3 Regression model

After conducting the exploratory analysis it is critical to determine what we want to achieve in our regression analysis and prepare a solid ground for finding the cause-and-effect relationship between variables. Choosing *"price"* as a predicted variable allows us to extract the valuable relationship between Airbnb rates and apartment characteristics.

Once we have established our predicted variable and goal, the next step is to gather and clean our data. This includes identifying relevant independent variables, such as location, room type, number of reviews, availability and etc., and making sure there are no missing or inconsistent data points. Next, we can perform data analysis to gain insights into the relationships between our variables and identify any outliers or patterns. Finally, we can use statistical methods, such as linear regression, to build a model that can accurately predict the *"price"* based on the characteristics of the apartment. The results of this analysis can then be used to make data-driven decisions and understand the impact of different factors on Airbnb rates.

3.1 Data preparation

Starting with deleting zero *"price"* data values in a dataset since it doesn't bring any information and is not economically sustainable - we continue with converting the *"price"* values into natural logarithm. In short, transforming the price data by taking the natural logarithm can improve the performance and reliability of the regression model, because the output of linear regression does not respect the bounds of zero. Additionally, using the logarithm of price can also help account for proportional changes in the dependent variable rather than absolute changes.

Before using regression analysis, it is important to understand that not all independent variables (also known as "regressors") are numerical. For example, a regressor may be a categorical variable such as gender, type of product, or location. To make use of categorical variables in regression analysis, they must first be transformed or encoded into numerical variables. That is exactly what we did with our regressors *"neighbourhood_group"* and *"room_type"*, simply converting them into dummy variables.

We performed some noteworthy transformations on our data, specifically calculating the length of the property title. Our reasoning behind this was that a longer property title would provide potential renters with more information about the property and its features,

potentially making it more appealing to them. This, in turn, could result in the owner being able to command a higher rental fee.

Furthermore, we enhanced the *"host_name"* information by transforming it from a string of characters to male and female variables using Natural Language Toolkit (NLTK). To accomplish this, we created and trained a classifier on a sample of over 12,000 names. This process involved using machine learning algorithms and NLTK's libraries to analyze the names and categorize them as male or female. The end result was a more organized and structured representation of the host name information, allowing us to better understand the gender distribution of hosts and its potential impact on other variables in our data.

The table below presents a summary of the overall transformations performed on the variables.

Variable	Before	After
price	0	- delete
		- convert to natural log
reviews_per_month	Nan	0
last_review	0	- change to datetime type
		- calculate days from the last review
name_length	str()	calculate the length of the title
host_name	str()	- NLTK
		- train a Naive Bayes classifier
		- conversion to dummy variables
neighbourhood_group,	str()	convert categorical variables into dummies
room_type		

Table 3: Variable transformation

3.2 Auto outlier detection methods

As previously stated, outliers are data points that deviate significantly from other observations in the dataset. It is important to detect and eliminate outliers when training machine learning models, particularly when linear regression is involved. This helps to ensure that the model is trained on accurate and reliable data, leading to more accurate predictions.

To begin with, we evaluated the performance of our baseline linear regression model, which was trained on 67% of the randomly selected data set without the removal of any outliers. The Mean Absolute Error (MAE) was utilized as the evaluation metric, and the model achieved a score of 62.05. This result served as a reference point for us to compare the performance of our future models. It provided us with insight into the baseline accuracy of our linear regression model and helped us determine if any improvements could be made by modifying the data set by deleting outliers with the specific method.

We progressed by applying various automatic outlier detection techniques and comparing their results. Each method was defined and applied to the training set, and the resulting model was used to predict which examples in the set were outliers and which were not. Once the outliers were identified and removed from the training data set, the model was refitted to the remaining examples. It's crucial to note that fitting the outlier detection method to the entire dataset would result in data leakage and a falsely optimistic assessment of model performance, and therefore must be avoided.

The following methods were used:

Isolation Forest is a tree-based anomaly detection algorithm, that is based on modeling the normal data in such a way as to isolate anomalies that are both few in number and different in the feature space. MAE: 61.788.

Minimum Covariance Determinant. If the input variables are assumed to have Gaussian distribution this simple statistical method can be used to detect outliers. This approach can be generalized by defining a hypersphere (ellipsoid) that covers the normal data, and data that falls outside this shape is considered an outlier. MAE: 62.192.

