The Perfect Vacation

A model to predict the rate of Airbnb's rental apartment

Background

As most of us were abroad, we all know that one of the biggest dilemmas is deciding **what is the best place to stay in during our vacation?**

In this project we will try to supply an ML model which predicting the rate of giving apartments, based on some of the apartment's features.

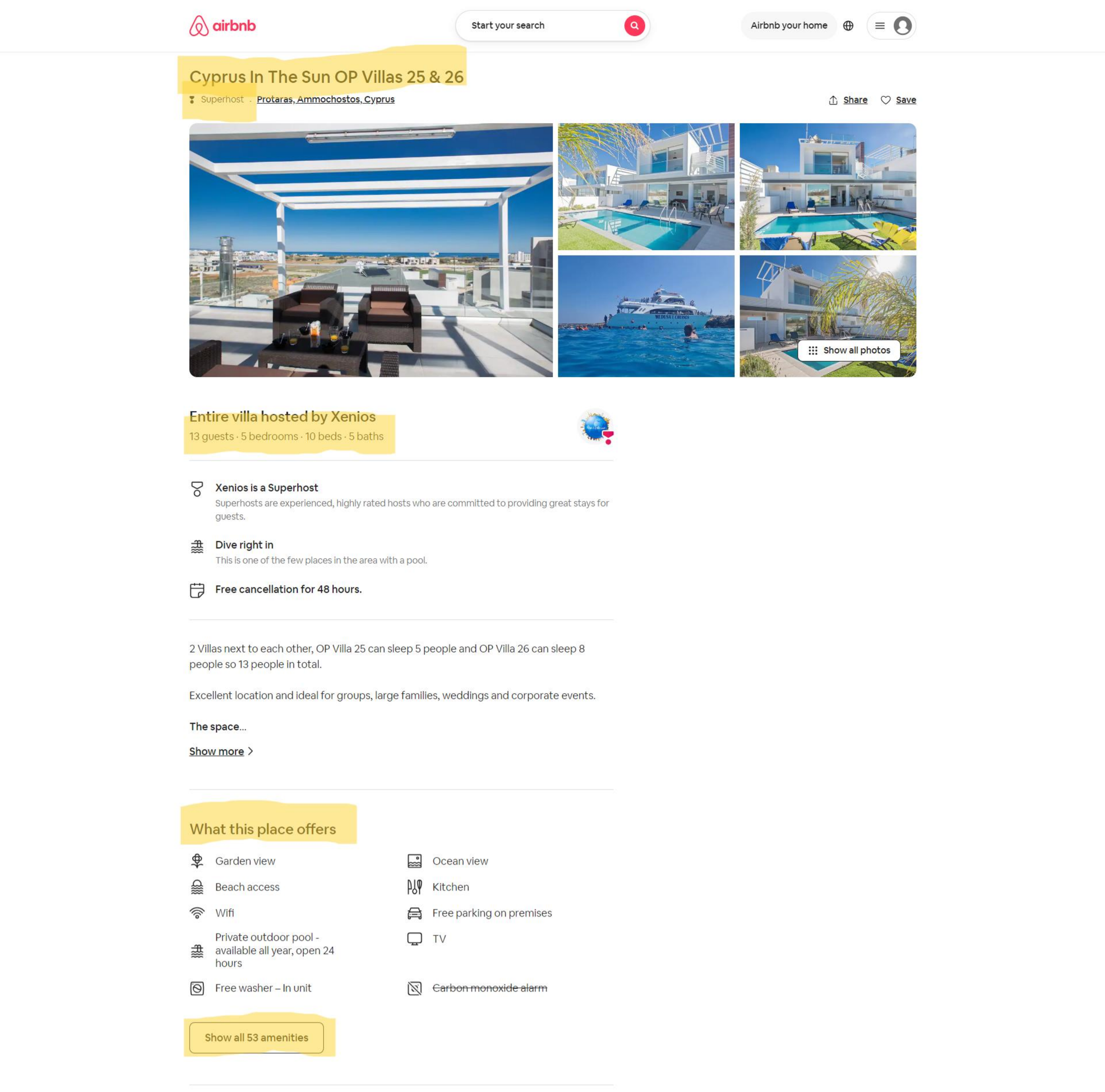
External Resources

Airbnb

Apartment page example

Example of rental apartment page as It show in Airbnb’s website.

We can see the yellow marks which is part of the main features we want to extract for each page



Cyprus In The Sun OP Villas 25 & 26

Superhost · Protaras, Ammochostos, Cyprus

Entire villa hosted by Xenios

13 guests · 5 bedrooms · 10 beds · 5 baths

Xenios is a Superhost

Superhosts are experienced, highly rated hosts who are committed to providing great stays for guests.

Dive right in

This is one of the few places in the area with a pool.

Free cancellation for 48 hours.

2 Villas next to each other, OP Villa 25 can sleep 5 people and OP Villa 26 can sleep 8 people so 13 people in total.

Excellent location and ideal for groups, large families, weddings and corporate events.

The space...

Show more >

What this place offers

Garden view

Beach access

Wifi

Private outdoor pool - available all year, open 24 hours

Free washer – In unit

Ocean view

Kitchen

Free parking on premises

TV

~~Carbon monoxide alarm~~

Show all 53 amenities

Stage 1: data acquisition

Crawling

using BeautifulSoup & Requests & Pandas

Extract all apartments information from various cities in USA

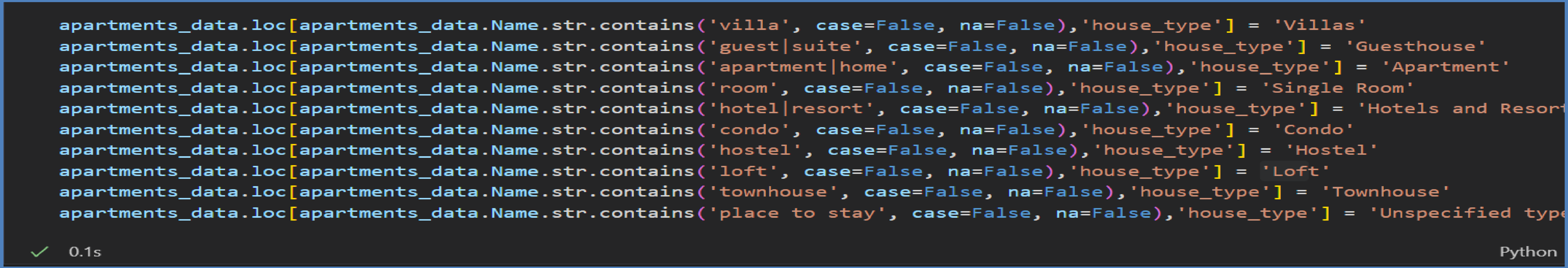
Extract :

