Business Intelligence Assignment 2

Noah Weidenhaupt (B)
12327501
Technische Universität Wien
Vienna, Austria

Felix Kempe (A)
12327971
Technische Universität Wien
Vienna, Austria

Abstract

This Paper presents a data-driven approach to predicting rental prices of AirBnB listings, based on the AirBNB Open Dataset 2023 for the United States. After initial dataset exploration, several preprocessing steps to address missing values, remove possibly sensitive attributes (host_name) , convert neighborhood information into ZIP-codes and apply threshold-based outlier filtering are performed. For the actual regression modeling, a linear regression model and a RandomForestRegressor are compared. Because of weak linear relationships present in the data, the linear regression model presented comparativley limited accuracy, while the RandomForestRegressor achieved superior performance and was thus chosen as the primary model for this paper. By applying thresholds to columns like "reviews_per_month", price, and feature selection using methods like "ANOVA" and a correlation Matrix, the RandomForestRegressor is able to achieve a R^2 score of 0.64 and a Mean Squared Error (MSE) of 3650. The results proof that it is possible to explain a substantial amount of the variance in price, based on limited public data.

Keywords

CRISP-DM, Pandas, AirBnB Dataset, Dataset Exploration, RandomForestRegressor

ACM Reference Format:
Noah Weidenhaupt (B) and Felix Kempe (A). 2018. Business Intelligence Assignment 2. In *Proceedings of Assignment 2 (Business Intelligence).* ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Business Understanding

Following chapter will be guided by CRISP DM Framework and focuses on business understanding.

1.1 Definition Datasource

The Dataset is a collection of AirBnB listings from the united states across different cities and states for the year 2023. It contains 232,147 rows and 18 attributes and provides detailed information about individual apartment listings and hosts. The dataset contains necessary information to analyze trends and pricing patterns, making data-driven decisions possible. The given dataset is useful in a scenario involving rental price optimization to maximize revenue for hosts. An analysis can possibly identify relationships between neighborhood popularity and pricing, allowing for the development of custom dynamic pricing based on the listings features.
Below you will see the datasets structure, including attribute name, main type, datatype and description.

Table 1: Dataset Structure (Transposed)

Attribute	Description
id	Unique listing ID (key).
name	Name of the listing.
host_id	Unique host ID (key).
host_name	Name of the host.
neighbourhood_group	Group in which the neighborhood lies.
neighbourhood	Name of the neighborhood.
latitude	Latitude of the listing.
longitude	Longitude of the listing.
room_type	Type of the room.
price	Price of the listing per night.
minimum_nights	Minimum number of nights required to book.
number_of_reviews	Total number of reviews on the listing.
last_review	Date of the last review.
reviews_per_month	Average reviews per month.
calculated_host_listing_count	Total number of listings by the host.
availability_365	Number of days in a year the listing is available.
number_of_reviews_ltm	Total reviews by the listing in its lifetime.
city	City of the listing.

1.2 Business Objectives

The company's goal is the prediction of rental prices for airbnb listings. The purpose of rent-price optimization is to develop an effective rent strategy based on the popularity and demand of the districts. Relate the popularity of the districts to the price. From this, prices can be derived that maximize revenue while maintaining competitiveness.

1.3 Business Success Critera

The company's success criteria are three main criteria. Firstly, a measurable increase in revenue for the AirBnB platform or the hosts who provide the accommodation. The second criterion is an

Business Intelligence, December 15th, 2024, Vienna, Austria

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

increase in the average occupancy rates of the accommodation. Finally, a high level of customer satisfaction, which is reflected in positive feedback through reasonable prices and more bookings. In summary, this means an increase in revenue, higher occupancy of accommodation and high customer satisfaction.

1.4 Data Mining Goals

A central objective of data mining is defined in order to achieve the company's goals. With regard to price optimization, a regression analysis is used to model the relationship between the characteristics of the offers and the rental prices.

1.5 Data Mining Success Criteria

The rent optimization model is evaluated using regression metrics such as the R^2 value. A high R^2 value (e.g., > 85%) indicates that the model explains a large proportion of the variance in rental prices. Additionally, the Mean Squared Error (MSE) is used. A low MSE demonstrates that the model provides accurate price predictions. Another success factor is the successful integration of the results into the rent prediction model.

2 Data Understanding

2.1 Attribute types and semantics

The attributes, shown in Table 2, form a basis for analyzing AirBnB listings in the US and provide insights into the geographic distribution, pricing and activity levels of hosts and guests.

Table 2: Dataset Structure

Attribute	Main Type	Type
id	Nominal	Integer
name	Nominal	String
host_id	Nominal	Integer
host_name	Nominal	String
neighbourhood_group	Nominal	String
neighbourhood	Nominal	String
latitude	Ratio	Float
longitude	Ratio	Float
room_type	Nominal	String
price	Ratio	Integer
minimum_nights	Ordinal	Integer
number_of_reviews	Ratio	Integer
last_review	Interval	Date
reviews_per_month	Ratio	Float
calculated_host_listing_count	Ratio	Integer
availability_365	Ratio	Integer
number_of_reviews_ltm	Ratio	Integer
city	Nominal	String

2.2 Statistical Properties

The descriptive statistics for selected numerical attributes in the dataset are shown in Table 3:

Table 3: AirBnB2023 Statistical Properties (Part 1)

Column	Mean	Max	Min
latitude	36.61	47.73	25.96
longitude	-98.3	-71.0	-123.09
price	259.47	100000	0
minimum_nights	13.5	1250	1
number_of_reviews	40.92	3091	0

Table 4: AirBnB2023 Statistical Properties (Part 2)

Column	Variance	Std
latitude	26.28	5.13
longitude	388.36	19.71
price	1,049,899.26	1,024.65
minimum_nights	779.56	27.92
number_of_reviews	6,504.29	80.65

The statistical properties of the data set reveal some anomalies that provide initial insights into the data quality. Some listings have very high prices (100,000) or unusually high minimum stays (1,250 nights), which could be outliers. In addition, there are listings with extremely high ratings or very many ratings per month, which are probably the most popular or strategic listings. The high number of listings per host could indicate a market that is heavily dominated by professional providers. The fluctuations in availability and number of reviews reflect a variety of usage patterns. These anomalies will be analyzed in the further course of the data quality analysis.

2.3 Data Quality

The analysis identified missing values in several attributes. The attribute **neighbourhood_group** has the highest percentage of missing entries, with 58.43% of values missing. **last_review** and **reviews_per_month** each have 21.14% missing values, while other attributes such as **name** and **host_name** show minimal missing values (0.01%). Attributes such as **id**, **host_id**, **latitude**, **longitude**, **room_type**, and **price** have no missing values. Reference: See the section "Check for Missing-Values" in the Notebook: Data_Exploration.

Inconsistencies in data types are detected in the following attributes:

- **name**: Two unique data types.
- **host_name**: Two unique data types.
- **neighbourhood_group**: Two unique data types.
- **last_review**: Two unique data types.