Local Outlier Factor, LOF for short, is a technique that attempts to harness the idea of nearest neighbors for outlier detection. This approach is straightforward - locate those examples that are far from the other examples in the feature space. MAE: 61.540.

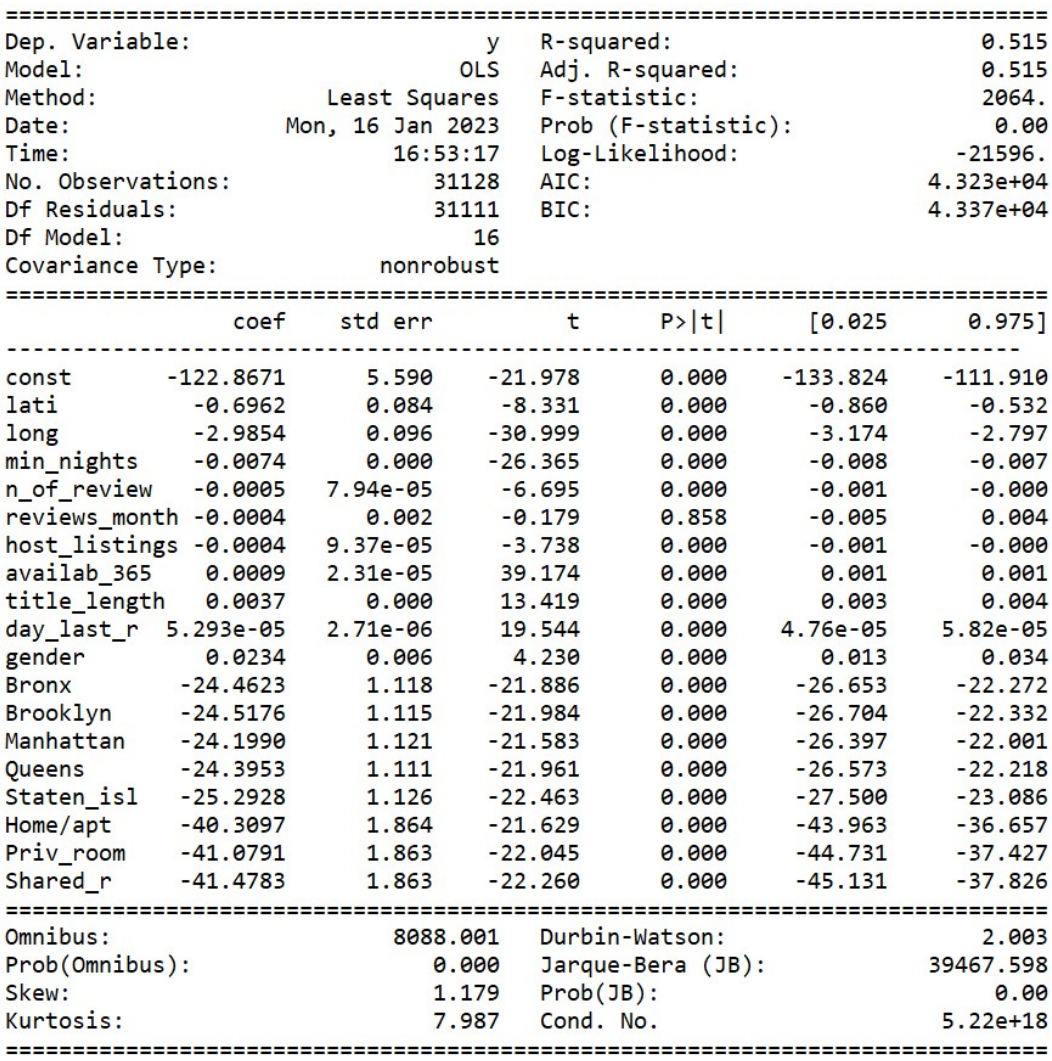
One-Class Support vector machine. When modeling one class, the algorithm captures the density of the majority class and classifies examples on the extremes of the density function as outliers. MAE: 61.764.

In conclusion, the LOF outlier detection method produced the best outcome with the lowest MAE (61.540).

3.3 OLS model summary

Based on the fact that the Local Outlier Factor model produced the smallest mean absolute error we use this method to delete outliers from the training data set and then build an ordinary least squares (OLS) model.