- Location
- Name
- Number of rooms
- Max guest
- Binary features (Pets, Wifi, TV, Cooling, Heating etc.)
- Price
- Review counts
- Total rate

Stage 2: New Features Creation

For future use, we have created 2 new features in our dataset:

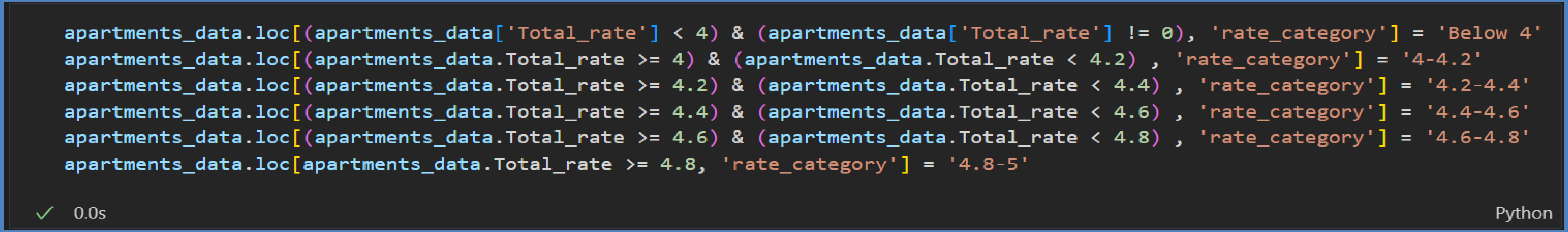
1. **“house_type”** feature



```
apartments_data.loc[apartments_data.Name.str.contains('villa', case=False, na=False),'house_type'] = 'Villas'
apartments_data.loc[apartments_data.Name.str.contains('guest|suite', case=False, na=False),'house_type'] = 'Guesthouse'
apartments_data.loc[apartments_data.Name.str.contains('apartment|home', case=False, na=False),'house_type'] = 'Apartment'
apartments_data.loc[apartments_data.Name.str.contains('room', case=False, na=False),'house_type'] = 'Single Room'
apartments_data.loc[apartments_data.Name.str.contains('hotel|resort', case=False, na=False),'house_type'] = 'Hotels and Resor
apartments_data.loc[apartments_data.Name.str.contains('condo', case=False, na=False),'house_type'] = 'Condo'
apartments_data.loc[apartments_data.Name.str.contains('hostel', case=False, na=False),'house_type'] = 'Hostel'
apartments_data.loc[apartments_data.Name.str.contains('loft', case=False, na=False),'house_type'] = 'Loft'
apartments_data.loc[apartments_data.Name.str.contains('townhouse', case=False, na=False),'house_type'] = 'Townhouse'
apartments_data.loc[apartments_data.Name.str.contains('place to stay', case=False, na=False),'house_type'] = 'Unspecified type
```

✓ 0.1s Python

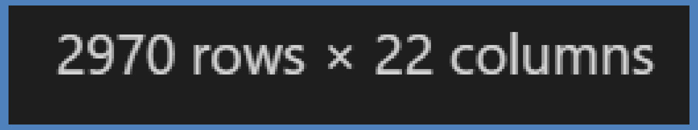
2. **“rate_category”** feature



```
apartments_data.loc[(apartments_data['Total_rate'] < 4) & (apartments_data['Total_rate'] != 0), 'rate_category'] = 'Below 4'
apartments_data.loc[(apartments_data.Total_rate >= 4) & (apartments_data.Total_rate < 4.2) , 'rate_category'] = '4-4.2'
apartments_data.loc[(apartments_data.Total_rate >= 4.2) & (apartments_data.Total_rate < 4.4) , 'rate_category'] = '4.2-4.4'
apartments_data.loc[(apartments_data.Total_rate >= 4.4) & (apartments_data.Total_rate < 4.6) , 'rate_category'] = '4.4-4.6'
apartments_data.loc[(apartments_data.Total_rate >= 4.6) & (apartments_data.Total_rate < 4.8) , 'rate_category'] = '4.6-4.8'
apartments_data.loc[apartments_data.Total_rate >= 4.8, 'rate_category'] = '4.8-5'
```