These inconsistencies indicate a mixture of expected data types (e.g., string and integer values or other formats). Reference: See the section "Check for Datatype Inconsistencies" in the Notebook: Data_Exploration.

Outliers are identified using the Z-score method:

- **price**: Extreme values were detected, ranging from 3334 to 100,000.
- **minimum_nights**: Outliers range from 98 to 1250 nights.

- **latitude**: Values outside the valid range, with extreme Z-scores up to 34,204,294,013.
- **longitude**: Values below the valid range, with extreme Z-scores down to -12,238,400,269.
- **number_of_reviews**: Outliers range from 283 to 3091.
- **reviews_per_month**: Outliers range from 706.0 to 10,142.0.
- **calculated_host_listings_count**: Outliers range from 356 to 1003.
- **availability_365**: No outliers detected.
- **number_of_reviews_ltm**: Outliers range from 74 to 1314.

Most of the outliers are data that is actually correct real world data, such as where the **price** has a value of 100,000. With **price**, a value of 0 or 1 makes no sense and is handled in the Data Preperation. The other outliers recognized by the Z-Score are also real world data, so they are identified as non-outliers. The identification of outliers is nevertheless of interest in order to better understand the data and also to identify outliers as non-outliers.
Reference: See the section "Z-Score based extreme value detection" in the Notebook: Data_Exploration.

The following inconsistencies and plausibility issues are shown below:

- **ID**: Duplicate entries detected, e.g., `['6,04269E+17', '6,24238E+17']`.
- **host_name**: Inconsistent host names for identical host IDs, e.g., `[14269516, 24...]`.
- **neighbourhood**: Numeric values `[Non-numeric values: ['Western Addition' 'Berna...]]`.
- **price**: Non-positive values detected, including `[0]`.

The identified inconsistencies in the dataset can lead to various issues that significantly affect the quality of analyses and modeling. For example, duplicate entries in the ID field cause data redundancy, which overestimates the availability or popularity of certain listings and skews subsequent analyses such as clustering or regression. Additionally, inconsistent host names for identical host_id values make it difficult to correctly associate listings with hosts, potentially leading to inaccurate analyses of host activity.Faulty data formats, such as numeric values in the neighbourhood field where textual information is expected, present another challenge. Such deviations can disrupt clustering and grouping by neighborhoods. Furthermore, invalid values such as non-positive prices (value of 0) negatively impact analyses, as they are unsuitable for price predictions or optimization models.
Reference: See the section "Check for inconsistencies based on various rules and display examples" in the Notebook: Data_Exploration, for all inconsistency checks.

2.4 Visual Exploration

The price distribution shows a strong right skew, indicating the presence of extreme outliers with very high prices (Z-score also refers to these outliers, for treatment of these, see section Z-score). However, most properties are in the lower price range, indicating a general affordability of the offers. The same is true for the minimum length of stay, with many properties requiring only 1-2 nights, making them particularly attractive for short-term stays. The number of reviews is also strongly skewed to the right. Many properties have few reviews, indicating that they are either new or rarely booked. Availability across the year, on the other hand, shows two clear peaks: many properties are either available all year round or not bookable at all, with a smaller, even share of properties that are only occasionally available. The correlations between the numerical variables are very weak overall. There is a slight positive correlation between price and availability_365, while number_of_reviews is weakly negatively correlated with minimum_nights. In terms of neighborhood groups, "City of Los Angeles" and "Manhattan" clearly dominate, while smaller neighborhoods are only marginally represented. In terms of room types, there is a clear preference for "Entire home/apt", followed by "Private room". "Shared room" and "Hotel room" play a subordinate role, which indicates lower demand. Looking at the cities, "New York City" and "Los Angeles" dominate by far, while smaller cities such as "Cambridge" or "Salem" only make up a small proportion of the offers.
Reference: See the section "Visual exploration" in the Notebook: Data_Exploration, for all visualizations.

2.5 Ethically Sensitive Data

The analysis and evaluation of the data and visualizations shows that several attributes are potentially ethically sensitive or prone to bias, as well as having unbalanced distributions or underrepresented groups. The attributes neighborhood and neighborhood_group are potentially sensitive because they can reveal social and demographic differences between neighborhoods, which could encourage discrimination. Likewise, host_name can allow conclusions to be drawn about ethnic or cultural identities, which could promote data protection issues or bias in models. The distribution of the data is unbalanced for room_type and neighborhood_group, with an overrepresentation of certain categories (e.g., Entire home/apt and Manhattan) and a strong underrepresentation of other groups. This could lead to biases in the modeling and must be taken into account. Numerical variables such as price, minimum_nights and number_of_reviews also show strong outliers or uneven distributions, which could affect the analysis. However, it was determined that these outliers are real world data.

2.6 Risks and biases

The dataset has potential risks and sources of bias that can affect both the fairness of the analysis and the data quality. Ethically sensitive attributes such as neighborhood, neighborhood_group and host_name could encourage discrimination or pose privacy risks. Additional bias arises from representation bias, as categories such as entire home/apt or cities such as New York City are overrepresented and smaller groups and neighborhoods are severely underrepresented. This can promote algorithmic bias that disadvantages rare groups and distort the results of the analysis. Historical bias could also exist, as existing inequalities between neighborhoods or socioeconomic groups are unconsciously reproduced. Numerical outliers in price, minimum_nights and number_of_reviews further increase bias, as extreme values could dominate the analysis. Risks include excluding underrepresented groups, misusing sensitive information or unfairly optimizing models. External experts should answer questions about data collection, quality assurance and ethical defensibility. Example questions might be: "Were underrepresented groups deliberately excluded, or is this random bias?" and "What steps have been taken to identify historical inequalities and algorithmic bias?" Measures such as oversampling underrepresented groups, transforming extreme values, and targeted weighting can help reduce bias and improve data quality.

2.7 Data Preparation Actionplan

Data preparation is based on the conclusions of the data analysis, taking into account both the identified ethical risks and potential biases. Several steps are taken to improve data quality and minimize biases. First, the sensitive attributes host_name and neighborhood_group are removed to reduce privacy risks. Neighborhood names are replaced with zip codes to ensure a standardized and anonymized representation. Missing values in the last_review and reviews_per_month columns are replaced with the value 0 to ensure data consistency. In addition, duplicate IDs are cleaned to avoid redundancies and entries with a price below $5 per night are removed to exclude implausible or erroneous values.

3 Data Preparation

3.1 Necessary pre-processing actions

The following Section will outline all pre-processing steps that were taken.

3.1.1 Drop Host-name. The column 'host_name' was dropped for two specific reasons. Firstly, there already is a unique identifier for each host with the column host_id. Secondly, the host_name column is potentially ethically sensitive, as the cultural background of the host could be potentially identified via the name.