The predicted variable *price* was transformed using the natural logarithm, resulting in a log-level regression. This transformation allowed us to analyze the relationship between the input variables and the target variable in terms of percent changes instead of absolute changes. In other words, a unit change in the regressors corresponds to a percent change in the predicted "price". This type of regression can provide insights into the underlying relationship between the variables, making it easier to interpret the results and draw meaningful conclusions from the model.



Dep. Variable:	y	R-squared:	0.515
Model:	OLS	Adj. R-squared:	0.515
Method:	Least Squares	F-statistic:	2064.
Date:	Mon, 16 Jan 2023	Prob (F-statistic):	0.00
Time:	16:53:17	Log-Likelihood:	-21596.
No. Observations:	31128	AIC:	4.323e+04
Df Residuals:	31111	BIC:	4.337e+04
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-122.8671	5.590	-21.978	0.000	-133.824	-111.910
lati	-0.6962	0.084	-8.331	0.000	-0.860	-0.532
long	-2.9854	0.096	-30.999	0.000	-3.174	-2.797
min_nights	-0.0074	0.000	-26.365	0.000	-0.008	-0.007
n_of_review	-0.0005	7.94e-05	-6.695	0.000	-0.001	-0.000
reviews_month	-0.0004	0.002	-0.179	0.858	-0.005	0.004
host_listings	-0.0004	9.37e-05	-3.738	0.000	-0.001	-0.000
availab_365	0.0009	2.31e-05	39.174	0.000	0.001	0.001
title_length	0.0037	0.000	13.419	0.000	0.003	0.004
day_last_r	5.293e-05	2.71e-06	19.544	0.000	4.76e-05	5.82e-05
gender	0.0234	0.006	4.230	0.000	0.013	0.034
Bronx	-24.4623	1.118	-21.886	0.000	-26.653	-22.272
Brooklyn	-24.5176	1.115	-21.984	0.000	-26.704	-22.332
Manhattan	-24.1990	1.121	-21.583	0.000	-26.397	-22.001
Queens	-24.3953	1.111	-21.961	0.000	-26.573	-22.218
Staten_isl	-25.2928	1.126	-22.463	0.000	-27.500	-23.086
Home/apt	-40.3097	1.864	-21.629	0.000	-43.963	-36.657
Priv_room	-41.0791	1.863	-22.045	0.000	-44.731	-37.427
Shared_r	-41.4783	1.863	-22.260	0.000	-45.131	-37.826

Omnibus:	8088.001	Durbin-Watson:	2.003
Prob(Omnibus):	0.000	Jarque-Bera (JB):	39467.598
Skew:	1.179	Prob(JB):	0.00
Kurtosis:	7.987	Cond. No.	5.22e+18

Figure 9: OLS model output

The OLS model we built showed an R^2 value of 0.515, indicating that 51% of the variation in the "price" variable can be explained by the independent variables included in the model. This R^2 value provides us with an insight into the goodness of fit of the model and helps us assess the accuracy of our predictions. A value of 0.515 indicates that the model has moderate predictive power, meaning that it is able to explain a substantial portion of the variability in the target variable, but there may still be room for improvement.

The P-value for our model is zero, which means that the null hypothesis is rejected and the results of the test are statistically significant. This means that the results obtained from

the model are not due to random chance or chance factors and that the relationship between the independent variables and the target variable is real and significant. The P-value provides us with an objective way to assess the statistical significance of the results, which helps us draw reliable and accurate conclusions from the data.

Below we can find some interesting insights from the OLS model summary:

- The longer the minimum number of nights required for a stay, the lower the price tends to be.
- If the property is located on Manhattan and it is an entire home or apartment - the price changes the least compared to other property types and neighborhoods.
- The longer the length of the property title - the price is higher.
- Listing a property by male results in higher price.

In conclusion, all the independent variables were found to be statistically significant, except for *reviews_per_month*, whose p-value exceeded 5%. This implies that the effect of "reviews per month" on the target variable is not significant enough to reject the null hypothesis. As a result, this variable may not have a meaningful impact on the target variable and could potentially be removed from the model in future analyses.