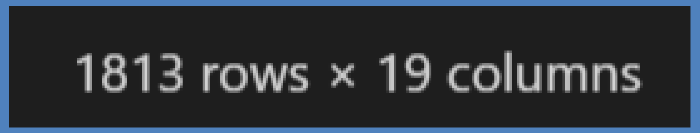
✓ 0.0s Python

Stage 3: Data cleanup



2970 rows × 22 columns

→



1813 rows × 19 columns

Binary classification of all True/False attributes

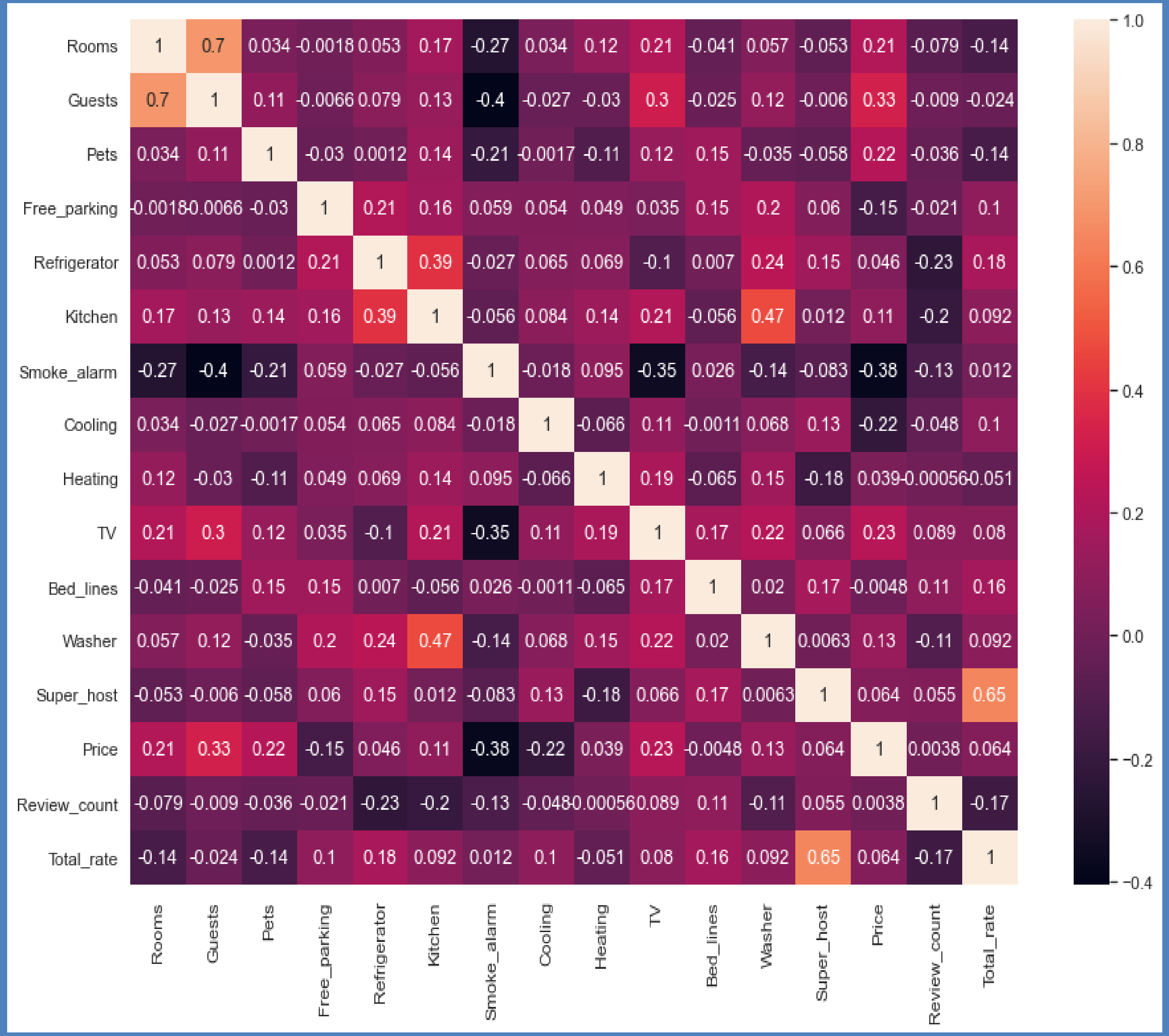
Checking and removing duplicates

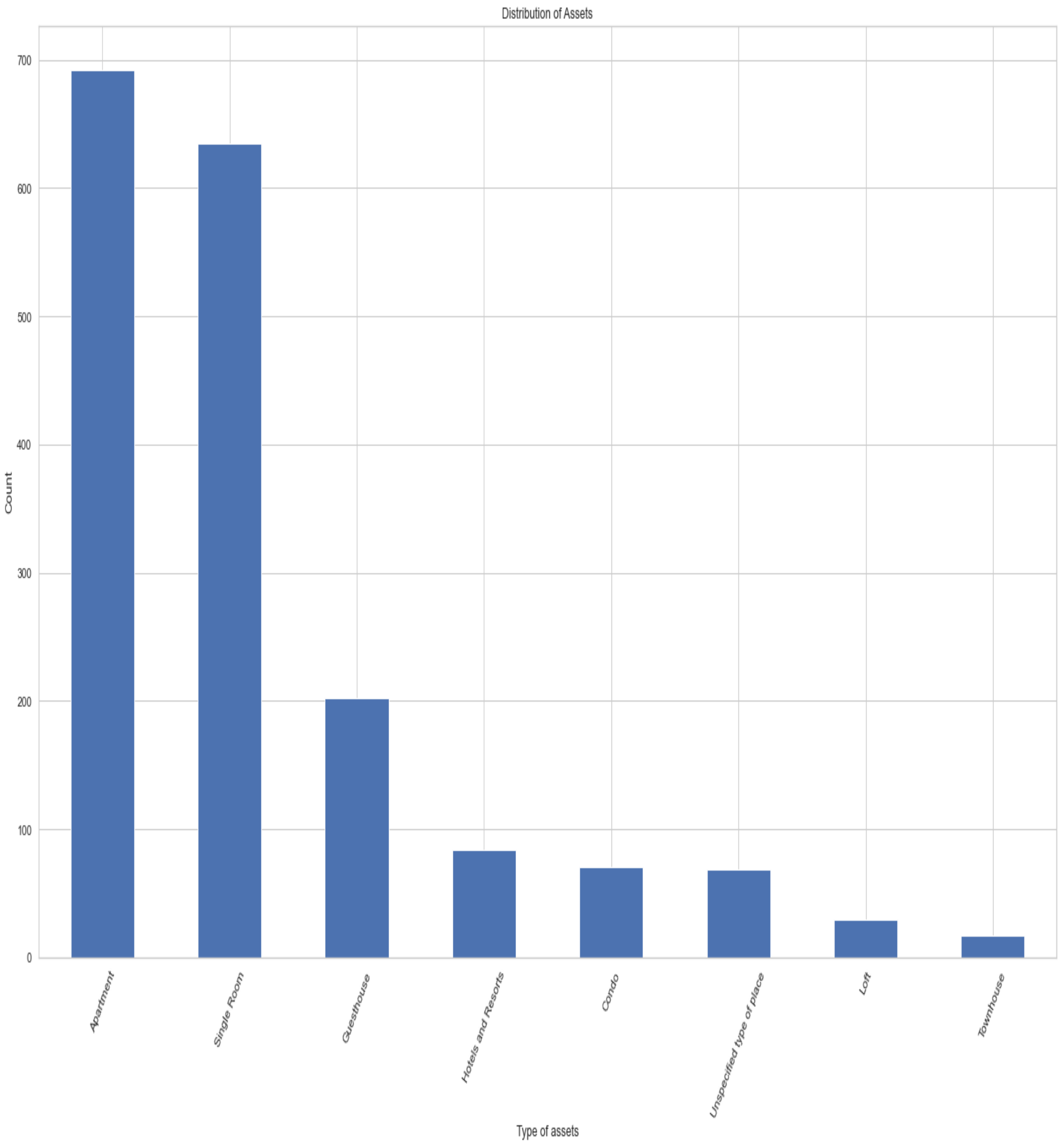
Filtering irrelevant features and records