3.1.2 Drop Neighborhood-Group. The column 'neighborhood_group' has over 130.000 missing values and it is not possible to fill the missing values with sensible content. In addition, this column is not relevant for the price prediction, as it is planned to use the ZIP-code provided through the Longitude / Latitude columns.

3.1.3 Encode Neighborhood names into ZIP-code. The original 'neighborhood' column constists of a mixed type. It is split between cleartext and zipcodes. For the purpose of further analysis, it is decided to encode all the cleartext into zipcodes using the longitude and latitude columns provided in the dataset. To calculate the Zipcodes, a Digit ZIP Code Tabulation Area (ZTCA5), from the United States census Bureau, in combination with geopandas was used.
Reference: See the Notebook: ZIP-Code encoding.

3.1.4 Replace Nan 'last review' / 'reviews per month' with 0. Earlier Data exploration showed that missing 'last review' / 'reviews per month' values are related to the total reviews of the listing. If the total reviews are 0, the values for 'last review' / 'reviews per month' remain empty. To fill the empty values, they were replaced with 0.

3.1.5 Fix duplicate ID. An instance of a duplicate ID was discovered during the data-exploration process. After closer investigation it turned out that the duplicate ID was linked to a faulty listing. The Faulty listing was dropped from the dataset to return the column 'id' back into a unique identifier

3.1.6 Remove all prices per night under 5$. Listings below 5$ were classified as outliers / faulty data and thus removed.
Reference: All steps are documented in the Notebook: Pre_processing.

3.2 Considered pre-processing steps (Not Applied)

It was considered to investigate the column 'availability_365', because it displayed and unusual amount of '0'

3.3 Options and potential for derived Attributes

During the pre-processing steps it was theorized that a column for a derived attribute called 'neighborhood_popularity could be derived from the columns number_of_reviews, reviews_per_month and availability_365, to summarize the relative attractiveness for a specific neighborhood. However, the derived attribute remained theoretical and was not implemented at this stage.

3.4 Options for external data sources

While consolidating the Business and Data Mining objectives it was theorized that additional consumer data could be useful in determining user preferences and improve the planned price optimization. However, finding relevant data, that is applicable to the current context proved to be very challenging and was not implemented at this stage.

4 Modeling

4.1 Data Mining Algorithms

For the price prediction problem, Linear Regression and a RandomForestRegressor are considered. Linear Regression is a basic and interpretable model, that serves as a baseline predictor and to help identify if strong linear relationships are present between features. In contrast, the RandomForestRegressor is used, because it is capable of handling higher feature complexity and non-linear interactions.
Initial tests of both models show that the random forest regressor is consistently superior to the linear regression model, which can be seen in the “Modeling_Regression” notebook under sections (f) Pipeline (split, cross-validation) and (g) Identifying subgroup bias. A view of the Linear Regressions accuracies, including threshold boundaries, which are outlined in the following chapter, is listed below.

Table 5: Pivoted Model Performance Results for Linear Regression

Metric	Training	Test	Training	Test
Thresholds	No	No	Yes	Yes
MSE	1,227,773.56	1,601,607.34	7,964.68	7,707.70
R^2	0.00	0.01	0.26	0.25

These poor performances indicate that there is a lack of linear relationship in the data. Subsequently, a non-linear regressor, like the previously mentioned RandomForestRegressor is chosen as the primary model for further testing and fine-tuning.

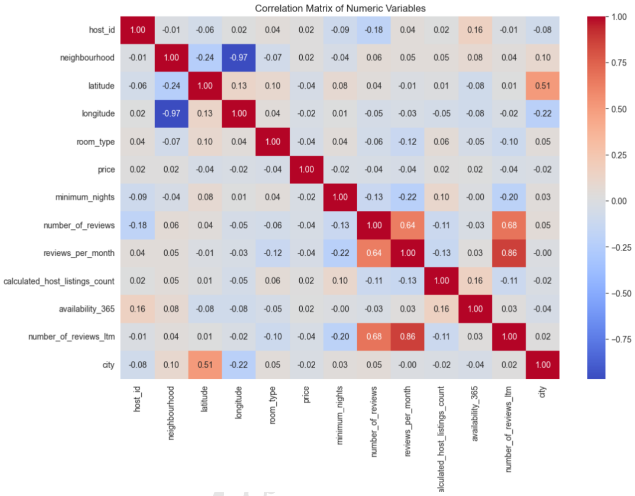
4.1.1 Feature Selection & Thresholds. Upon further testing, it is discovered that outliers within the data set prove to be a significant issue with respect to model performance. The Outliers in questions technically are valid data, but still prove to be a significant issue when training the model. To combat these challenges thresholds on selected variables are implemented.

To determine the threshold values, box plots are created with the variables listed below, whereby the whiskers of the plots are used as outlier thresholds, which can be seen in the six box plots in the appendix (see Figures 3 to 8). It is decided to set a threshold value for the selection of the 6 variables below, as the analysis refers to all locations and room types. Therefore, the variables for location and 'room_type' are not included in the thresholds. The creation of box plots can be found in the "Modeling_Regression" notebook under the section (d) Box-plots, within the function `visualize_distributions_with_whiskers(data)`. A complete list of all thresholds is shown below:

```
thresholds = {
    'reviews_per_month': (0,4.9),
    'number_of_reviews': (0, 106),
    'availability_365': (0, 365),
    'number_of_reviews_ltm': (0, 40),
    'minimum_nights': (1, 72),
    'price': (5, 488.5)
}
```

To compare performance between the threshold and non-threshold approach, the data is applied, with and without thresholds in each training and validation cycle on the models. Implementing these feature threshold increased accuracies on the linear regression model from 0.01 to around 0.25 and doubled the accuracy on the RandomForestRegressor.

To determine the optimal feature selection on which to train the models, three different methods are used: a correlation matrix, ANOVA, and lastly RFE. This process is detailed in the "Modeling_Regression" notebook under the sections (b) ANOVA Analysis, (c) Correlation Matrix and RFE, and is implemented in the respective functions `create_correlation_matrix(data)`, `perform_ANOVA(data)` and `apply_RFE(data)`. A correlation matrix calculates the linear relationship between two variables. It creates a matrix that depicts a correlation coefficient for each pair of variables. This correlation coefficient determines the strength and direction of the relationship and ranges from -1 to 1, where values close to 0 suggest that there is no linear relationship between the variables. As outlined by the figure below, the correlation coefficients related to price are all close to 0, which means that there is no significant linear relationship between a single variables and price, explaining the poor performance of the linear regression model.



