Inside Airbnb Analysis - Group 080

Philipp Moeßner
Student A, Matr.Nr.: 12412779

Vasili Savin
Student B, Matr.Nr.: 12449668

1 Business Understanding

1.1 Data Source and Scenario

We work for a private investor who wants to invest into apartments in Vienna to rent them on Airbnb. The goal is to understand which types of apartments are likely to perform well and how different property characteristics influence quality, popularity and price. We have access to the current listings of apartments in Vienna on Airbnb which serves as our data base for the analysis. Data source: https://insideairbnb.com/get-the-data/

1.2 Business Objectives

a) Identify which characteristics of Airbnb listings in Vienna are associated with higher rating scores. b) Understand which property characteristics contribute to higher popularity, measured as monthly review frequency. c) Identify which factors enable higher nightly prices, to support pricing and investment strategies. d) Identify high-opportunity neighborhoods: detect districts with high demand but relatively low supply.

1.3 Business Success Criteria

a) Provide a ranked list of factors (identify the top 3 modifiable attributes) that explain high rating scores. b) Provide a ranked list of factors (identify the top 3 modifiable attributes) that contribute most to a high monthly review frequency. c) Provide pricing guidelines based on modifiable attributes. d) Provide a ranked list (identify the top 3) of high-opportunity districts.

1.4 Data Mining Goals

a) Build a well-performing regression model for predicting rating score. b) Build a regression model that predicts monthly review frequency based on listing characteristics and quantifies the influence of individual features. c) Develop a regression model to predict the nightly price of listings based on property attributes, host characteristics, and location. d) Identify / construct a reasonable measure indicating „demand“ of a district and compare it against the corresponding amount of listings.

1.5 Data Mining Success Criteria

a) high R-squared (>0.4), low RMSE (<0.25) measure on the test set for the model. b) high R-squared (>0.5), low RMSE (<0.3) measure on the test set for the model. c) high R-squared (>0.6), low RMSE (<20€) measure on the test set for the model.
Reason for increasingly harder metric thresholds: rating score is probably the most noisy / subjective category followed by rating frequency while price is probably the variable that is more closely tied to the „hard facts“ / features of a listing. d) Classify all districts into one of the 3 categories: high, mid, low opportunity

1.6 AI Risk Aspects

Based on the current version of the EU AI Act, the proposed models would fall into the category of low-risk AI systems. The models do not operate in any regulated domains such as employment, access to essential services, migration or law enforcement and they do not involve biometric identification, social scoring of individuals etc. Therefore, no specific compliance requirements apply beyond general best practices regarding transparency, documentation, reproducibility and bias awareness.

2 Data Understanding

2.1 Data loading and variable subsetting

What we did. Load the raw Airbnb 'listings.csv' data for Vienna, join it with the variable dictionary, select a subset of relevant variables, and split them into numeric and non-numeric groups based on the data dictionary.

Load the raw Airbnb 'listings.csv' data for Vienna, join it with the variable dictionary, select a subset of relevant variables, and split them into numeric and non-numeric groups based on the data dictionary. We select a subset of 38 out of 79 available variables based on business relevance and feasibility. Primarily, we exclude free-text fields and short-term aggregates of variables where different aggreagation intervals are available.

2.2 Summary statistics

What we did. Computation of summary statistics for the selected Airbnb Vienna variables (data understanding phase). For numeric variables, mean, median, variance, min, max and skewness were computed. For non-numeric variables, the mode was reported. The resulting report is used to assess distributions, skewness, plausibility of values and potential preprocessing steps.

What we found. The summary statistics reveal highly skewed distributions for maximum nights to stay, bedrooms, price, revenue estimates and host listing counts. Rating variables show low variance and left-skewness, which suggests a positivity bias in guest feedback. These findings could motivate to use log-transformations and careful outlier treatment in the following data preparation steps.
Appendix references. Summary statistics table: Table 11

2.3 Variable correlations

What we did. Compute Pearson correlation coefficients on a numerical subset of the Airbnb Vienna dataset. Non-numeric attributes (strings, categories, dates) are excluded based on the data dictionary units. Boolean variables are mapped from 't'/'f' to 1/0 and all values are coerced to numeric. The correlation matrix is used for exploratory analysis.