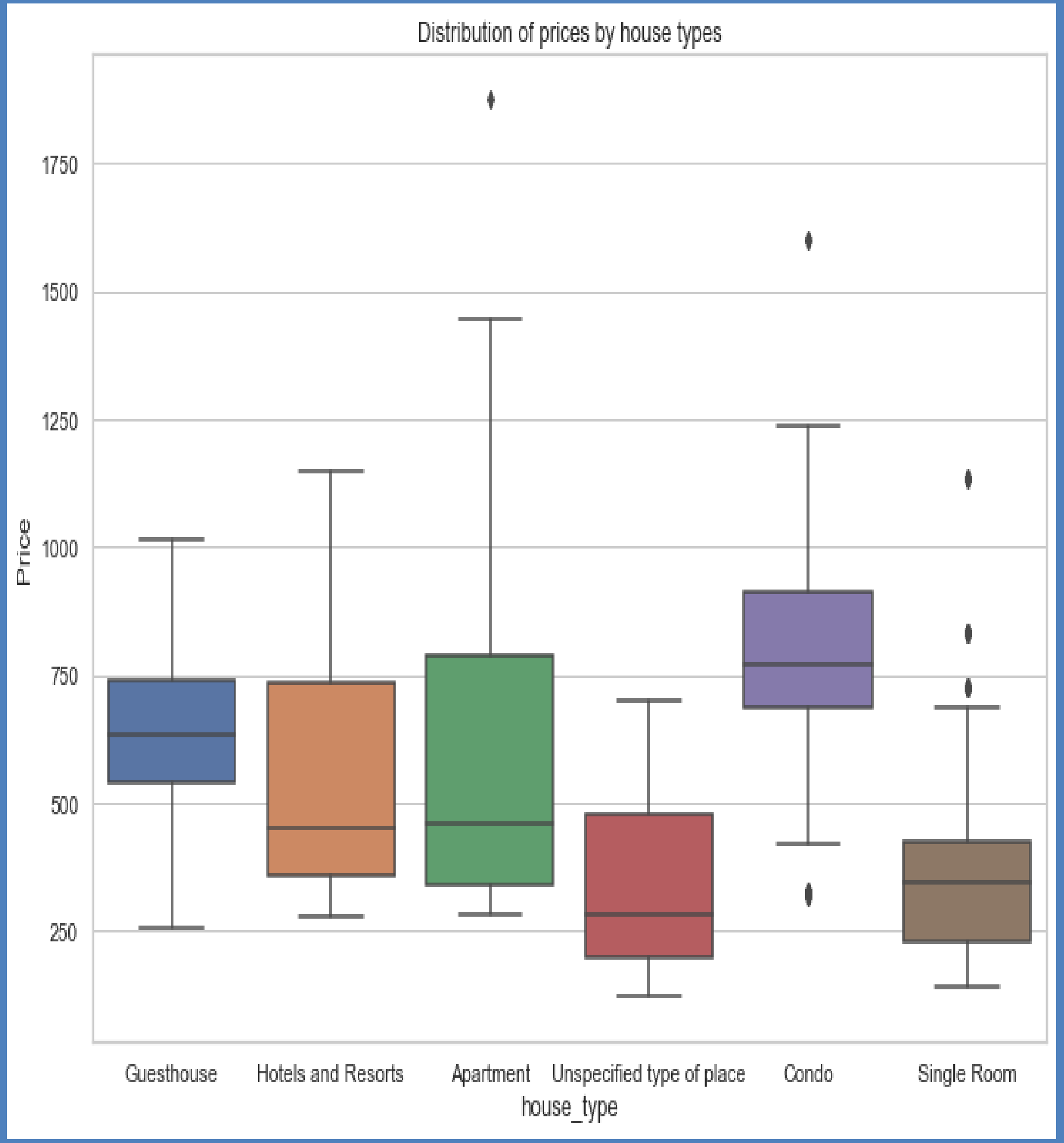
Handle missing values in our dataset

Stage 4: EDA & Visualization

We've created some charts and plot in order to understand the data better





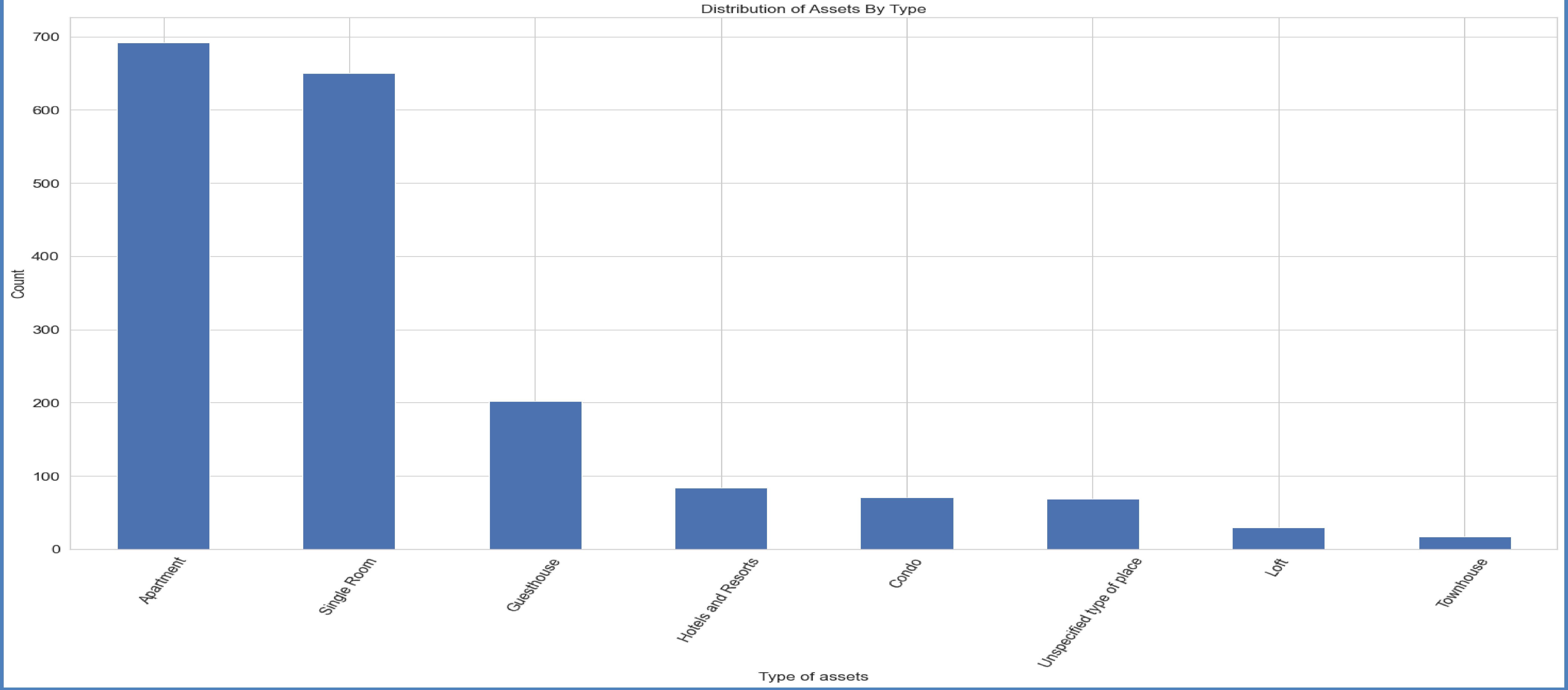


Heatmap

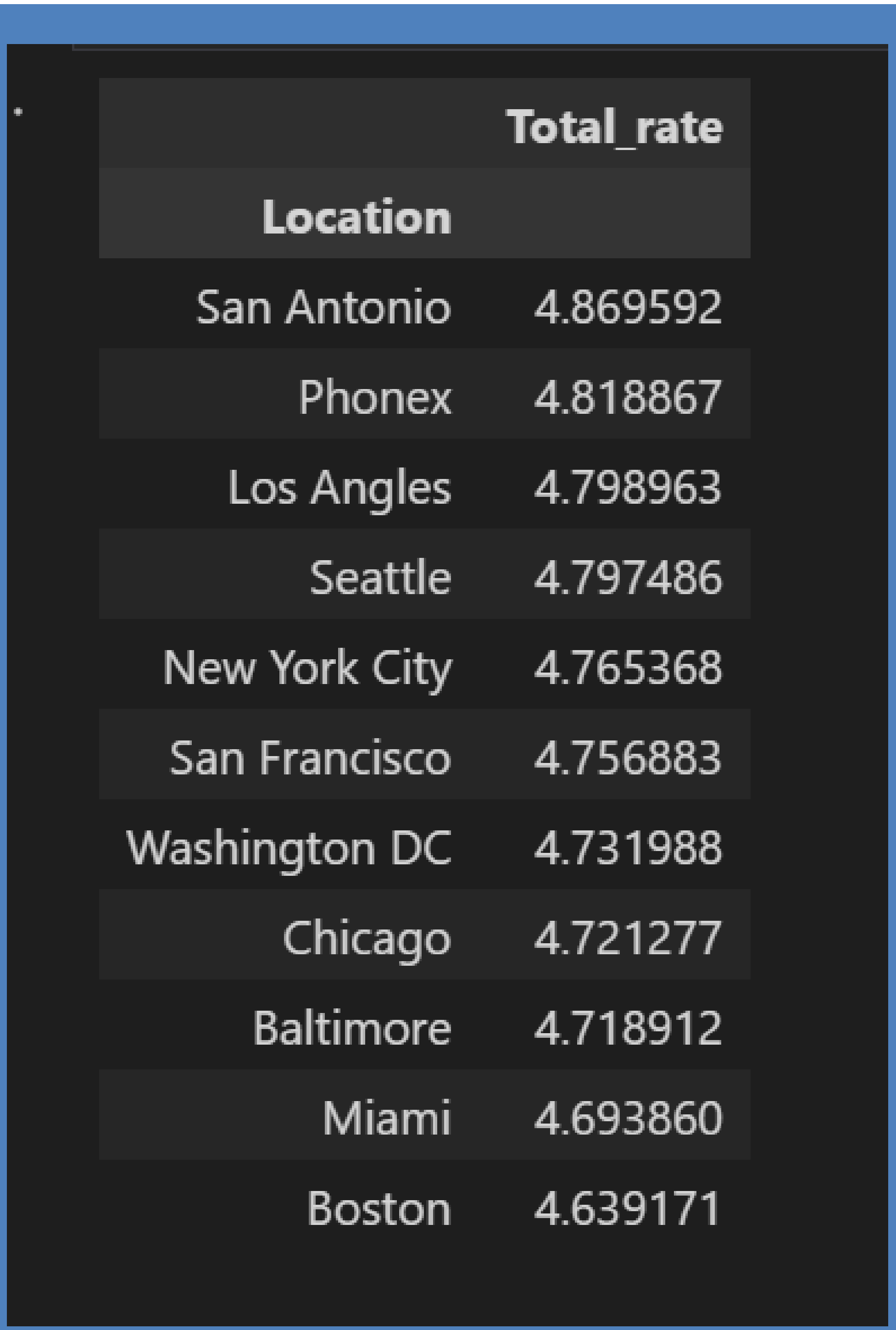