	host_id	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews	reviews_per_month	alculated_host_listings_count	availability_365	number_of_reviews_ltm	city
host_id	1.00	-0.01	-0.06	0.02	0.04	0.02	-0.09	-0.18	0.04	0.02	0.16	-0.01	-0.08
neighbourhood	-0.01	1.00	-0.24	-0.97	-0.07	0.02	-0.04	0.06	0.05	0.05	0.08	0.04	0.10
latitude	-0.06	-0.24	1.00	0.13	0.10	-0.04	0.08	0.04	-0.01	0.01	-0.08	0.01	0.51
longitude	0.02	-0.97	0.13	1.00	0.04	-0.02	0.01	-0.05	-0.03	-0.05	-0.08	-0.02	-0.22
room_type	0.04	-0.07	0.10	0.04	1.00	-0.04	0.04	-0.06	-0.12	0.06	-0.05	-0.10	0.05
price	0.02	0.02	-0.04	-0.02	-0.04	1.00	-0.02	-0.04	-0.04	0.02	0.02	-0.04	-0.02
minimum_nights	-0.09	-0.04	0.08	0.01	0.04	-0.02	1.00	-0.13	-0.22	0.10	-0.00	-0.20	0.03
number_of_reviews	-0.18	0.06	0.04	-0.05	-0.06	-0.04	-0.13	1.00	0.64	-0.11	-0.03	0.68	0.05
reviews_per_month	0.04	0.05	-0.01	-0.03	-0.12	-0.04	-0.22	0.64	1.00	-0.13	0.03	0.86	-0.00
calculated_host_listings_count	0.02	0.05	0.01	-0.05	0.06	0.02	0.10	-0.11	-0.13	1.00	0.16	-0.11	-0.02
availability_365	0.16	0.08	-0.08	-0.08	-0.05	0.02	-0.00	-0.03	0.03	0.16	1.00	0.03	-0.04
number_of_reviews_ltm	-0.01	0.04	0.01	-0.02	-0.10	-0.04	-0.20	0.68	0.86	-0.11	0.03	1.00	0.02
city	-0.08	0.10	0.51	-0.22	0.05	-0.02	0.03	0.05	-0.00	-0.02	-0.04	0.02	1.00

Figure 1: Correlation Matrix without thresholds

ANOVA is deployed as a secondary method to identify variables with a strong relationship to price, it evaluates the statistical significance of differences in mean values _"add more detailed description". This second approach is able to achieve more significant findings in comparison to the correlation matrix. Most of the variables outlined here are selected for training the model, as they provided the best performance.

```
     Feature        Score
8              reviews_per_month  411.398748
3                       latitude  404.204394
5                      room_type  356.665499
7              number_of_reviews  288.303007
11         number_of_reviews_ltm  284.846754
2                  neighbourhood  140.955690
4                      longitude  125.613623
6                 minimum_nights  122.318169
12                          city  100.056149
1                        host_id   94.928926
9   calculated_host_listings_count   82.050186
0                             id   68.027772
10              availability_365   56.144169
```

c

- 'latitude'
- 'room_type'
- 'reviews_per_month'
- 'longitude'
- 'neighbourhood'
- 'number_of_reviews_ltm'
- 'availability_365'
- 'number_of_reviews'
- 'minimum_nights'

4.2 Train / Validation

To enable accurate hyperparameter tuning the code relies on 10-fold cross-validation. As the given dataset is quite large with roughly 300.000 entries, cross-validation on the entire dataset would take significant time. To overcome these time-constraints, the inital data is down-sampled to 10% of its original size (30.000 entries) using stratified_split (see Notebook: Modeling_Regression, def process_pipeline, def stratified_split). This stratified_split is used to ensure that the downsamples subset still maintains the a representitive distribution of the target variable. In other words, the target variable distribution in the 10% subset, mirrors the distribution of the original dataset. Due to computational constraints, only this 10% sample of the original dataset is used for this analysis. The down-sampled subset is then split into 80% training and 20% validation data, which is used in the subsquent cross-validation for hyperparemeter tuning.

4.3 Hyperparameters & Scaling

The RandomForestRegressor offers a variety of hyperparameters that can influence the performance. Most common Parameters include:

- n_estimator: Number of trees in the forest
- max_depth: Maximum depth of each tree.
- min_samples_split: Minimum number of samples required to split an internal node.

As previously mentioned, a 10-fold cross-validation is then performed over these paremeters using GridSearchCV. GridSearchCV is a sklearn function used for tuning hyperparameters, by training and evaluating a machine learning model using different combination of hyperparameters. The specific hyperparameters can be defined using a dictionary to limit the scope of the cross-validation, in addition it is possible to set the number of cross-validations using the "cv" parameter, which in this case is 10 (see Notebook: Modeling_Regression, def process_pipeline, def cross_validate_and_train).

```
param_grid = {
'n_estimators': [50, 100,  200],
'max_depth': [None, 5, 10, 15],
'min_samples_split': [2, 5, 8]
}
```

To evaluate the models performance, R^2 and MSE are used as metrics. R^2 describes the proportion of variance in the dependent variable that is predictable from the independent variables, making it a relevant metric to the current scenario. MSE serves to measure the average of the squared differences between predicted values and target values, placing a higher weight on larger errors, making this metric sensitive to outliers. After performing the cross-validation the best combination of hyperparameters identified by GridSearchCV, for the RandomForestRegressor, using threshold boundaries is:

```
n_estimators: 200
max_depth: None
min_samples_split: 8
```

In addition, scaling is performed using a RobustScaler, this ensures that outliers have less influence on the models performance. This step improved accuracy even with the previously mentioned threshold boundaries already in-place.

4.4 Training the model

With the identified hyperparameters and thresholds applied, the RandomForestRegressor achieved the following results.

- Training set (cross-validation process): R^2 = 0.66, MSE = 3715.35
- Test set: R^2 = 0.64, MSE = 3649.87

The following chapter explains the performance of the model and its evaluation in more detail.

5 Evaluation

5.1 Final Performance

The final model used to evaluate both training and test data is the Random Forest regression model (also see Notebook: Modeling_Regression, def process_pipeline, def cross_validate_and_train). Two scenarios are distinguished: the application of the model without considering thresholds and the application of the model with thresholds to remove outliers. The results of the models are summarized in the following table:

Table 6: Pivoted Model Performance Results

Metric	Training	Test	Training	Test
Thresholds	No	No	Yes	Yes
Max_depth	None	None	None	None
Min_samples	8	8	8	8
N_estimators	100	100	200	200
MSE	882,913.01	1,066,854.04	3,715.35	3,649.87
R^2	0.28	0.34	0.66	0.64

The results demonstrate that the consideration of thresholds to remove outliers significantly improved model performance. While the R^2 value without thresholds is 0.34 on the test data, it increased to 0.64 with the application of thresholds. This indicates a clear improvement in model performance. However, the R^2 value of 0.64 remains below the ideal value of 1 for a perfect model and explains only about two-thirds of the variance in the price, highlighting the model's limitations in price prediction. Additionally, the MSE decreased. Without thresholds, the MSE on the test data is 1,066,854.04, while it drops to 3,649.87 after removing outliers. This reduction reflects the improved prediction accuracy of the model after outlier removal.