What we found. Interpretation of correlation analysis:
reviews_per_month shows strong positive correlations with other

variables related to review-count (number_of_reviews_l30d, number_of_reviews_ly, number_of_reviews), which is expected because they capture related constructs. For feature selection, these variables should be treated carefully to avoid target leakage. Additional positive correlations with host and booking convenience indicators (instant_bookable, host_is_superhost, host_response_rate, host_identity_verified) suggest that professional host behavior and reduced booking friction may be associated with higher review frequency.
price correlates weak-to-moderately with size/capacity related variables (bedrooms, beds, accommodates, bathrooms), which is expected. The association with host_total_listings_count may indicate systematic pricing differences between professional and private hosts.
review_scores_rating is strongly correlated with the sub-scores (value, accuracy, cleanliness, communication, checkin, location), which is expected because the overall rating is derived from these aspects. Using all sub-scores as predictors for the overall score could reduce business insight and introduce multicollinearity; alternatively one could model overall rating from all other attributes.
Appendix references. Top correlations for reviews_per_month: Table 12 Top correlations for price: Table 13 Top correlations for review_scores_rating: Table 14

2.4 Missing values per variable

What we did. Analyze missing values for all selected variables by computing absolute and relative missing counts. Additionally, missingness indicator variables are created and correlated with numerical attributes to assess whether missing values occur randomly or follow a structural pattern.

What we found. The missing value analysis reveals a high proportion of missing values for several variables (e.g. price, estimated_revenue_l365d, beds, bathrooms, host_response_rate), affecting approximately 27% of the listings. Strong correlations between missingness indicators (near 1.0) indicate that these variables are jointly missing for the same subset of listings, suggesting structural rather than random missingness. Because price is a core response variable for subsequent models and no reliable imputation strategy exists for such a large fraction of missing values, the recommended strategy is to remove listings with missing price values. This removal simultaneously resolves most missingness in other variables while preserving a sufficiently large dataset for model training.
Appendix references. Missing values table: Table 15

2.5 Outliers per variable

What we did. Detect potential outliers for numeric non-boolean variables using the IQR rule (boxplot approach). For each variable, the interquartile range IQR = Q3 - Q1 is computed. Upper bound: Q3 + 1.5 * IQR; Lower bound: max(0, Q1 - 1.5 * IQR) (clipped to 0 to avoid negative lower bounds for inherently non-negative variables). A summary table reports the number and fraction of detected outliers as well as the computed bounds per variable. This step is exploratory and used to identify variables requiring plausibility checks; detected outliers are not automatically removed in Data Understanding.

What we found. Plausibility checks were conducted for variables with high outlier fractions or extreme upper quantiles. Because many distributions are highly concentrated (e.g., bathrooms around 1, minimum_nights around small integers), the IQR rule flags many values as outliers that are still plausible. Therefore, outlier detection results were validated via visual inspection (histograms) and by inspecting a small subset of extreme listings (>=99.9th percentile).
Findings / recommendations:
minimum_nights: extreme values often correspond to long-term stay offerings; considered plausible (no removal recommended solely due to IQR outlier flag).
maximum_nights: values like 1125 (approx. 3 years) appear frequently as placeholders for long-term rental; a single extreme value 99999 likely represents an invalid placeholder -> recommend removing this one data point.
number_of_reviews_ly: extreme values can be plausible for high-turnover listings (e.g. central locations) or atypical listing types; no blanket removal recommended.
price: extreme values above the 99.9th percentile appear implausible for nightly prices (e.g., >8000€ in low-cost districts); likely represent non-nightly pricing or errors -> recommend removing these extreme outliers.
bedrooms: extreme values often correspond to hotels or large apartments; considered plausible.
host_total_listings_count: very high values can be plausible for corporate hosts; considered plausible.
Appendix references. Outlier summary table: Table 16

2.6 Visual inspection of distributions

Visual exploration: plot distributions with reduced influence of heavy outliers. For numeric variables, the plotted range is restricted using quantile-based trimming (typically 1% or 99%), to improve readability and reveal structure in the central mass of the distribution. This view complements the raw distribution plots and supports plausibility assessment of outlier detection.
Appendix references. Outlier-robust histograms (split): Figures 2–4

2.7 Bias and risk reflection

Variable selection bias: Only a subset of 38 out of 79 available variables was selected based on business relevance and feasibility, excluding free-text fields and short-term aggregates (where different aggreagation intervals were available for a variable), which may omit relevant information contained in the deleted variables.
Rating bias: Rating scores are subjective and scales are typically interpreted differntly by different users. Often users tend to avoid low ratings, resulting in left-skewed distributions and reduced variance (as we can see in the summary statistics).
Popularity proxy bias: We want to use monthly review frequency as a proxy for popularity, but it reflects recent review activity rather than true or long-term demand, potentially underestimating popular listings with long review histories (where users might feel less of a need to still post their review).
Platform and sampling bias: The dataset only represents Airbnb listings and does not capture the full short-term rental or housing market, which limits generalizability beyond the platform.