Barplot

Boxplot

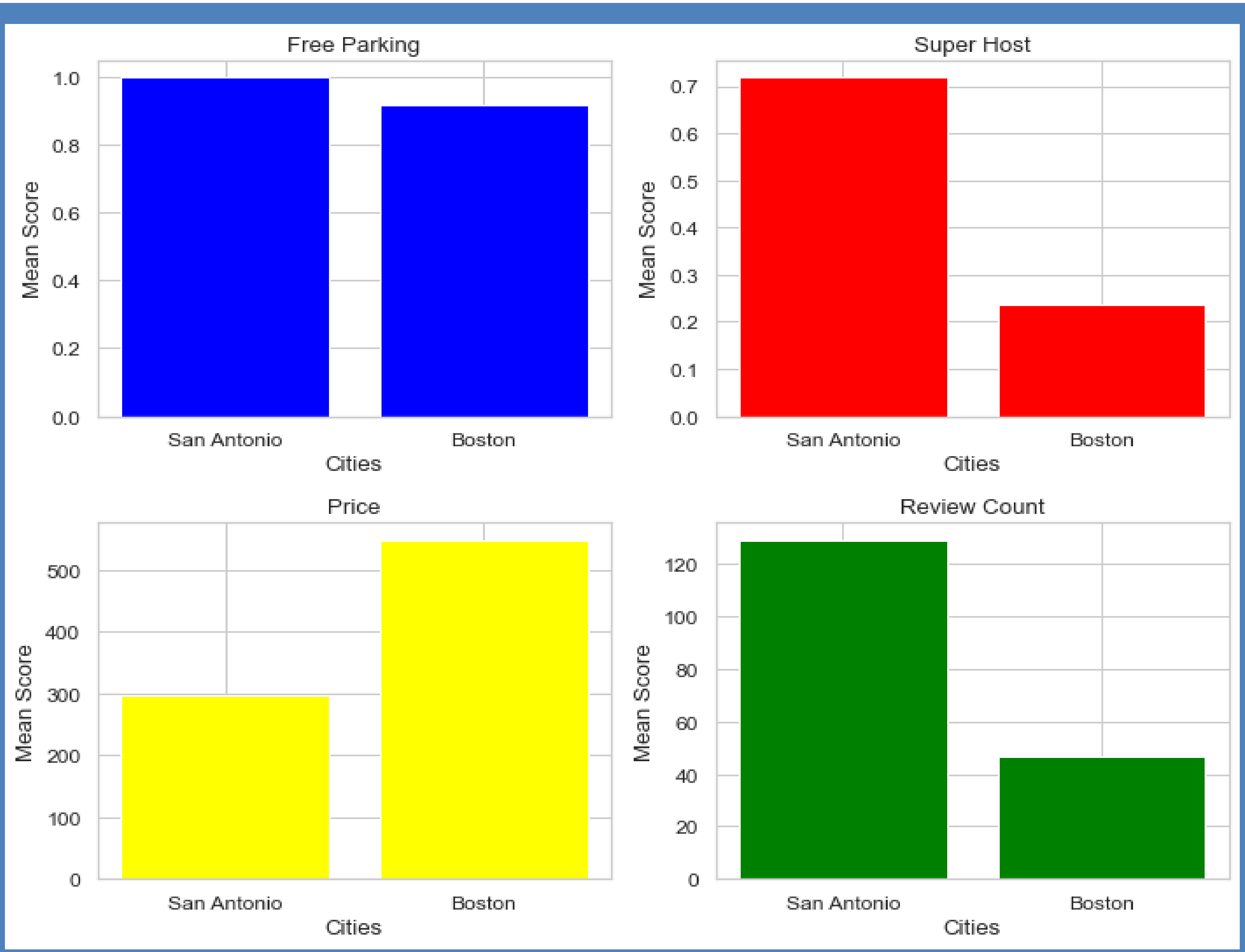
EDA & Visualization



EDA & Visualization



	Total_rate
Location	
San Antonio	4.869592
Phonex	4.818867
Los Angles	4.798963
Seattle	4.797486
New York City	4.765368
San Francisco	4.756883
Washington DC	4.731988
Chicago	4.721277