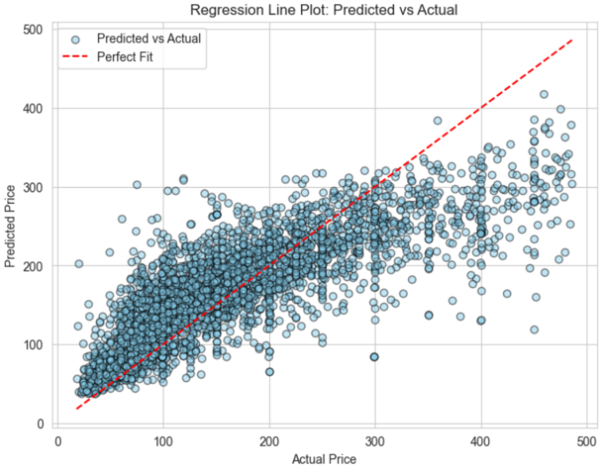


Figure 2: RandomForestRegressor with Threshold boundaries

The scatter plot shown in Figure 2: RandomForestRegressor with Threshold boundaries visualizes the relationship between the actual rental prices (x-axis) and the predicted prices from the Random Forest regression model (y-axis). The red dashed line represents the perfect prediction, where the predicted price exactly matches the actual price. The scatter points indicate that the model generally approximates the actual prices well, especially in the lower price ranges (up to around \$300), where the data is more densely distributed. However, deviations become more noticeable at higher prices (>\$300), suggesting increased complexity in the price structure. Individual points above or below the line highlight systematic underestimations or overestimations by the model.
Overall, the plot shows that the model is capable of capturing the general trend of rental prices but does not fully account for all price variations. This aligns with the previously reported R^2 value, which explains 64% of the variance, indicating room for further optimization.

5.2 Re-train

Retraining the model with the entire training dataset is unnecessary, as this was already performed during hyperparameter optimization using 10-fold cross-validation, as described in Chapter 5.2. This method splits the entire training data into 10 folds and ensures that each fold is used for both training and validation. Therefore, the results presented in the table represent both the performance on the training data and the model's generalization capability on the test data. The robust evaluation through cross-validation reduces the likelihood of overfitting and demonstrates the efficiency of the Random Forest model in price prediction, especially after removing outliers.

5.3 State of the Art performance

The paper *"Machine Learning Prediction of New York Airbnb Prices"* and the present analysis are based on the same underlying dataset but differ in their geographical focus, choice of predictors, and preprocessing steps [1]. While the paper exclusively examines listings from New York City, the present analysis includes offerings from multiple cities across the United States. This broader geographical coverage results in greater heterogeneity in the data and potentially more complex relationships between variables. A key distinction lies in the choice of predictors. The paper utilizes the computed variable *sentiment_score*, derived using Natural Language Processing (NLP) from the title descriptions of listings. In contrast, the present analysis relies solely on variables directly sourced from the dataset and removes outliers through defined thresholds. Predictors include *latitude*, *longitude*, *room_type*, *reviews_per_month*, *number_of_reviews*, *availability_365*, and *minimum_nights*. In terms of modeling results, the Random Forest model from the paper achieves an R^2 value of 0.612, whereas the present analysis, after outlier removal, achieves an R^2 value of 0.64 and a MSE of 3,649.87. Despite the heterogeneity of the data and the absence of derived variables like *sentiment_score*, the present analysis demonstrates comparable or even slightly better model performance. These results emphasize that the approach of the present analysis delivers competitive performance despite the challenges posed by broader data diversity and the lack of computed predictors. The careful preprocessing, particularly the removal of outliers, significantly contributes to the improvement in model quality.
The baseline performance of trivial models is determined to provide a foundation for evaluating a more complex regression model (see Notebook: Modeling_Regression, def process_pipeline, def evaluate_model_on_datasets). Two simple baselines are considered: the mean predictor and the median predictor of the target variable *price*. The mean predictor, which always predicts the mean of the target variable, achieves a MSE of 1,050,234.05 and an R^2 value of -0.00. This serves as a typical baseline for measuring prediction accuracy. The median predictor, which always predicts the median of the target variable, achieves an MSE of 1,062,456.42 and an R^2 value of -0.01. This performance is slightly worse than that of the mean predictor, indicating that prediction accuracy using the median is less precise for this dataset.
To assess the added value of the more complex model, the baseline performances were first established in the previous chapter. These serve as a starting point for comparing with the performance of the Random Forest regressor. In the next step, the performance of the Random Forest model after applying outlier thresholds is analyzed and compared to the baseline results. The goal is to evaluate the prediction accuracy of the model using various metrics. After the removal of outliers, the model achieved a MSE of 3,649.87 and an R^2 value of 0.64 on the test data. The baseline performances, defined by the mean predictor and median predictor, were an MSE of 10,233.31 (R^2 = -0.00) and 10,801.75 (R^2 = -0.06), respectively. The model significantly outperformed the baseline performances, reducing the MSE by 6,583.44 compared to the mean predictor and by 7,151.88 compared to the median predictor. The substantial improvement in model performance through the application of outlier thresholds is also reflected in the variance explained. While the baseline models could not explain any variance ($R^2 \approx 0$), the Random Forest model was able to capture 64% of the variance in the target variable. This underscores the effectiveness of data preparation to improve model accuracy.

5.4 Performance compared to success criteria

In the subsequent analysis, the performance of the Random Forest regression model is evaluated based on the success criteria defined in Chapter 1.5 for predicting rental prices. These criteria include a high explanatory power (R^2) and a low prediction error measured by the MSE. The goal is to develop a model that enables precise price predictions to support strategic decision making. After applying outlier thresholds, the model achieved an R^2 value of 0.64 on the test data, indicating that 64% of the variance in rental prices can be explained by the model (see Chapter 6.1). This demonstrates that the model is already capable of predicting a substantial portion of the target variable. However, the value falls short of the target of 0.85 defined in Chapter 1.5, indicating further optimization potential. The error metric analysis revealed a MSE of 3,649.87 on the test data, which represents a significant improvement compared to the training data (3,715.35). This shows that the model generalizes well and achieves a solid prediction accuracy. Nonetheless, a lower MSE would be desirable to further improve the precision of the prediction and better meet the requirements outlined in Chapter 1.5.

5.5 Protected attributes

The attribute *room_type* is selected as a protected attribute, because it describes the type of accommodation and defines the subgroups Entire home/apt, Hotel room, Private room, and Shared room. The objective of this analysis was to evaluate the model's performance for each subgroup and identify potential biases in the model. The results reveal significant differences in model performance across subgroups:

- **Entire home/apt (Subgroup 0)**: R^2 value of 0.01, MSE of 1,285,429.75. The model explains almost no variance, indicating low prediction accuracy.
- **Hotel room (Subgroup 1)**: R^2 value of 0.75, MSE of 274,114.85. This subgroup demonstrates the best model performance with 75% of the variance explained and a significantly lower MSE.
- **Private room (Subgroup 2)**: R^2 value of 0.18, MSE of 1,348,762.76. The model performance is slightly better but remains inadequate.
- **Shared room (Subgroup 3)**: R^2 value of -0.25, MSE of 108,878.18. The negative R^2 value indicates that the model performs worse than a trivial mean predictor.

