Supervised Machine Learning: Regression

Course Project

Evgeny Zorin

26. September 2021

Airbnb NYC 2019 Price Prediction

Context

Since 2008, guests and hosts have used Airbnb to expand on traveling possibilities and present a more unique, personalized way of experiencing the world. Today, Airbnb became one of a kind service that is used and recognized by the whole world. Data analysis on millions of listings provided through Airbnb is a crucial factor for the company. These millions of listings generate a lot of data - data that can be analyzed and used for security, business decisions, understanding of customers' and providers' (hosts) behaviour and performance on the platform, guiding marketing initiatives, implementation of innovative additional services and much more.

Content

The dataset used in this project includes all the information needed to find out more about hosts, geographical availability, necessary metrics to make predictions and draw conclusions.

Main Objective

The main objective of the analysis is to build a predictive model based on linear regression that could predict the price of listings in New York City.

Data Summary

The dataset was taken from Kaggle and it contains information about 48895 Airbnb listings in New York City in 2019. Listing features include price, borough, neighbourhood, room types, location, host names, etc. There are 10052 null values in **last_review** and **reviews_per_month** columns, 16 null values in the **name** column and 21 null values in the **host_name** column. There are no duplicated values in the dataset. The features of the dataset and their description are shown in Table 1.

Table 1: Columns and Descriptions

Column	Description
id	Listing ID
name	Name of the listing
host_id	Host ID
host_name	Name of the host
neighbourhood_group	Location
neighbourhood	Area
latitude	Latitude coordinates
longitude	Longitude coordinates
room_type	Listing space type
price	Price in dollars
minimum_nights	Amount of nights minimum
number_of_reviews	Number of reviews
last_review	Latest review
reviews_per_month	Number of reviews per month
calculated_host_listings_count	Amount of listing per host
availability_365	Number of days when listing is available for booking

Data Cleaning, EDA and Feature Engineering

Data Cleaning

Columns such as **id, name, host_id, host_name** and **last_review** were dropped from the dataset, since they don't provide any useful information for the price prediction. It was also decided to drop the **latitude** and **longitude** columns because the dataset already has a **neighbourhood** column, which can more objectively explain the price of the listings.
Null values in column **reviews_per_month** were replaces with "0".

Summary Statistics of Numerical Features

The summary statistics of the numerical features is shown in Table 2.

Table 2: Summary Statistics of Numerical Features

	count	mean	std	min	25%	50%	75%	max
price	48895.0	152.720687	240.154170	0.0	69.00	106.00	175.00	10000.0
minimum_nights	48895.0	7.029962	20.510550	1.0	1.00	3.00	5.00	1250.0
number_of_reviews	48895.0	23.274466	44.550582	0.0	1.00	5.00	24.00	629.0
reviews_per_month	48895.0	1.090910	1.597283	0.0	0.04	0.37	1.58	58.5
calculated_host_listings_count	48895.0	7.143982	32.952519	1.0	1.00	1.00	2.00	327.0
availability_365	48895.0	112.781327	131.622289	0.0	0.00	45.00	227.00	365.0

The summary statistics indicates that most of the numerical features have a right skewed distribution and probably contain outliers. For example, 75% of the prices are lower than $175, but the maximum price is $10000. This can be seen more clearly with the histograms in Figure 1.

Distribution of Numerical Features

Figure 1 confirms the assumption that all of the numerical features are right skewed. Figure 2 also confirms the existence of outliers.

