_365                16251 non-null  int64  
dtypes: float64(4), int64(7), object(5)
memory usage: 2.0+ MB
```

Figure 5: Detailed data set information

Statistical analysis of this information is also helpful for data analysis. In statistical analysis, many results such as the frequencies of the data, the most repetitive samples, mean, standard deviation, quantile values, min and max samples etc. are obtained. According to these results, more information about the data can be obtained. For example, there are 39 different neighborhoods in total, the most preferred neighborhood is Beyoglu, the most preferred room type is private etc. Figure 6 includes these results in detail.

```
#Slightly more detailed descriptive statistics for features
airbnb.describe(include = "all").T
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
id	16251	NaN	NaN	NaN	1.88564e+07	1.0548e+07	4826	8.50098e+06	2.16198e+07	2.87022e+07	3.24576e+07
name	16160	15494	İstanbul Birden fazla bölümden oluşan bina	46	NaN	NaN	NaN	NaN	NaN	NaN	NaN
host_id	16251	NaN	NaN	NaN	8.88871e+07	8.16211e+07	6603	1.78823e+07	5.21074e+07	1.68135e+08	2.43734e+08
host_name	16244	3797	Mehmet	220	NaN	NaN	NaN	NaN	NaN	NaN	NaN
neighbourhood_group	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
neighbourhood	16251	39	Beyoglu	4245	NaN	NaN	NaN	NaN	NaN	NaN	NaN
latitude	16251	NaN	NaN	NaN	41.0265	0.0431984	40.8147	41.0044	41.0314	41.0478	41.4144
longitude	16251	NaN	NaN	NaN	28.9854	0.114358	28.032	28.9741	28.9843	29.0224	29.9078
room_type	16251	3	Private room	8565	NaN	NaN	NaN	NaN	NaN	NaN	NaN
price	16251	NaN	NaN	NaN	354.724	1428.94	0	105	190	327	59561
minimum_nights	16251	NaN	NaN	NaN	4.69294	28.9161	1	1	1	2	1125
number_of_reviews	16251	NaN	NaN	NaN	7.18676	21.4396	0	0	0	4	307
last_review	7767	1160	2019-02-10	169	NaN	NaN	NaN	NaN	NaN	NaN	NaN
reviews_per_month	7767	NaN	NaN	NaN	0.914766	1.08691	0.01	0.18	0.52	1.19	12
calculated_host_listings_count	16251	NaN	NaN	NaN	4.10381	7.64823	1	1	1	4	77
availability_365	16251	NaN	NaN	NaN	249.495	136.153	0	101	340	365	365

Figure 6: Descriptive statics results

It is very useful to benefit from visualization as it is a preliminary information when analyzing data. Therefore, the visuals below will provide a better understanding of the data set used in this project. In the figure 7, when examining the Airbnb lists, there is the relation of the room type, which is one of the most selected filters, with the price. Accordingly, the price of private rooms is higher than other room types.

```
#Relationship of room type to pricing
sns.scatterplot(x = "room_type", y = "price", data = airbnb);
```

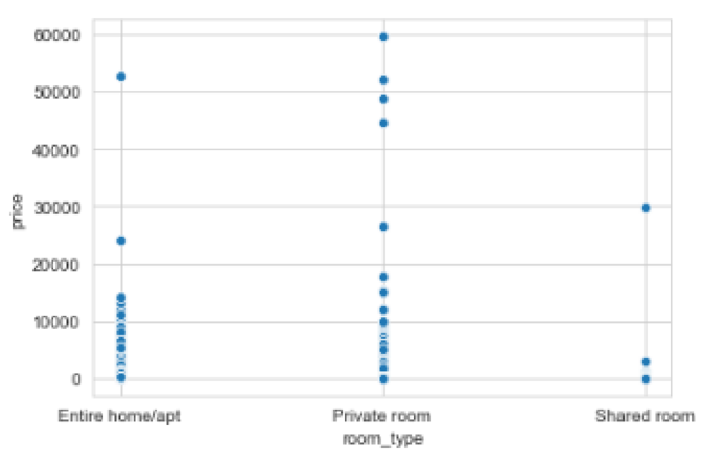

Figure 7: Relationship room type to price

One of the other important filters chosen during the listing is the neighborhood criterion. In figure 8, the relationship between this *neighbourhood* criterion and the *number_of_reviews* is given. These examinations are divided depending on the *room_type*. Accordingly, the most reviewed neighborhoods are Beyoglu and Fatih. In addition, the relationship of this *neighbourhood* criterion with the *price* is given in the figure 9 below. These examinations are likewise divided according to the type of room.

```python
#Investigation of the number of reviews with the neighborhood according to room types
sns.scatterplot(x = "number_of_reviews", y = "neighbourhood", hue = "room_type", style = "room_type", data = airbnb);
```