The analysis highlights that the model exhibits a strong bias in favor of the Hotel room subgroup, while the subgroups Entire home/apt, Private room, and especially Shared room are considerably disadvantaged. Possible causes could include insufficient data representation for certain subgroups or high variance within the subgroups.

6 Deployment

6.1 Performance compared to business Objectives

The analysis of the results concerning the business objectives shows that the goal of optimizing rental prices has been partially achieved. The aim was to develop an effective rental strategy based on the popularity and demand of neighborhoods to derive competitive prices that maximize revenue.

The final model, a Random Forest Regressor, achieved an R^2 of 0.64 and an MSE of 3,649.87 (see Chapter 6.1). These results indicate that the model can explain approximately 64% of the variance in rental prices, providing a solid foundation. However, the R^2 value falls short of the originally targeted value of 0.85, indicating room for optimization. The MSE demonstrates that the model predictions are precise but not accurate enough for fully automated pricing.

The model uses variables such as latitude, longitude, room_type, reviews_per_month, and availability_365, which are key determinants of popularity and demand (see Chapter 4.1). Thus, the relationship between neighborhoods and prices is addressed through geographic data and demand-related variables, such as the number of monthly reviews. However, a derived attribute explicitly modeling neighborhood popularity, such as a popularity score based on factors like the number of reviews and availability, is missing. Integrating such a variable could enhance prediction accuracy and interpretability (Chapter 6.3). Moreover, external data sources, such as information on local events or seasonal trends, and temporal factors (e.g., weekend pricing) were not considered, which could further improve model performance.

Regarding the utilization of the results, a hybrid approach is recommended. Given the current prediction accuracy, fully automated pricing is not advisable. Instead, the model could generate price suggestions that managers or hosts review and adjust. Partial deployment of the model is feasible for well-represented neighborhoods, such as Manhattan or Los Angeles, where sufficient data exists, and model performance is higher. For underrepresented neighborhoods, separate data analysis should be conducted to avoid biases. Additionally, the model is suitable for specific segments, such as "Entire home/apt" or "Hotel room," where performance is above average. Poorly predicted categories, such as "Shared room," require additional models or methods (see Chapter 6.5).

To operationalize the results, a user-friendly dashboard could be developed to display price ranges based on model predictions. This would allow users to quickly evaluate suggestions and make manual adjustments if necessary. Additionally, the model should be regularly updated with new data to account for trends, such as changing demand or new listings.

For future analyses, integrating external data sources, such as local event information or competitor prices, is recommended to improve model performance. Sociodemographic data could also provide further insights into neighborhood popularity. Deriving a popularity score explicitly modeling the attractiveness of a neighborhood could improve not only prediction accuracy but also decision support. Furthermore, employing advanced algorithms such as Gradient Boosting (e.g., XGBoost) could further enhance model performance, as demonstrated in the literature "Machine Learning Prediction of New York Airbnb Prices" (see Chapter 6.3). Finally, temporal trends should be addressed using time-series analyses or seasonal dummy variables to capture seasonal effects more precisely.

6.2 Ethical impact assessment

When developing and deploying the rental price prediction model, potential ethical aspects, impacts, and risks must be considered.

A central ethical concern relates to the fairness and potential biases of the model. Since it relies on historical data, existing social inequalities or injustices could be inadvertently reproduced. For example, the overrepresentation of certain neighborhoods or accommodation types could systematically disadvantage underrepresented groups. Specifically, the use of variables such as `neighbourhood` or `room_type` carries the risk that prices in less popular neighborhoods may be disproportionately low or excessively high in sought-after areas, potentially affecting the competitiveness of certain offerings. Furthermore, privacy protection is essential. Although personal data such as `host_name` has already been removed, there remains a risk that geographic data like `latitude` and `longitude` could be used to infer individual persons or sensitive areas. Anonymization would be necessary to comply with data protection laws such as the GDPR. Another ethical issue involves the use of price optimizations. Higher prices in already popular neighborhoods could unintentionally contribute to gentrification, displacing socially weaker groups and exacerbating existing inequalities. To avoid this, the model should be designed to account for social impacts and maintain fairness. Additionally, transparency is crucial: hosts should be informed about the model's functionality and limitations to prevent misunderstandings or unfair decisions.

The impact analysis reveals both positive and potentially negative effects. Among the positive impacts is increased efficiency: the model can provide hosts with data-driven insights into pricing, leading to higher revenues and more efficient resource utilization. Dynamic price adjustments enable hosts to better respond to local demand fluctuations, strengthening competitiveness and potentially improving guest and host satisfaction in the long term. However, there are also negative impacts, such as the risk of price discrimination. Guests could be disadvantaged if prices in popular neighborhoods or for certain accommodation types rise disproportionately. Moreover, overrepresented categories might be favored, leading to the neglect of smaller or newer providers. Another risk is that if many hosts use similar models, algorithmic price recommendations could result in a "race to the top," causing overall price increases and reducing market attractiveness.

Specific risks arise during deployment. Technical risks, such as overfitting, could result in poor generalization of the model to new, unknown data, particularly in underrepresented categories. Data quality also remains a critical factor: missing values, outliers, and inconsistent data formats could impair prediction accuracy. Operationally, there is a risk of incorrect price recommendations, which could lead to revenue losses (due to prices being too low) or deter potential customers (due to prices being too high). Additionally, the model's relevance could diminish if it is not regularly updated with current data. Finally, reputational risks must be taken into account.

6.3 Monitoring

During the deployment of the rental price prediction model, various aspects must be monitored to ensure its performance and reliability while mitigating ethical and operational risks.

The model's performance should be tracked by regularly reviewing key metrics such as R^2 and MSE to ensure accurate predictions. Additionally, the model's ability to generalize to new, previously unseen data is a crucial indicator of its robustness and versatility. Intervention is necessary if there is a significant drop in R^2 (e.g., below 0.50) or a noticeable increase in MSE. Similarly, discrepancies between predicted prices and actual booking prices should be examined for potential model drift (see Chapter 6.1).

Data quality and input consistency are other critical factors. New data, such as listings or reviews, must be complete and correctly formatted to avoid biases. Changes in the distributions of key input variables such as `price`, `reviews_per_month`, and `availability_365` should be monitored, as significant shifts or an increase in outliers in `price` may indicate problems. A high volume of missing or invalid data also requires immediate intervention to maintain model accuracy.

In terms of bias and fairness, the model's predictions for different groups, such as `room_type` or `neighbourhood`, should be regularly evaluated to detect potential biases early. This includes assessing whether protected attributes like accommodation types are not systematically disadvantaged. Large differences in R^2 or MSE across groups or indications of systematic over- or underpricing in certain neighborhoods necessitate model adjustments (see Chapter 6.5).