Figure 1: Distribution of Numerical Features

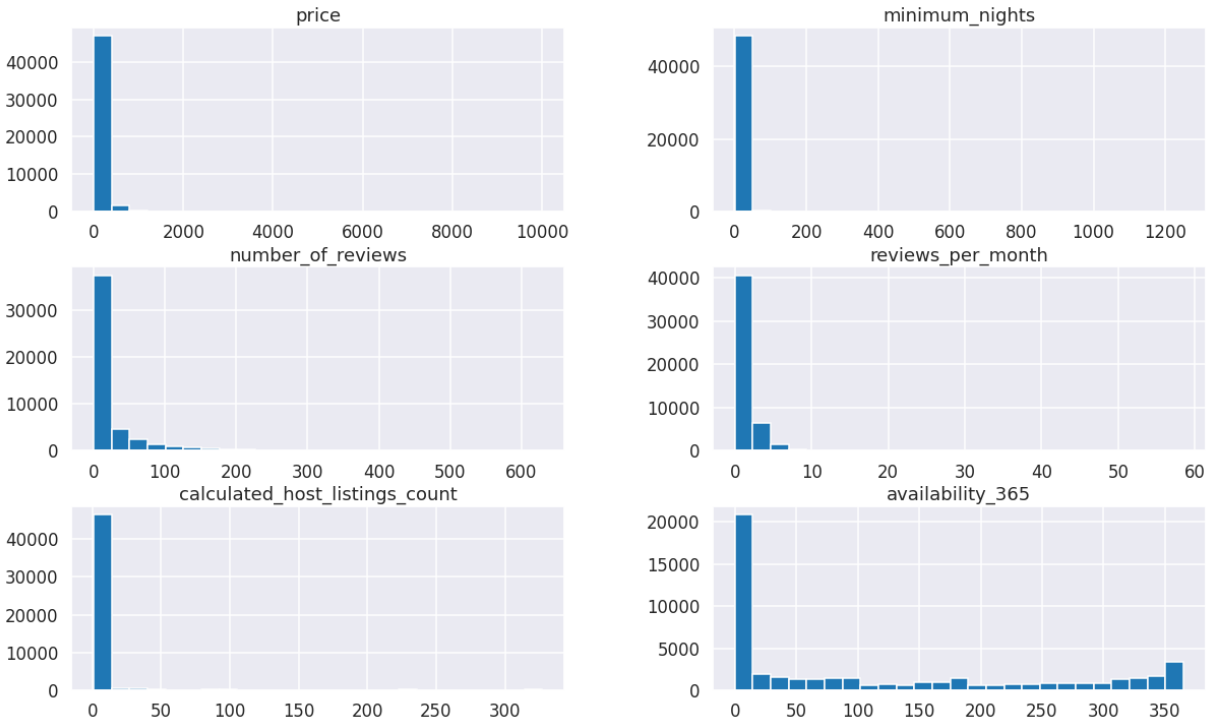
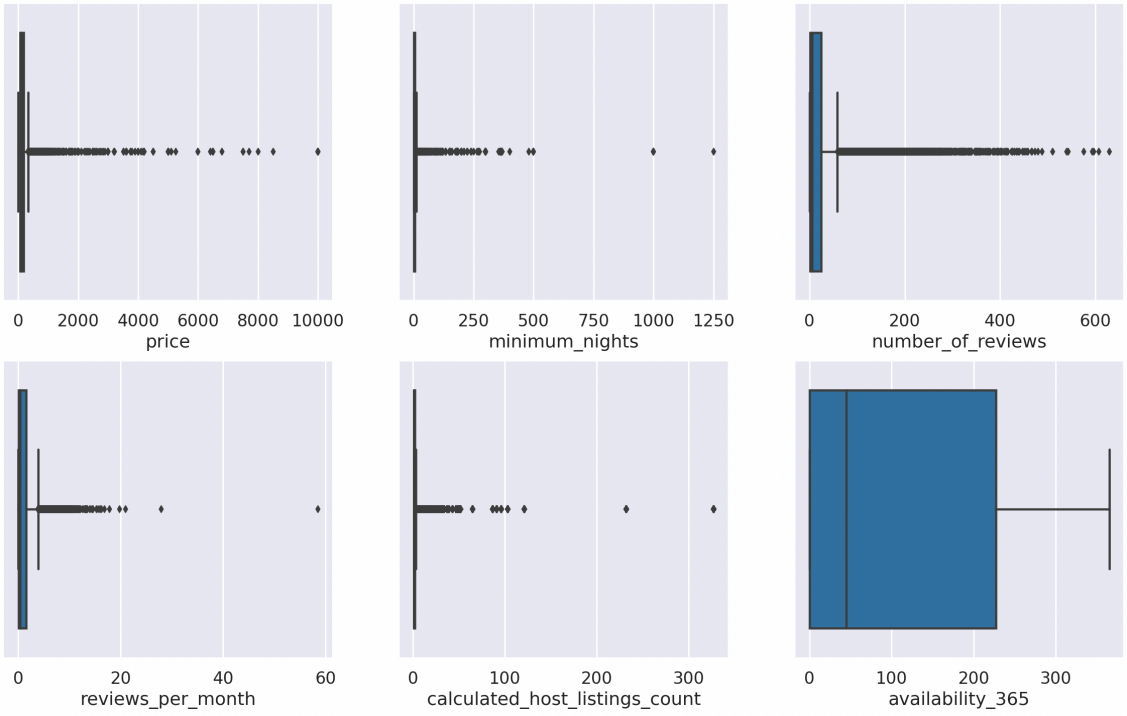