Baltimore	4.718912
Miami	4.693860
Boston	4.639171



Stage 5: Machine learning

Our Machine Learning question:

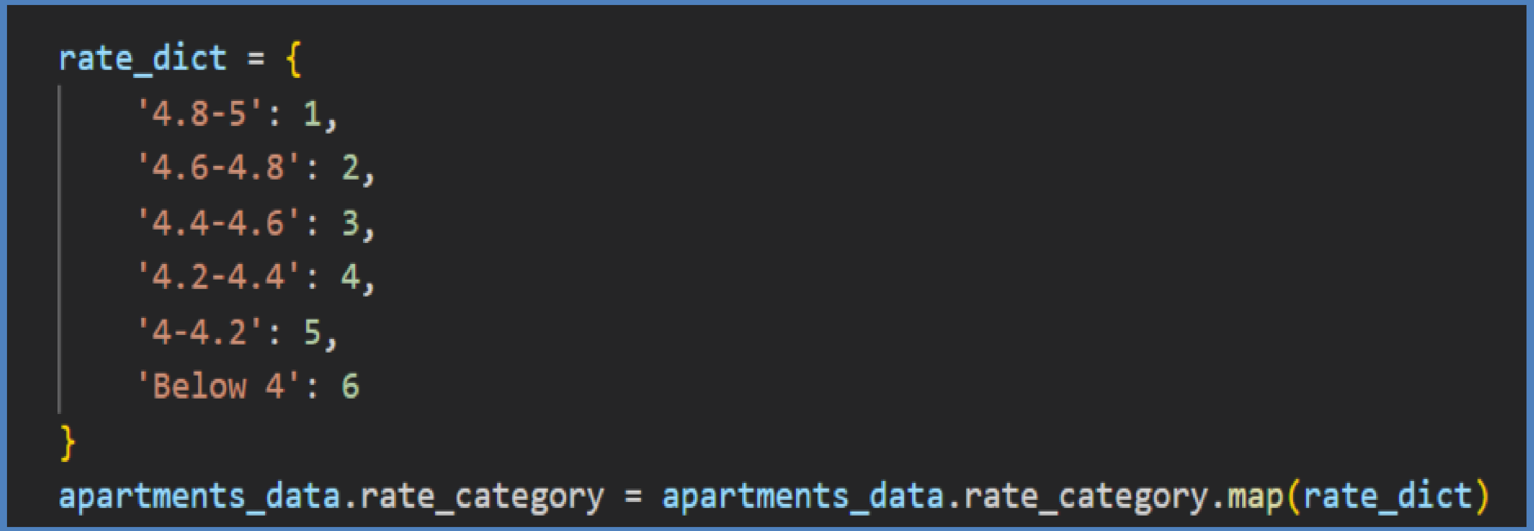
Can we predict Airbnb's apartment rate based on pre-defined attributes and make our vacation a successful experience?