Monitoring the ethical and social impacts is also essential. Price trends should be analyzed to avoid unintended consequences such as price inflation or gentrification in sensitive neighborhoods. Additionally, collecting feedback from hosts and guests is important to evaluate perceived fairness and satisfaction. Significant complaints about unfair or unrealistic price recommendations or indications of disproportionate price increases in low-income neighborhoods should be promptly investigated.

Operational stability of the model must be ensured by monitoring the availability and stability of prediction generation. Regular updates to the model and its data foundation are crucial to prevent outdated predictions. Moreover, it should be ensured that the model integrates seamlessly into existing Airbnb systems without causing bottlenecks or outages. Delays or failures during peak times and missing or outdated data updates are clear indicators of the need for action.

Finally, potential reputational risks must also be considered. This includes regularly reviewing public perception and reporting on pricing. Additionally, compliance with data protection laws such as GDPR and anti-discrimination regulations must be ensured.

6.4 Reproducibility

The reproducibility of the analysis is supported by several documented aspects. The dataset used, Airbnb 2023, has been clearly defined, including its structure, variables, and data types (Chapters 2.1 and 2.2). This provides a foundation for future analyses based on the same data. Variables such as `latitude`, `longitude`, `price`, and `reviews_per_month` have been identified, significantly facilitating the traceability of model inputs.

The data preparation processes are also thoroughly described. Steps such as removing outliers, cleaning duplicate IDs, replacing missing values, and transforming variables (e.g., converting neighborhood names into ZIP codes) have been documented in detail (Chapter 4). This information allows for identical dataset preparation, ensuring consistent results.

The model selection and hyperparameters have also been clearly justified. The Random Forest Regressor was chosen as the primary model, and the tested hyperparameter grid is comprehensively described (Chapter 6.2). This facilitates the replication of the model training process. Additionally, the use of 10-fold cross-validation for evaluating model performance is outlined, ensuring a consistent and robust evaluation of the results.

7 Conclusion

Initial data exploration of the Airbnb Open Data 2023 dataset reveals significant variations in pricing, availability and apparent demand. After subsequent data quality checks and removal of attributes with limited use, implementing thresholds to combat outliers, model performance improved significantly. The chosen RandomForestRegressor using predictors chosen through ANOVA, which included geospatial data, reviews, and room characteristics managed to achieve a R^2 score of 0.64. While this accuracy does not meet the original success criteria, it does show that a notable portion of the variance in the rental prices can be explained. The introduction of thresholds for outlier reduction is essential in improving the models accuracy, even if the outliers are legitimate data. Additionally the encoding of room_types, cities and neighborhood to ZIP codes shows that the applied pre-processing steps had a significant impact on the models result. In summary, key lessons learned are, that systematic data preparation, including outlier handling and the encoding of variables is crucial to achieve a viable predictive performance. Using the previously mentioned steps, this report is able to show that with the given public data, it is possible to develop a reliable foundation for data-driven decision-making in price prediction.

References

[1] Ang Zhu, Rong Li, and Zehao Xie. 2020. Machine learning prediction of new york airbnb prices. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*. IEEE, 1–5.