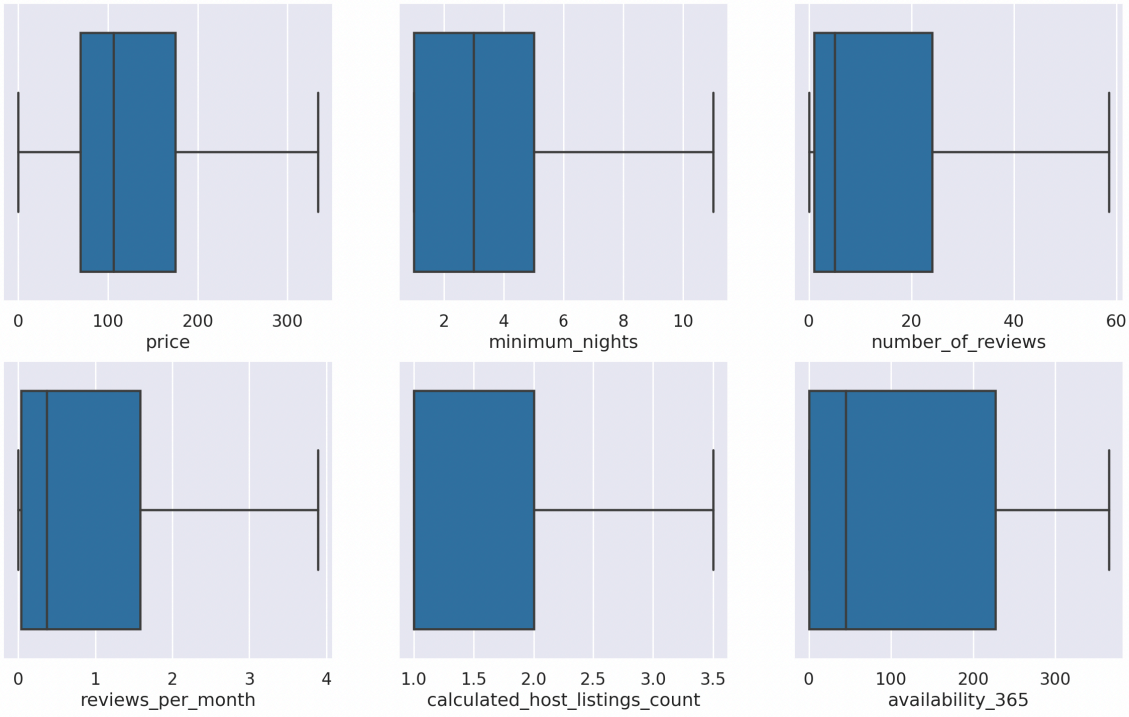
Figure 2: Distribution of Outliers in Numerical Features



Outliers Handling using 1.5IQR rule

The 1.5IQR rule will be used to solve the problem of outliers in the data. This rule finds the lower and upper bounds for outliers in an array. The distribution of the numerical features after the removal of outliers is shown in Figure 3.