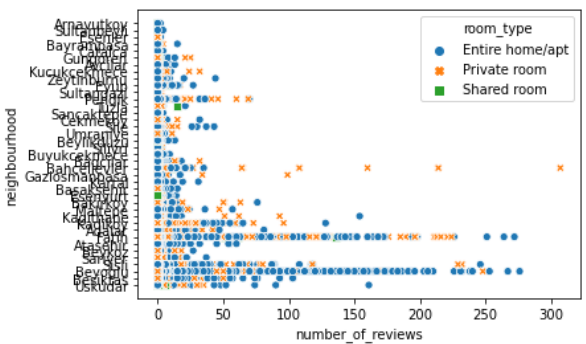

Figure 8: Investigation of the number of reviews with neighborhood according to room types

```python
#Investigation of the price with the neighborhood according to room types
sns.scatterplot(x = "price", y = "neighbourhood", hue = "room_type", style = "room_type", data = airbnb);
```

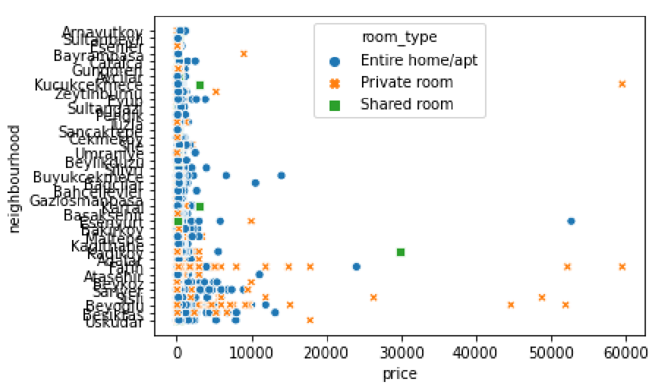

Figure 9: Investigation of the price with neighborhood according to room types

The relationships of these features with each other are very important to make inferences from the data set. For this, the correlation function can be used. According to these correlation values, for example, if it is -1 or +1, it means that the similarity of the two features is quite high, but if it is 0.5 or -0.5, the correlation of the two features is not good. Additionally, heat map provides a clear result to see these relationships visually.

```
#relationships of features with each other
airbnb.corr()
```

	id	host_id	neighbourhood_group	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
id	1.000000	0.679662	NaN	-0.021689	-0.025538	0.005983	-0.025142	-0.269707	0.269992	-0.030279	-0.169436
host_id	0.679662	1.000000	NaN	-0.014529	-0.058144	0.009993	-0.027572	-0.205073	0.147374	-0.103338	-0.123720
neighbourhood_group	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
latitude	-0.021689	-0.014529	NaN	1.000000	-0.184363	0.032536	0.006076	-0.025143	-0.039408	0.001483	-0.001116
longitude	-0.025538	-0.058144	NaN	-0.184363	1.000000	-0.022089	-0.006377	-0.001883	-0.015978	-0.033867	-0.034483
price	0.005983	0.009993	NaN	0.032536	-0.022089	1.000000	0.016585	-0.019262	-0.032012	0.030100	0.047015
minimum_nights	-0.025142	-0.027572	NaN	0.006076	-0.006377	0.016585	1.000000	-0.015149	-0.036223	-0.020916	0.015297
number_of_reviews	-0.269707	-0.205073	NaN	-0.025143	-0.001883	-0.019262	-0.015149	1.000000	0.496183	0.174663	0.043230
reviews_per_month	0.269992	0.147374	NaN	-0.039408	-0.015978	-0.032012	-0.036223	0.496183	1.000000	0.051228	-0.063728
calculated_host_listings_count	-0.030279	-0.103338	NaN	0.001483	-0.033867	0.030100	-0.020916	0.174663	0.051228	1.000000	0.173068
availability_365	-0.169436	-0.123720	NaN	-0.001116	-0.034483	0.047015	0.015297	0.043230	-0.063728	0.173068	1.000000