3.4 Supervised learning models comparison

Supervised learning models are a type of machine learning algorithm that are trained on a labeled dataset, where the target variable is known. The goal of supervised learning is to build a model that can predict the target variable based on the input features. Examples of supervised learning algorithms include linear regression, logistic regression, decision trees, random forests, and support vector machines. The model learns from the labeled data to make predictions on new, unseen data.

In our final comparison, we evaluate the performance of three different supervised learning models: a linear model based on the local outlier factor (LOF), a decision tree model, and support vector machines (SVM). To compare their performance, we use two commonly used error metrics, mean absolute error (MAE) and root mean square error (RMSE). Both metrics give us an understanding of how well the models are able to predict the target variable, with

lower values indicating better performance. By comparing the performance of these models using these metrics, we can determine which model is the best fit for our data.

Linear model (LOF). The target value is expected to be a linear combination of the features. We minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

Decision Tree with max_depth=5. The goal of the Decision Tree model is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The idea behind using a maximum depth is to prevent the tree from growing too large and becoming overly complex, which can lead to overfitting the data and poor generalization performance on new, unseen data.

SVM with kernel="rbf". The SVM algorithm tries to find the hyperplane that maximizes the margin between the closest data points in the training set, which are known as support vectors. We used radial basis function kernel, which is more complex and efficient compared to linear or polynomial.

Error measure	LM	Decision Tree	SVM
MAE	61.540	61.141	80.402
RMSE	215.349	214.775	228.201

Table 4: Supervised learning models comparison

Based on MAE and RMSE Decision Tree model outperformed both the Linear model and the SVM model. While the linear model was slightly inferior to the Decision Tree model, the SVM model performed significantly worse.

Both the Linear and Decision Tree models have their own advantages and disadvantages, and the choice ultimately depends on the specific use case and requirements. The Linear model is simpler and easier to implement, but may not always produce the most accurate results, especially if the data has a non-linear relationship. On the other hand, the Decision Tree model offers more flexibility and can capture non-linear relationships in the data, but may result in overfitting if not properly managed. It's important to carefully consider the trade-offs and make an informed decision based on the data and the desired outcome.

4 Conclusion

Within this analysis some observations about the situation on the real estate rental market in 2019 in NYC were made. The main and most important one is undoubtedly the fact that Manhattan and then Brooklyn, more specifically the borderland of those two boroughs, are the locations where highest prices are observed. Hosts can expect average prices of 161 US$ per night when renting out an apartment in Manhattan, which is 50 US$ more than in Brooklyn and two times more than in the cheapest district - Bronx. It also turns out that hosts who have more listings can count on higher price per night. Perhaps, renting out the room or apartment for a month can be of benefit, as there is a relatively big group of hosts who require minimum 30 nights of rental time. Also it appears that the more a listing is available throughout the year the more reviews it tends to get. And as positive reviews can be an incentive for booking[2] it may be a good strategy to make the property available for booking for more days throughout the year.

Additionaly, the aim of this analysis was to perform build a regression model that predicts Airbnb rates based on apartment characteristics. The process started with selecting the predicted variable, *price*, and gathering and cleaning the data. Several transformations were applied to the data, such as converting the *price* values into natural logarithm, transforming categorical variables, enhancing the host name information and etc.

Outlier detection was also an important aspect of this analysis, and various automatic outlier detection methods were applied to improve the accuracy of the model. The importance of detecting and removing outliers in a dataset before training a machine learning model cannot be overstated. By doing so, we are able to ensure that the model is trained on accurate and reliable data, leading to more accurate predictions. The Local Outlier Factor method was found to be the most effective outlier detection method in this study.

The comparison of three supervised learning models was made: Linear (LOF), Decision Tree, and Support Vector Machine (SVM). Based on the error metrics, the Decision Tree outperformed both the Linear model and SVM, while the Linear model was slightly inferior to the Decision Tree and the SVM performed significantly worse. The choice between the models depends on the specific use case and requirements, with the Linear model being simpler and easier to implement, but not always producing the most accurate results, and the

[2]E. Masłowska, E.C. Malthouse, S. F. Bernritter, *Too Good To Be True: The Role of Online Reviews' Features in Probability to Buy.* International Journal of Advertising 2017

Decision Tree offering more flexibility but prone to overfitting.

The results of this analysis provide valuable insights into the relationship between Airbnb rates and apartment characteristics and can be used to make data-driven decisions.