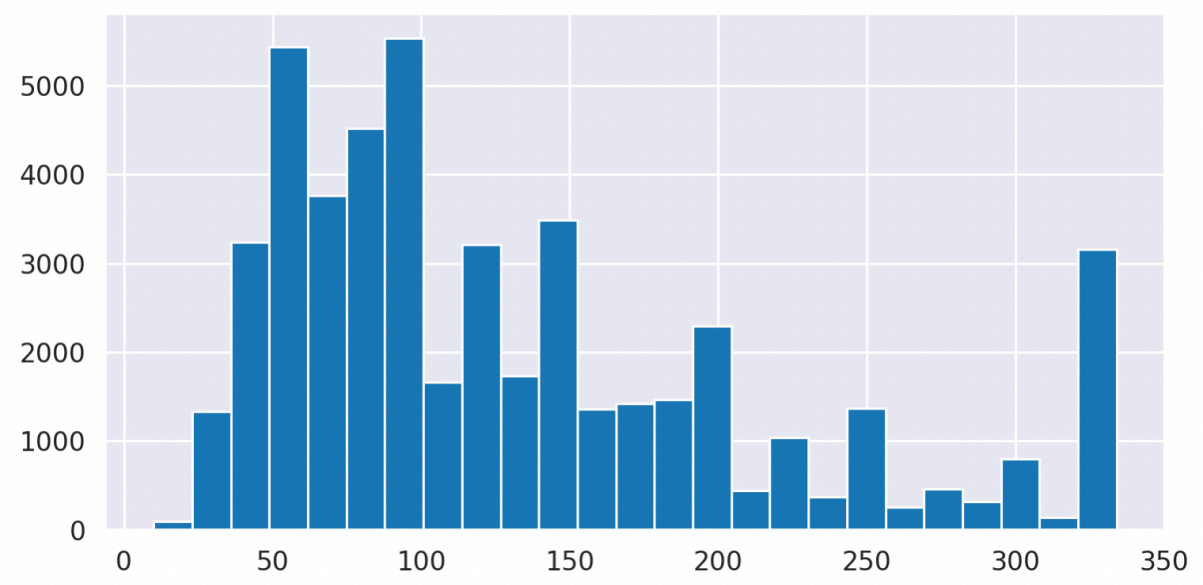
Figure 3: Distribution of Numerical Features without Outliers



Target Feature Handling

There are 11 listings in the dataset with a price of $0. These listings were interpreted as promotions or bad data and were dropped from the dataset. The distribution of the **price** after outliers and target feature handling is shown in Figure 4.

Figure 4: Distribution of the Price



Summary Statistics of Categorical Features

The summary statistics of the categorical features is shown in Table 3.

Table 3: Summary Statistics of Categorical Features

	count	unique	top	freq
neighbourhood_group	48895	5	Manhattan	21661
neighbourhood	48895	221	Williamsburg	3920
room_type	48895	3	Entire home/apt	25409

There are 5 unique boroughs and 221 unique neighbourhoods in New York City. The most represented boroughs are Manhattan with 21661 listings and Brooklyn with 20095 listings, due to the fact that almost all of the attractions of New York City are in these two boroughs. There are also 3 unique types of the offered accommodations such as entire home/apartment, private room and shared

room, most of which represented by category entire home/apartment with 25409 listings. Private rooms and entire homes/apartments cover almost 97% of all the listings. This can be seen more clearly with the distributions in Figure 5 and Figure 6.

Since these two boroughs are the most popular, it can be assumed that the price of listings in them will be much higher than in the other boroughs. The same can be said about the types of rooms. The distribution of the average prices in Figure 7 confirms this.

The most represented neighbourhood is Williamsburg with 3920 listings. As expected, the top 10 neighbourhoods according to the listings are in Manhattan and Brooklyn. The less represented neighbourhoods (less than 500 listings) were combined into one category **others**.