A Appendix

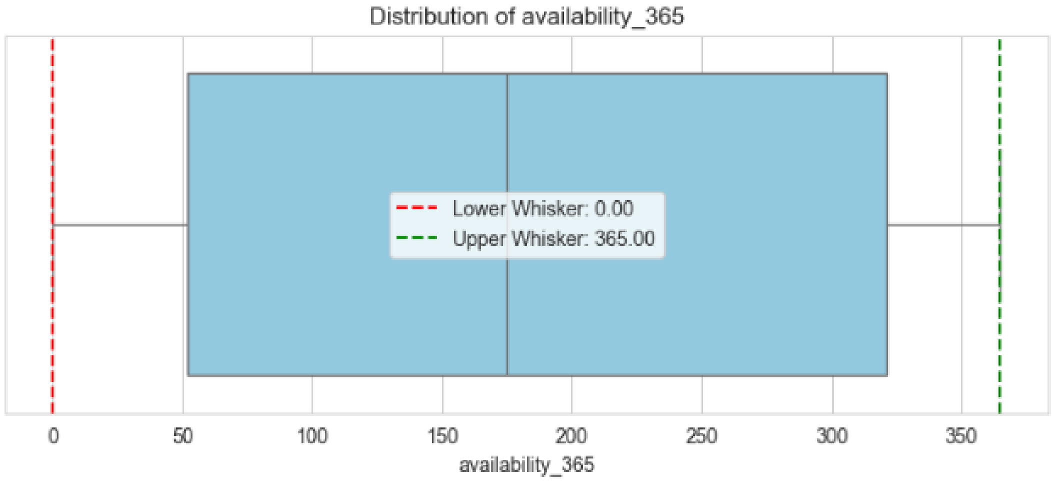


Figure 3: Boxplot: Distribution of availability_365

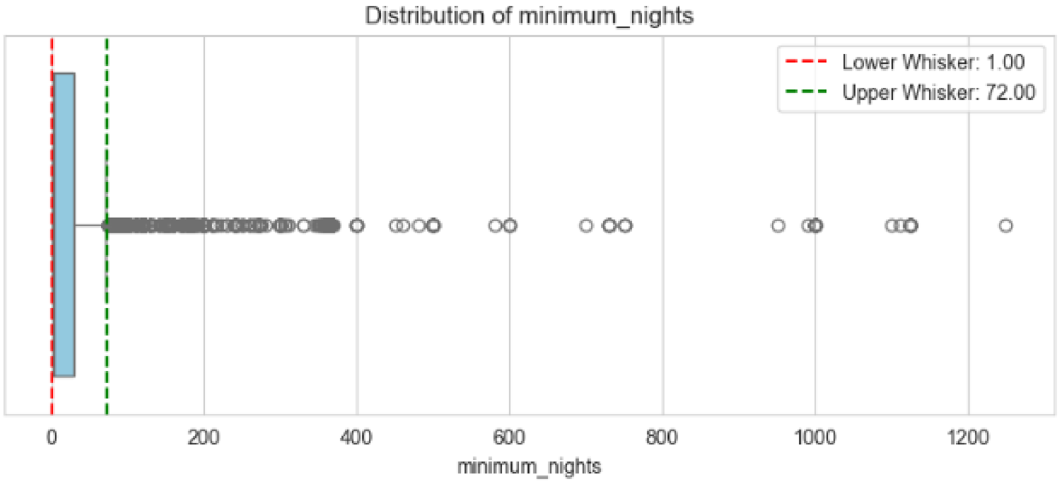


Figure 4: Boxplot: Distribution of minimum_nights

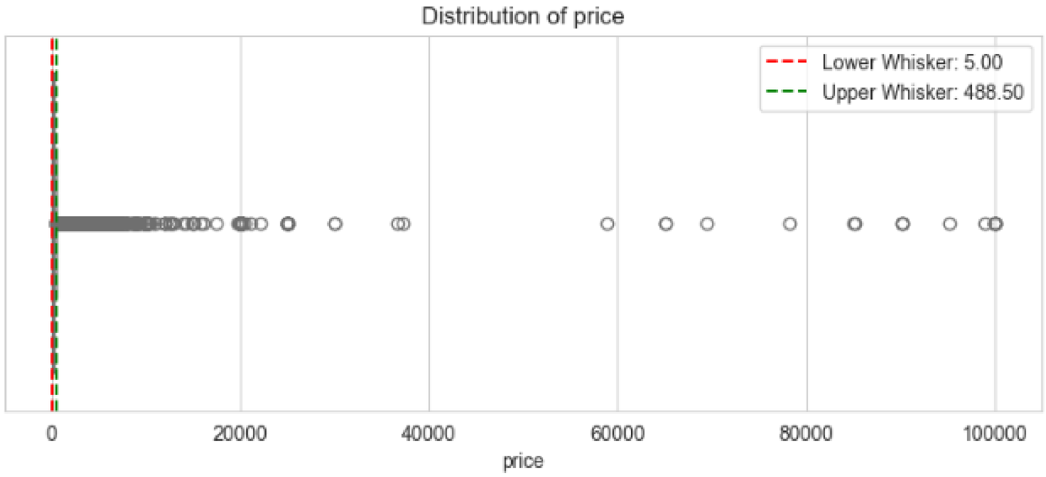


Figure 5: Boxplot: Distribution of price

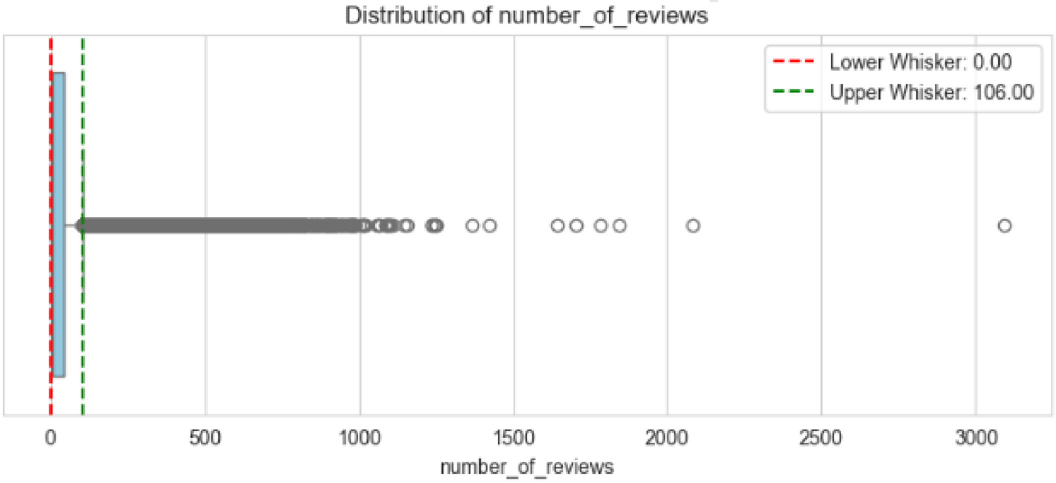


Figure 6: Boxplot: Distribution of number_of_reviews

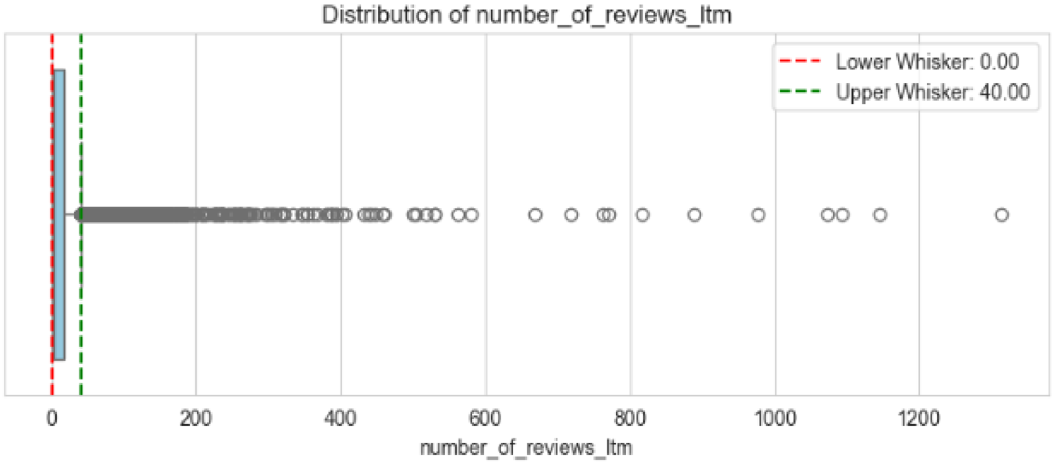


Figure 7: Boxplot: Distribution of number_of_reviews_ltm

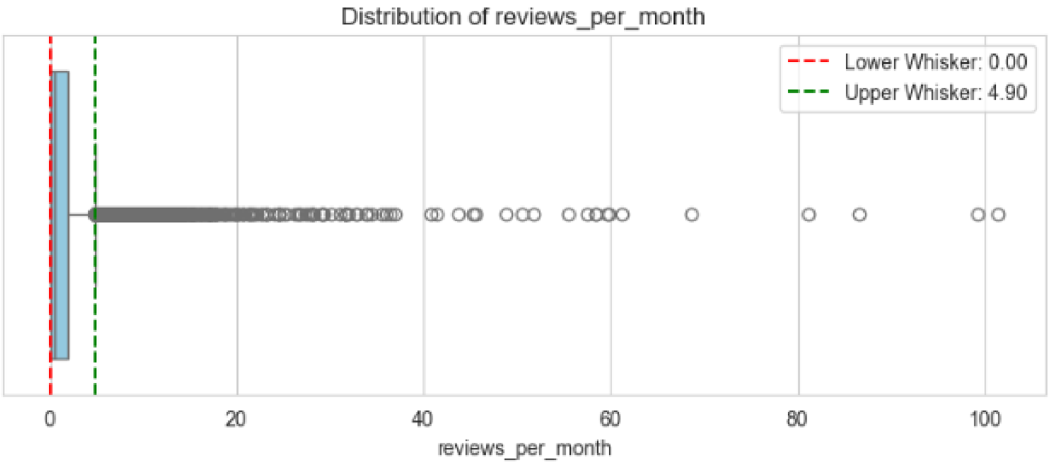


Figure 8: Boxplot: Distribution of reviews_per_month

B Online Resources

ZHU, Ang; LI, Rong; XIE, Zehao. Machine learning prediction of new york airbnb prices. In: 2020 Third International Conference on Artificial Intelligence for Industries (AI4I). IEEE, 2020. S. 1-5.