Figure 10: Correlation function results

According to the heat map in figure 11, the relationships with the correlation values are shown in different colors. For example, the highest correlation value in diagonal appears in blue because this means that the two compared features are the same.

```
#Visual representation of the properties' relationships with each other
corr=airbnb.corr()
h=sns.heatmap(corr,vmin=-1,vmax=1,center=0,cmap=sns.diverging_palette(20,220,n=200),square=True)
```

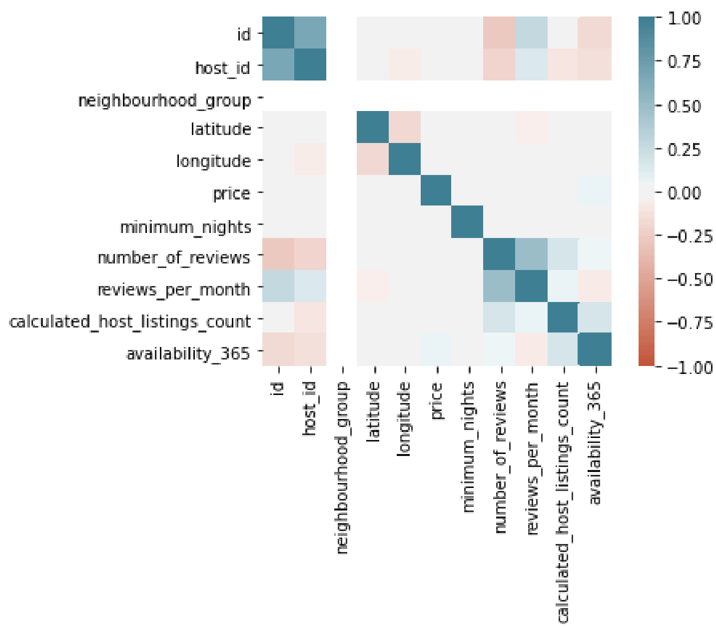

Figure 11: Heat Map results

In addition, according to this data, the relationship between the list numbers of the host ids on the Airbnb platform can be observed. The pricing of these hosts according to their ids can also be observed. Accordingly, the host id with the most listings is 21907588. On average, the highest pricing host id is 161593238. These results are shown in the two figures below.[1]

```
#The number of airbnbs lists by hosts with ids (for top 10 observation)
chart_host = airbnb["host_id"].value_counts().head(10).plot.bar()
chart_host.set_title(" Hosts with Most Listings")
chart_host.set_xlabel("Host ID")
chart_host.set_ylabel("Number of Listings")
```

```
Text(0, 0.5, 'Number of Listings')
```

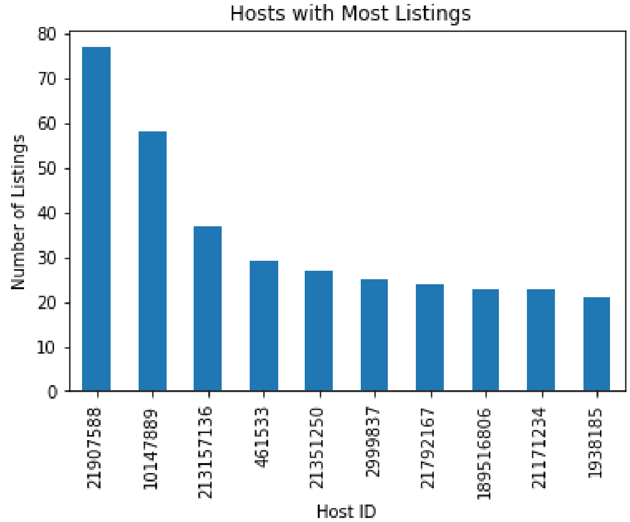

Figure 12: The number of Airbnb lists by hosts with ids

```
#Average price by host_id (for top 10 observation)
neighbourhood_price = airbnb.groupby("host_id")["price"].agg(['mean'])
chart_avg_price = neighbourhood_price.sort_values(by='mean', ascending=False).head(10).plot.bar()
chart_avg_price.set_ylabel('Price TRY')
chart_avg_price.set_xlabel('Host ID')
chart_avg_price.set_title("Average Price of Host IDs")
```

```
Text(0.5, 1.0, 'Average Price of Host IDs')
```