Figure 5: Distributions of Neighbourhood Groups and Room Types

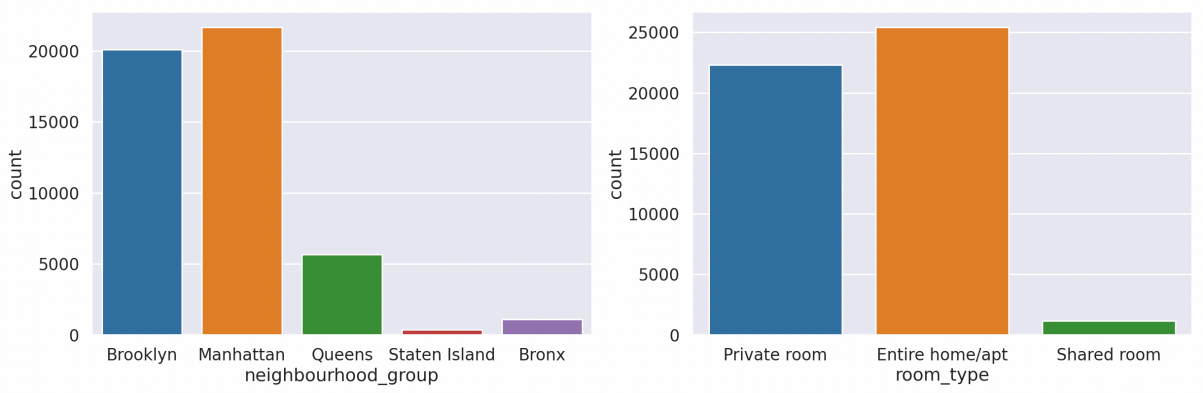


Figure 6: Distribution of Neighbourhoods

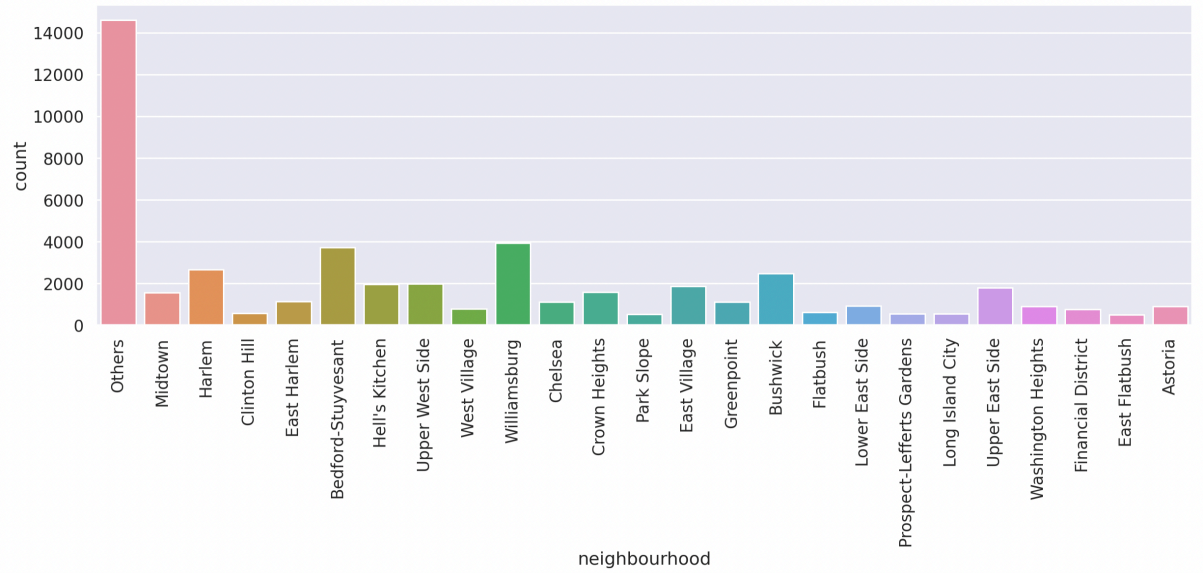
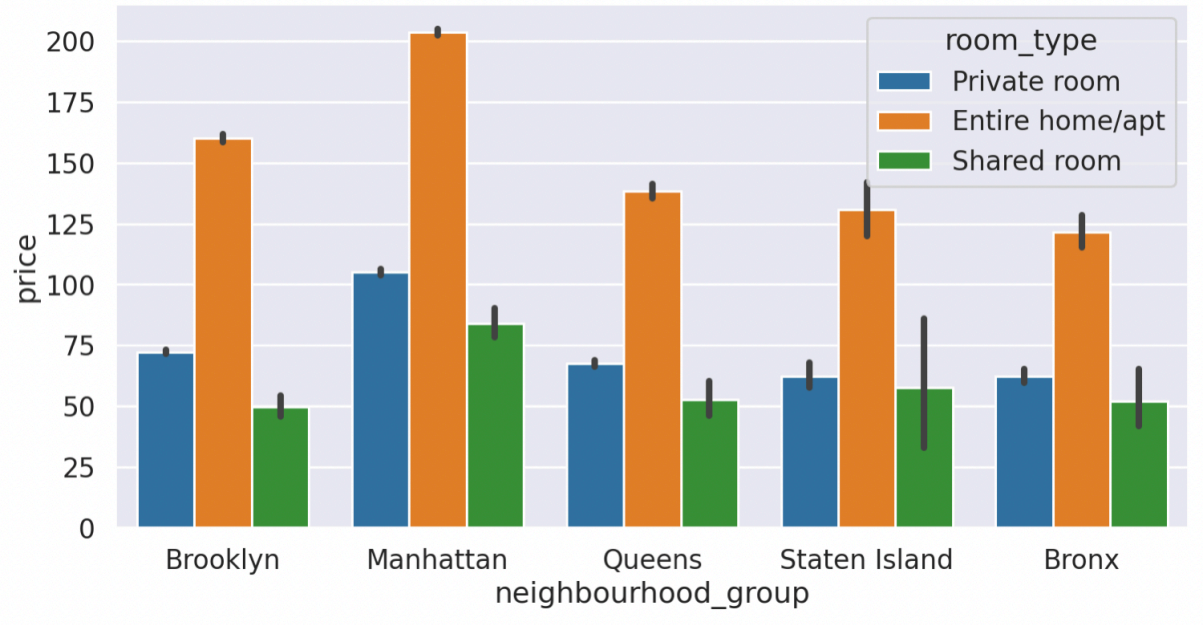