In order to try and answer this question we have used 2 machine learning methods:

1. **Linear Regression**
2. **KNN**

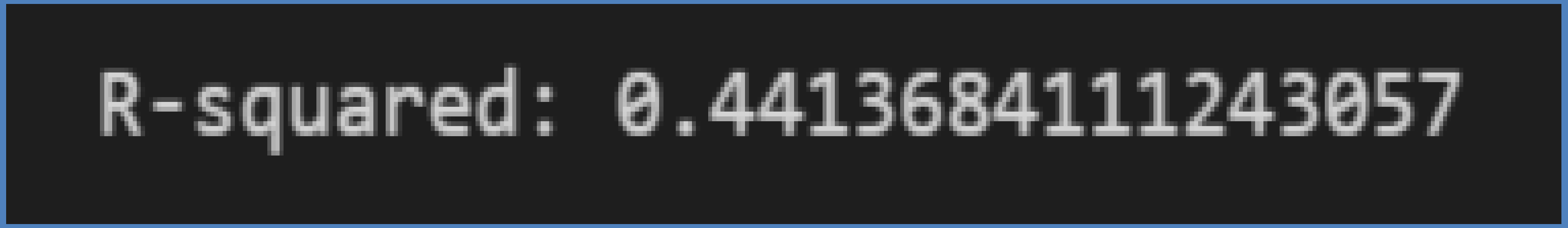
Linear Regression

Our first machine learning method was linear regression.
We used the ‘**rate_category**` column as our target column and change the different rate categories to scale from 1 to 6

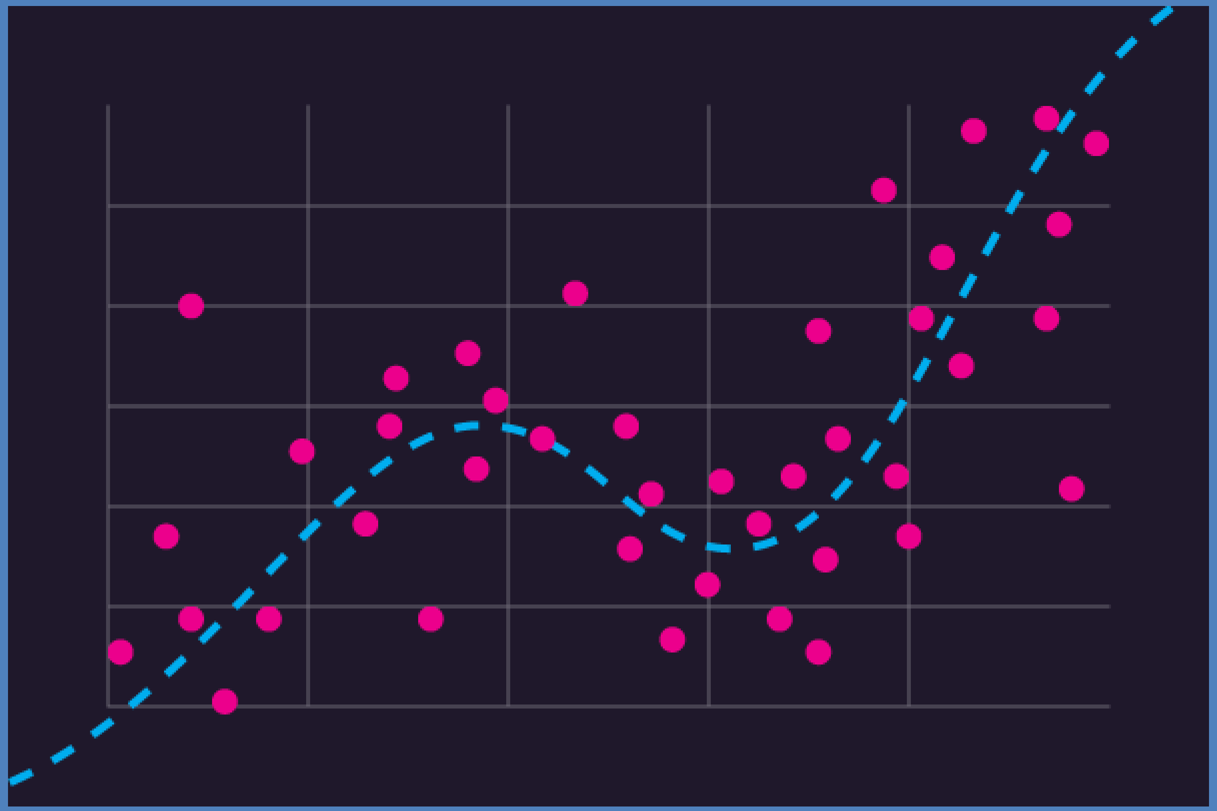


```
rate_dict = {
    '4.8-5': 1,
    '4.6-4.8': 2,
    '4.4-4.6': 3,
    '4.2-4.4': 4,
    '4-4.2': 5,
    'Below 4': 6
}
apartments_data.rate_category = apartments_data.rate_category.map(rate_dict)
```