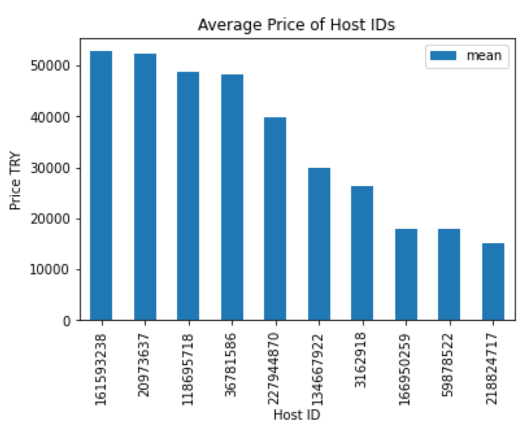

Figure 13: Average price by host_id

The same relations above can be applied to the *neighbourhood* feature. Accordingly, the neighborhood with the highest listings is Beyoglu, and the neighborhood with the highest average price is Kucukcekmece. The visual outputs of these are as follows:[1]

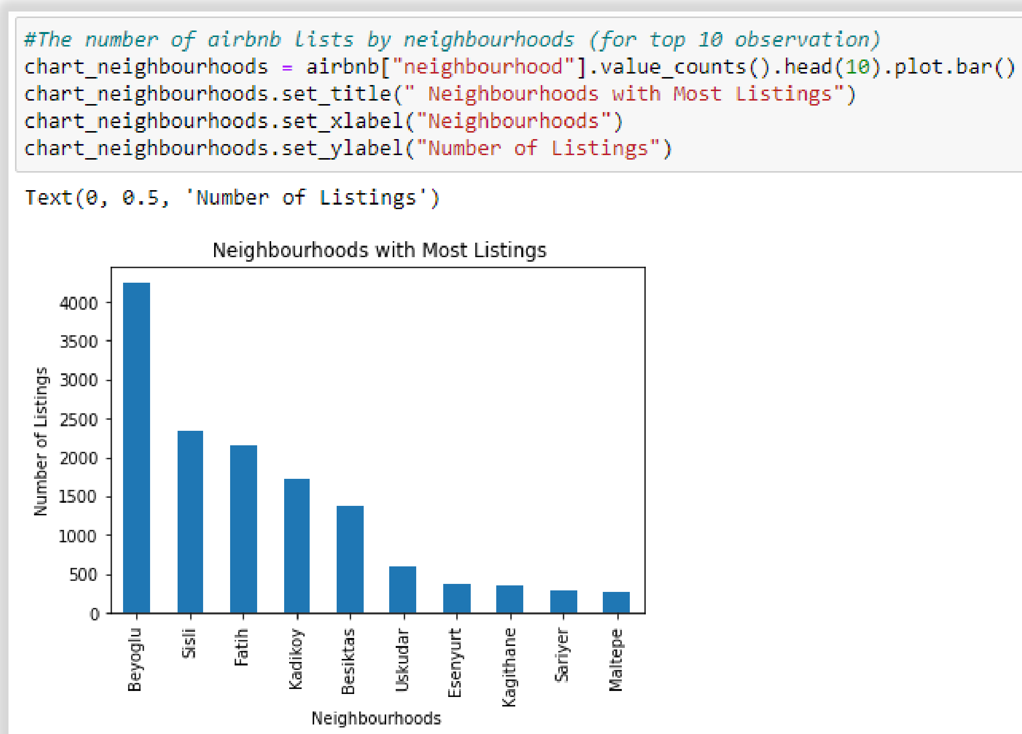

Figure 14: The number of Airbnb lists by neighbourhoods

```
#Average price by neighborhood (for top 10 observation)
neighbourhood_price = airbnb.groupby("neighbourhood")["price"].agg(['mean'])
chart_avg_price = neighbourhood_price.sort_values(by='mean', ascending=False).head(10).plot.bar()
chart_avg_price.set_ylabel('Price TRY')
chart_avg_price.set_xlabel('Neighbourhoods')
chart_avg_price.set_title("Average Price of Neighbourhoods")
```

```
Text(0.5, 1.0, 'Average Price of Neighbourhoods')
```