Figure 7: Distribution of the Price by Neighbourhood Groups and Room Types



Final Data Cleaning

After exploratory analysis of the numerical and categorical features it was decided to drop some features such as neighbourhood, reviews_per_month and availability_365 from the dataset. This decision was based on the assumption that the neighbourhood groups can predict the price of the listings as good as the neighbourhoods separately. Also the number_of_reviews has the greater impact on the price prediction than the reviews_per_month. And the availability_365 has not impact on the price prediction at all.
Column calculated_host_listings_count was renamed into host_listings_count.

Ordinal Encoding

The ordinal encoding was performed on the features neighbourhood_group and room_type as follows:

neighbourhood_group: Bronx = 0, Brooklyn = 1, Manhattan = 2, Queens = 3, Staten Island = 4

room_type: Entire home/apt = 0, Private room = 1, Shared room = 2

The dataset after the encoding of the categorical features is shown in Table 4.

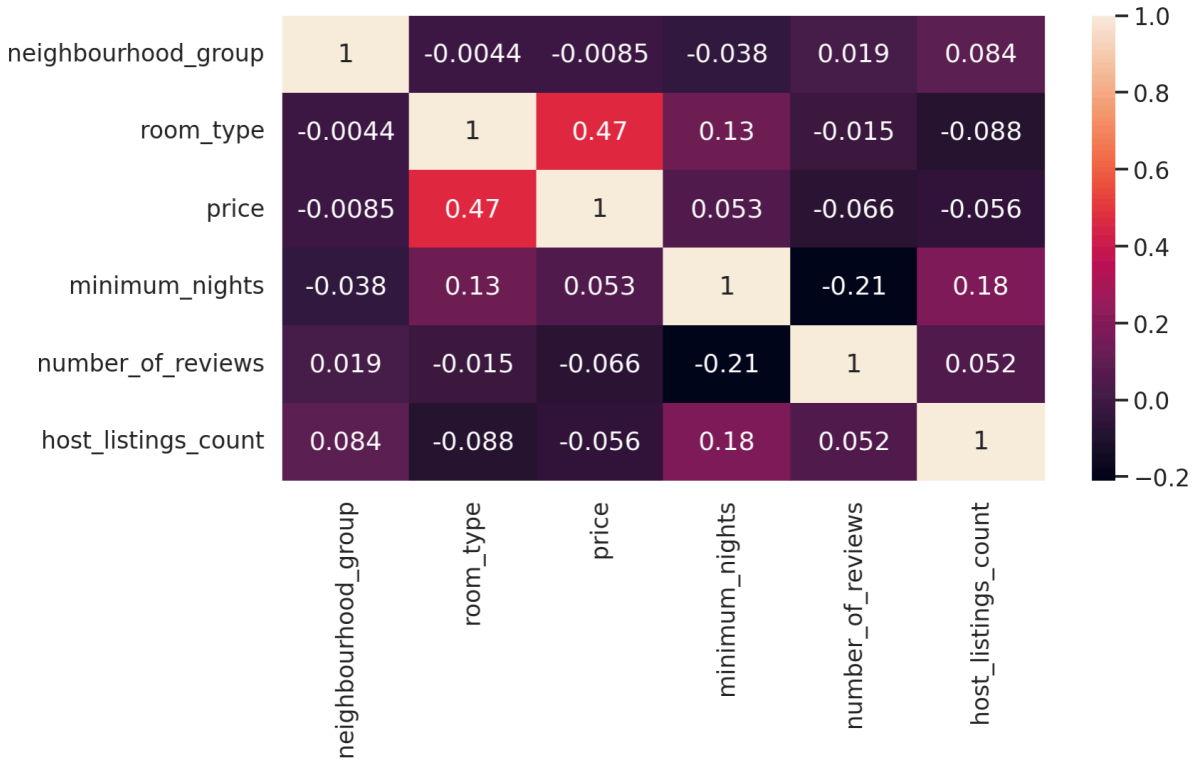
Table 4: Final Dataset Sample

	neighbourhood_group	room_type	price	minimum_nights	number_of_reviews	host_listings_count
0	0	0	149.0	1.0	9.0	3.5
1	1	1	225.0	1.0	45.0	2.0
2	1	0	150.0	3.0	0.0	1.0
3	0	1	89.0	1.0	58.5	1.0
4	1	1	80.0	10.0	9.0	1.0

Correlation Map of the Features

There is no correlation between features on the heat map, other than a weak correlation between **price** and **room_type** features.
The correlation map of the features is shown in Figure 8.

Figure 8: Correlation Map of the Features



Modeling and Prediction

To build a predictive model, it was decided to train the dataset on the Baseline Linear Regression, Lasso Regression and Ridge Regression, and based on the largest R-squared score and the smallest Root Mean Squared Error, choose the best model for the test.

Train Test Split and Modeling Preparation

The dataset was splitted into a training set and a testing set in the ratio of 70% of the training set and 30% of the testing set. StandardScaler was used to train the dataset to get all the data at the same scale. To obtain the most accurate score, a cross validation method was applied. It splits the training set into 5 folds and produces an average score of those folds. Lasso and Ridge Regressions also used polynomial features with degree=3 to add degree and some interaction terms. The Pipeline model was used to combine all training steps together.