Unfortunately, we got score of 0.44, which is not good enough for us.



```
R-squared: 0.4413684111243057
```

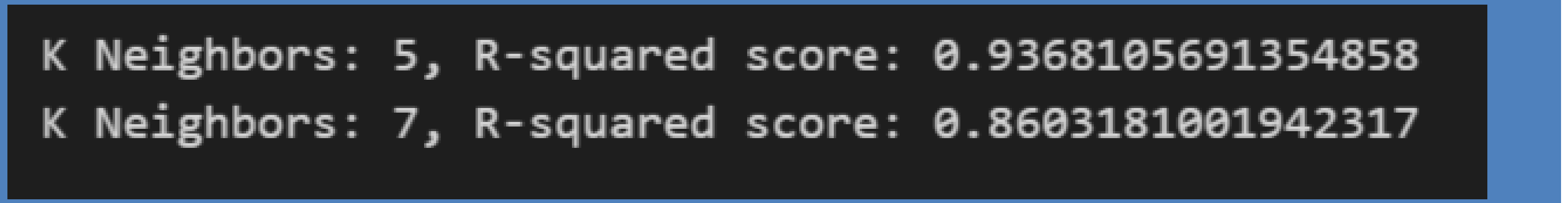


KNN

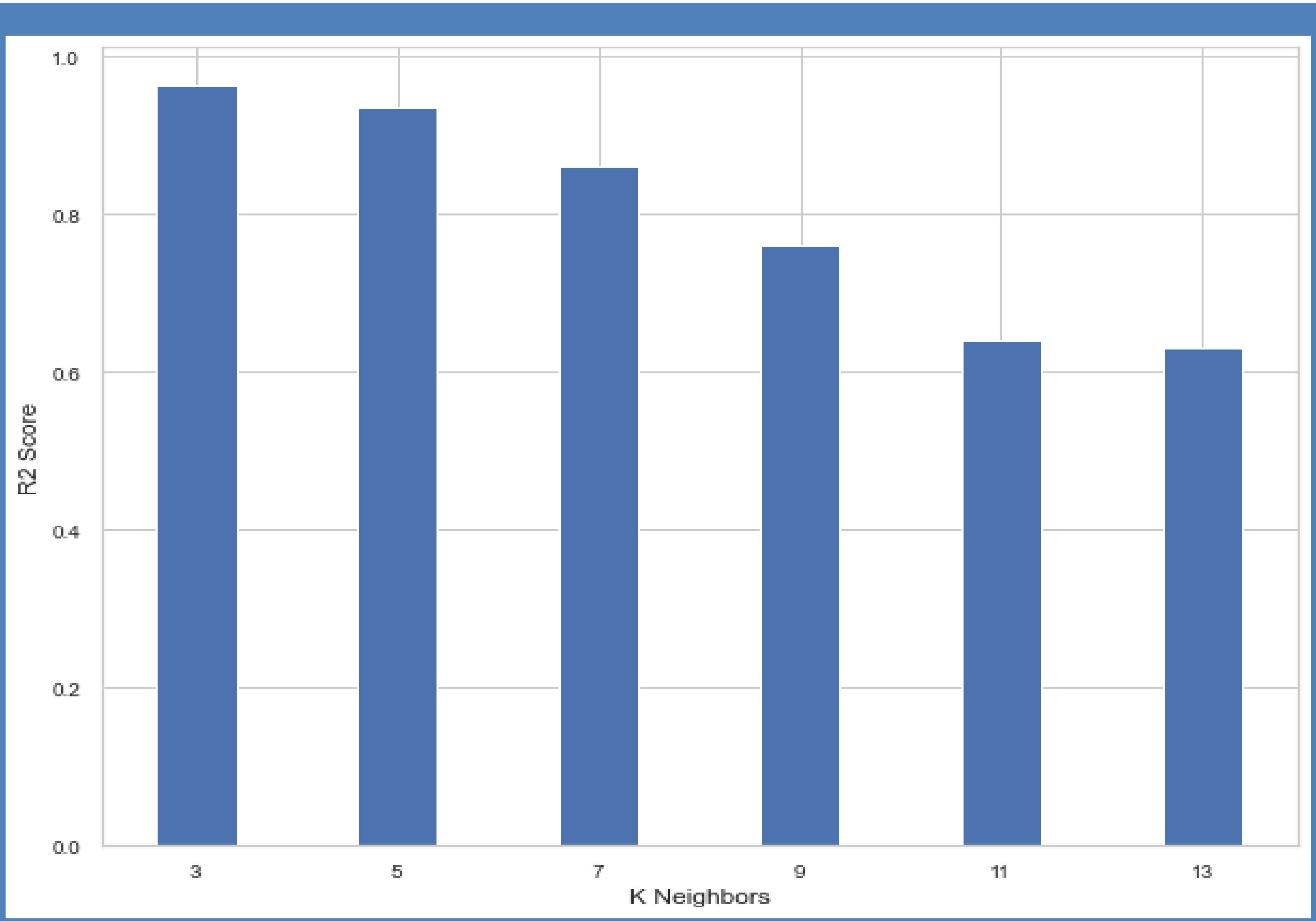
When we used KNN method, we noticed that our prediction results was much better.

We've used different K values (between 3 to 13), and notice that when K value increased the R-squared score decreased.

According to the results we got, the optimal K value for us was 5 or 7, which gave us success rate of more than 86%



```
K Neighbors: 5, R-squared score: 0.9368105691354858
K Neighbors: 7, R-squared score: 0.8603181001942317
```



Conclusion

- Our research question was trying to predict Airbnb's apartment rate based on pre-defined features

- We've tried to reach our goal by using different machine learning models and get the best prediction results

- We have reached out that linear regression method with our existing dataset and its features didn't gave us the result we want to achieve

- With KNN we got accuracy rate of more than 86% (depends on K value we chose) and we can determine the asset score based on our defined features

Thank you for listening!