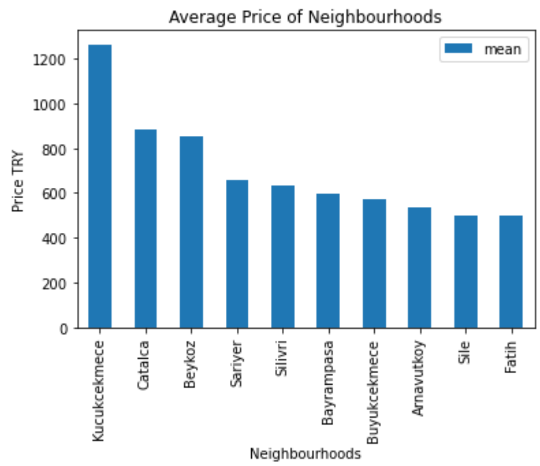

Figure 15: Average price by neighborhood

After accessing more detailed information about the features in the data using these visuals, it should be cleaned. First of all, it should be checked whether there are missing values in the data, and if so, their numbers should be extracted according to the features. It is noticed that the *neighbourhood_group* column is completely empty.

```
#Are there any missing observations (values)
airbnb.isnull().values.any()
```

```
True
```

```
#In which variable how many
airbnb.isnull().sum()
```

```
id                                    0
name                                 91
host_id                               0
host_name                             7
neighbourhood_group               16251
neighbourhood                         0
latitude                              0
longitude                             0
room_type                             0
price                                 0
minimum_nights                        0
number_of_reviews                     0
last_review                        8484
reviews_per_month                  8484
calculated_host_listings_count        0
availability_365                      0
dtype: int64
```

Figure 16: The Number of missing values in data

In addition, it is very easy to visually see how much these missing values are, thanks to the heat map.The white colors here represent the missing values.Accordingly, there are quite a lot of missing values in the two features which is *last_review* and *reviews_per_month.*

```
#Some features have missing data:
sns.heatmap(airbnb.isnull(), yticklabels = False, cbar = False)
#White color indicates missing values
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x22045e295b0>
```

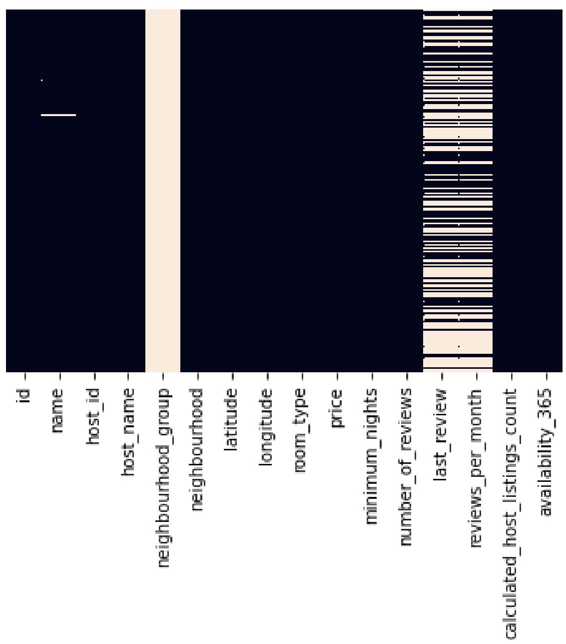

Figure 17: Heatmap results for missing values

For data cleaning process, operations should be done on missing values. Columns that are completely empty or contain a large number of empty values should be dropped. Unnecessary columns for the analyzed data can also be dropped. Another method for cleaning is to fill in and correct the blank values.