Baseline Linear Regression

Training the data with Baseline Linear Regression shows that only 22.2% of dependent feature which is price can be explained by independent features. Root Mean Squared Error is also high and equals to 73.76 (see Table 5).

Table 5: Linear Regression Results

	R2 Score	RMSE
0	0.222251	73.767782

Lasso Regression

Training the data with Lasso Regression, it was decided to run the model with a bunch of alphas (10 alphas between 0.02 and 0.2) and choose the one that best tunes the model. The best result was reached with alpha 0.043. The Lasso Regression result are much better and show that now 43.8% of dependent feature can be explained by independent features. Root Mean Squared Error is still high and equals to 62.7. Distribution of Lasso Regression results with different alphas is shown in Table 6.

Table 6: Lasso Regression Results

	Alpha	R2 Score	RMSE
0	0.020000	0.438037	62.704808
1	0.025831	0.438047	62.704271
2	0.033362	0.438055	62.703833
3	0.043089	0.438058	62.703682
4	0.055651	0.438051	62.704068
5	0.071876	0.438012	62.706246
6	0.092832	0.437945	62.709936
7	0.119897	0.437839	62.715899
8	0.154853	0.437631	62.727492
9	0.200000	0.437250	62.748721

Ridge Regression

Training the data with Ridge Regression, it was decided to run the model with a bunch of alphas (10 alphas between 5 and 50) and choose the one that best tunes the model. The best result was reached with alpha 38.7. The Ridge Regression results are almost the same as the Lasso Regression results and show also that 43.8% of dependent feature can be explained by independent features. Root Mean Squared Error is still high and equals to 62.7. Distribution of Ridge Regression results with different alphas is shown in Table 7.

Table 7: Ridge Regression Results

	Alpha	R2 Score	RMSE
0	5.000000	0.437992	62.707329
1	6.457748	0.437993	62.707286
2	8.340503	0.437994	62.707233
3	10.772173	0.437995	62.707169
4	13.912797	0.437996	62.707096
5	17.969068	0.437998	62.707015
6	23.207944	0.437999	62.706933
7	29.974213	0.438000	62.706864
8	38.713184	0.438001	62.706837
9	50.000000	0.438000	62.706898

Comparing the Training Results

Comparing the results of all three models, it can be seen that the results of the Lasso regression are slightly better than those of the ridge regression. Thus, Lasso regression with alpha 0.043 was chosen as the best model. Comparison table of all the models is shown in Table 8.