According to these operations, the *neighbourhood_group* column that is empty for Airbnb Istanbul data should be dropped. Also, the *last_review* column, which contains a large number of missing values and is unnecessary for the review in this project, should be dropped.

```
airbnb.drop('neighbourhood_group',axis=1,inplace=True)
airbnb.drop('last_review',axis=1,inplace=True)
```

```
#Checking drop process
airbnb.columns
```

```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood', 'latitude',
       'longitude', 'room_type', 'price', 'minimum_nights',
       'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365'],
      dtype='object')
```

Figure 18: Dropping features which have missing values

The empty values in the *reviews_per_month* column, which are required for the price review in the project, but contain missing values, can be filled with 0, because emptying may mean that it has never been examined. Finally, after the empty values remaining in the *name* column are dropped, the data cleaning process ends.

```
#Correction function of reviews_per_month data
def impute_reviews_per_month(cols):
    reviews = cols[0]

    if pd.isnull(reviews):
        return 0 #that means it has never been examined
    else:
        return reviews
```

```
#Function call
airbnb['reviews_per_month']=airbnb[['reviews_per_month']].apply(impute_reviews_per_month, axis =1 )
```

```
#correction for missing data in name
airbnb.dropna(inplace = True)
```

Figure 19 : Correction reviews_per_month feature and drop missing values in the name feature

As a result of all these cleaning processes, the output is as follows:

```
sns.heatmap(airbnb.isnull(), yticklabels = False, cbar = False)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x22045e0b760>
```

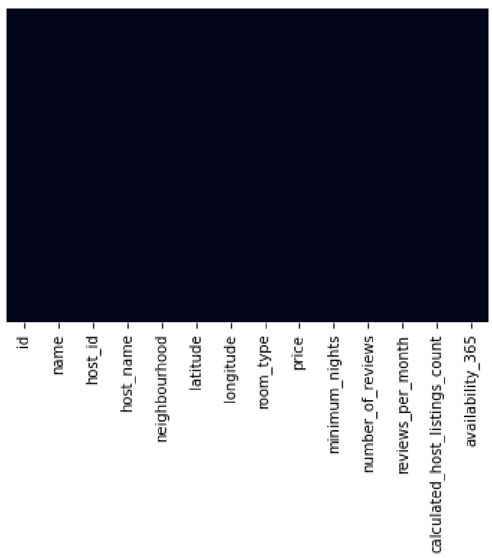

Figure 20: Heatmap results after data cleaning

```
airbnb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16155 entries, 0 to 16250
Data columns (total 14 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              16155 non-null  int64  
 1   name                            16155 non-null  object 
 2   host_id                         16155 non-null  int64  
 3   host_name                       16155 non-null  object 
 4   neighbourhood                   16155 non-null  object 
 5   latitude                        16155 non-null  float64
 6   longitude                       16155 non-null  float64
 7   room_type                       16155 non-null  object 
 8   price                           16155 non-null  int64  
 9   minimum_nights                  16155 non-null  int64  
 10  number_of_reviews               16155 non-null  int64  
 11  reviews_per_month               16155 non-null  float64
 12  calculated_host_listings_count  16155 non-null  int64  
 13  availability_365                16155 non-null  int64  
dtypes: float64(3), int64(7), object(4)
memory usage: 1.8+ MB
```

Figure 21: Number of non-null values after data cleaning

After the data cleaning process, the categorical data can be converted into numerical data and the preparation stage for model development can be done. The categorical features in Airbnb Istanbul data are as follows:

```
#Selecting categorical data
a=airbnb.copy()
categorical_a = a.select_dtypes(include = ["object"])
categorical_a.head(5)
```

	name	host_name	neighbourhood	room_type
0	The Place	Kaan	Uskudar	Entire home/apt
1	The Bosphorus from The Comfy Hill	Gülder	Besiktas	Entire home/apt
2	House for vacation rental furnutare	Yesim	Besiktas	Entire home/apt
3	LOVELY APT. IN PERFECT LOCATION	Mutlu	Beyoglu	Entire home/apt
4	Duplex Apartment with Terrace	Alen	Sisli	Entire home/apt

Figure 22: Categorical Features in Airbnb Istanbul

The distribution of classes in the *neighbourhood*, which is one of these categorical features, is as follows:

```
#Visual representation
categorical_a["neighbourhood"].value_counts().plot.barh();
```

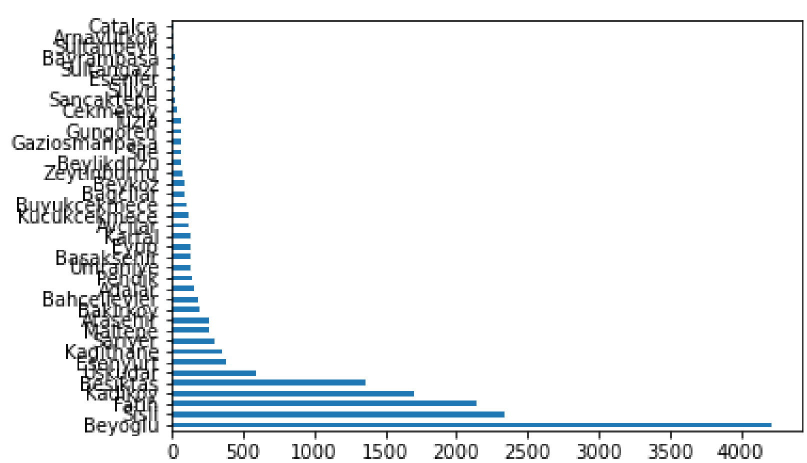

Figure 23: neighbourhood feature representation

Likewise, one of the categorical features is the *room_type*:

```
#Visual representation
categorical_a["room_type"].value_counts().plot.barh();
```

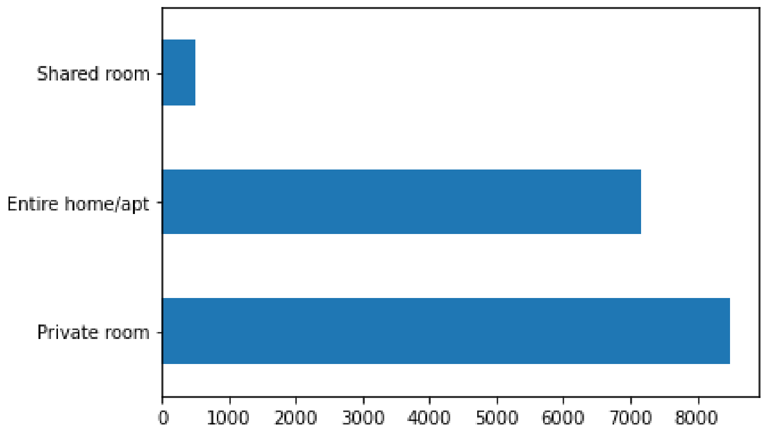

Figure 24: room_type feature representation

Such categorical data can be easily converted into numerical data as 1 and 0 with one hot encoding, and new edited features can be added to the data frame. The demonstration of these operations is as follows:

```
#Converting categorical variables into "dummy" or indicator variables
Room_type =pd.get_dummies(airbnb['room_type'])
Room_type.head()
```

	Entire home/apt	Private room	Shared room
0	1	0	0
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0

Figure 25: Converting categorical variables to numerical variable

```python
Neighbourhood=pd.get_dummies(airbnb['neighbourhood'])
Neighbourhood.head()
```

	Adalar	Arnavutkoy	Atasehir	Avcilar	Bagcilar	Bahcelievler	Bakirkoy	Basaksehir	Bayrampasa	Besiktas	...	Sariyer	Sile	Silivri	Sisli	Sultanbeyli	Sult
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	1	...	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	

5 rows × 39 columns

Figure 26: Converting categorical variables to numerical variable

```python
# Add new dummy columns to data frame
airbnb = pd.concat([airbnb,Neighbourhood,Room_type],axis = 1)
```

```python
#airbnb.columns
airbnb.head(2)
```

	id	name	host_id	host_name	neighbourhood	latitude	longitude	room_type	price	minimum_nights	...	Sisli	Sultanbeyli	Sultangazi	Tuzla	Ur
0	4826	The Place	6603	Kaan	Uskudar	41.05650	29.05367	Entire home/apt	554	1	...	0	0	0	0	
1	20815	The Bosphorus from The Comfy Hill	78838	Gülder	Besiktas	41.06984	29.04545	Entire home/apt	100	30	...	0	0	0	0	