Table 8: Comparison Table

	Model	R2 Score	RMSE
0	Linear	0.222251	73.767782
1	Lasso	0.438058	62.703682
2	Ridge	0.438001	62.706837

Prediction

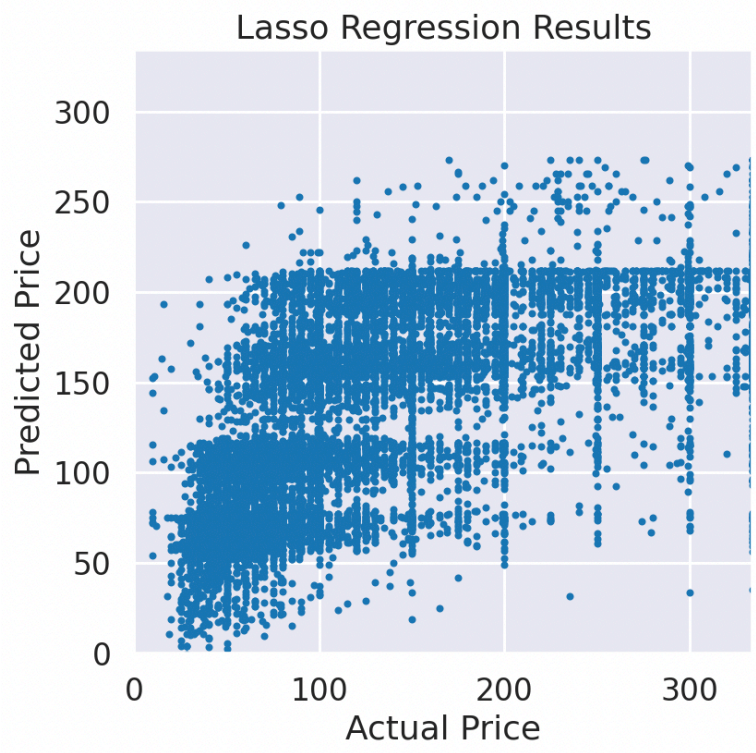
To test the model, Lasso regression with an alpha of 0.043 was trained on the whole training set without using the cross validation method. The model was further tested on the testing set, and the prediction score was 43.5%. The prediction score is slightly lower than the training score of 43.8%, but the Root Mean Squared Error in this case is also slightly lower (62.5 vs 62.7). The prediction results are shown in Table 9.

Table 9: Prediction Results

	R2 Score	RMSE
0	0.435035	62.542625

To examine how well the prediction model performs, a scatter plot was plotted to visualize the predicted price compared to the actual price (see Figure 9). The plot shows that the prediction model is not accurate, and the independent features that were chosen to build this model are not able to ensure correct price prediction.

Figure 9: Predicted Price vs Actual Price



Key Findings and Insights

As assumed at the beginning of the project, the features for building a prediction model should have been the following five key features: **neighbourhood_group**, **room_type**, **number_of_reviews**, **host_listings_count** and **minimum_nights**. Among them, the following two features have been defined as the main with the highest impact on the listing price: **neighbourhood_group** and **room_type**.

By analyzing these two features, it became clear that the average price of accommodations in Manhattan was 2 times higher and in Brooklyn 1.5 times higher than in other boroughs. While the average price in other boroughs is not significantly varied. Also the average price for an entire home/apartment is more than 2 times higher than the other room types.

Both Lasso and Ridge Regressions with proper hyper-parameter tuning perform better training results than the Baseline Linear Regression, due to polynomial features, which help better understanding the nonlinear relationship between the independent features. Lasso has the smallest Root Mean Square Error,

however the difference in scores and errors are insignificant and almost identical. Comparing these three regressions, the best candidate to build a prediction model, based on Root Mean Square Error and R2 Score results is Lasso Regression.
Prediction results show that the five key features mentioned above can predict the price of listings with an accuracy of only 43.5% and a fairly large error of 62.5.

Next Steps

The next steps to achieve the better model prediction would be to return the neighbourhood and perhaps the latitude/longitude features to the dataset, as well as to perform one-hot encoding instead of ordinal encoding for all features. Using GridSearchCV to find the optimal optimization parameters would help to achieve better results either.

Appendix

GitHub URL:
https://github.com/evgenyzorin/IBM-Machine-Learning/tree/main/Regression