2 rows × 56 columns

Figure 27: Adding new features to data frame

```python
#Drop the old categorical columns
airbnb.drop(['neighbourhood', 'room_type'], axis = 1, inplace = True)
airbnb.head(2)
```

	id	name	host_id	host_name	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	...	Sisli	Sultanbeyli	Sultanga
0	4826	The Place	6603	Kaan	41.05650	29.05367	554	1	1	0.01	...	0	0	
1	20815	The Bosphorus from The Comfy Hill	78838	Gülder	41.06984	29.04545	100	30	41	0.38	...	0	0	

2 rows × 54 columns

Figure 28:Dropping the old categorical columns in data frame

```python
#Checking
airbnb.columns
```

```
Index(['id', 'name', 'host_id', 'host_name', 'latitude', 'longitude', 'price',
       'minimum_nights', 'number_of_reviews', 'reviews_per_month',
       'calculated_host_listings_count', 'availability_365', 'Adalar',
       'Arnavutkoy', 'Atasehir', 'Avcilar', 'Bagcilar', 'Bahcelievler',
       'Bakirkoy', 'Basaksehir', 'Bayrampasa', 'Besiktas', 'Beykoz',
       'Beylikduzu', 'Beyoglu', 'Buyukcekmece', 'Catalca', 'Cekmekoy',
       'Esenler', 'Esenyurt', 'Eyup', 'Fatih', 'Gaziosmanpasa', 'Gungoren',
       'Kadikoy', 'Kagithane', 'Kartal', 'Kucukcekmece', 'Maltepe', 'Pendik',
       'Sancaktepe', 'Sariyer', 'Sile', 'Silivri', 'Sisli', 'Sultanbeyli',
       'Sultangazi', 'Tuzla', 'Umraniye', 'Uskudar', 'Zeytinburnu',
       'Entire home/apt', 'Private room', 'Shared room'],
      dtype='object')
```

Figure 29: Checking new features

In addition, categorical data can be observed separately, as well as numerical data can be observed in this way. A statistical analysis that gives a detailed result for numerical data can be made. These results are given below:

```python
#Selecting numerical data
a=airbnb.copy()
numerical_a = a.select_dtypes(include = ["float64", "int64"])
numerical_a.head()
```

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	calculated_host_listings_count	availability_365
0	4826	6603	41.05650	29.05367	554	1	1	0.01	1	365
1	20815	78838	41.06984	29.04545	100	30	41	0.38	2	49
2	25436	105823	41.07731	29.03891	211	21	0	0.00	1	83
3	27271	117026	41.03220	28.98216	237	5	2	0.04	1	228
4	28277	121607	41.04471	28.98567	591	3	0	0.00	13	356

Figure 30: Numerical Features in Airbnb Istanbul

```python
#Descriptive statistics for price feature
numerical_a["price"].describe()
```

```
count    16155.000000
mean       355.289941
std       1433.062707
min          0.000000
25%        105.000000
50%        185.000000
75%        327.000000
max      59561.000000
Name: price, dtype: float64
```

```python
#Slightly more detailed descriptive statistics for price feature
print("Mean: " + str(numerical_a["price"].mean()))
print("Number of Full Observations: " + str(numerical_a["price"].count()))
print("Max Value: " + str(numerical_a["price"].max()))
print("Min Value: " + str(numerical_a["price"].min()))
print("Median: " + str(numerical_a["price"].median()))
print("Standart Deviation: " + str(numerical_a["price"].std()))
```

```
Mean: 355.28994119467654
Number of Full Observations: 16155
Max Value: 59561
Min Value: 0
Median: 185.0
Standart Deviation: 1433.0627072935586
```

Figure 31: Results of statistical analysis of price

To sum up, with exploratory data analysis and visualization of these analyzes, very detailed information is obtained about the data to be given to the model. The data is better understood so that the next processes can be done more accurately. According to the information, the data is cleaned and the preparation is made for the development of models. After this stage, models are developed according to the data prepared and how the predicts of these models are evaluated.

References

[1] https://www.kaggle.com/fehmifratpolat/istanbul-airbnb-data-analysis-and-visualization

[2] https://www.kaggle.com/kavanozkafa/airbnb-istanbul-dataset