Temporal snapshot bias: The dataset represents a single scrape / time point (14th September 2025); changes over time (trends, seasonality, and platform or market dynamics) are not observed.

3 Data Preparation

In the data preparation phase we turn the Airbnb subset into a model-ready dataset for further modelling. We implement outlier handling decisions derived in the Data Understanding phase, standardise data types and mark relevant variables as categorical. The result is a consistent feature table that can be reused across different models.

3.1 Outlier handling

Implement outlier handling decisions derived from the Data Understanding phase. Listings with maximum_nights=99999 are removed, and extreme price outliers above the 99.9th percentile are dropped as implausible nightly prices. This step produces a dataset where obviously error producing outlier values are removed while realistic extremes are kept.

3.2 Data conversion

Convert selected variables to readable formats suitable for modelling. Boolean and percentage variables are mapped to numeric values, date strings are parsed into proper datetime objects, and some fields are coerced to numeric types. This prevents subtle type-related errors in later analyses and model fitting.

3.3 Categorical conversion

Mark selected variables as categorical to prepare them for encoding in the modelling phase. The variables are cast to categorical dtypes instead of being treated as free-text or numeric fields. This makes the structure of the feature space explicit, supports reproducible one-hot encoding and simplifies feature selection in later steps.

4 Modeling

For the regression tasks, we use regularized linear models because we need interpretability (top 3 actionable factors as business success criterion for the investor) and robustness under multicollinearity and missingness. We first try ElasticNet regression for both predicting ratings (review_scores_rating) and reviews per month (our proxy for indicating popularity). Later, we also apply LogisticRegression with elastic-net penalty for rating prediction (separating the data into top-tier ratings >= 4.8 and low-tier ratings < 4.8). Finally, we use HistGradientBoostingRegressor as the main method for a set of the tasks, as it showed the most promising results.

4.1 Predicting ratings

4.1.1 ElasticNet Regression. First, we try to predict review_scores_rating through ElasticNet regression. After splitting the dataset by 80/20 into train and test set, we tune ElasticNet regression using GridSearchCV with 5-fold CV on the training set. Primary metric is R2. For the alpha parameter (strength of the regularization) we specify the grid range as [0.01, 0.1, 1.0, 10.0], for the l1_ratio parameter (weighting between l1 and l2 penalty) we use [0.1, 0.5, 0.9]. This lets us cover a large range of parameters without letting the grid explode in size. The best combination of parameters was identified with alpha at 0.01 and penalty ratio at 0.5. Still, the mean R2 score on all of the CV folds with these hyperparameters is 0.15 meaning only 15% of the variance could be explained by the model (see table 1). This rather bad performance motivated changing the modeling objective from a linear regression task to a logistic regression / binary classification task (classifying the data into top-tier and low-tier rated listings).

Table 1: Top 5 GridSearchCV configurations for ElasticNet rating regression (5-fold CV, metric: R^2).

rank_test_score	mean_test_score	std_test_score	params
1	0.149488	0.010945	model__alpha = 0.01 model__l1_ratio = 0.5
2	0.148409	0.014362	model__alpha = 0.01 model__l1_ratio = 0.1
3	0.144986	0.009040	model__alpha = 0.01 model__l1_ratio = 0.9
4	0.141733	0.007980	model__alpha = 0.1 model__l1_ratio = 0.1
5	0.085766	0.005979	model__alpha = 0.1 model__l1_ratio = 0.5

4.1.2 LogisticRegression. Since our classic linear regression model performed so bad on the rating prediction task, we instead decided to try to classify the data into top-tier (>= 4.8) and low-tier ratings (< 4.8) by a LogisticRegression (elastic-net penalty). We therefore transform the continuous numerical to a binary target. Since both classes which emerged from that transformation are already almost equal in size, we don't have to consider stratification. Again, we use GridSearchCV with 5-fold CV on the training set and the primary metric as balanced accuracy. The best combination of hyperparameters was identified with alpha at 0.01 and penalty ratio at 0.5 (see table 2) achieving a mean balanced accuracy for the CV folds of 71.4%. GridSearchCV was run with refit=True, hence after selecting the best hyperparameters via 5-fold CV on the training set, scikit-learn automatically refits the best configuration on the full training set. Thus, we did not need an additional validation dataset and a refitting on the training data.
The variables with the largest positive coefficients (factors where an increase of the variable led to an increase of the probability for having top-tier ratings) were host_is_superhost, estimated_revenue_l365d, host_acceptance_rate__missing, minimum_nights and bathrooms (see table 3).

Table 2: Top 5 GridSearchCV configurations for Logistic Regression (elastic-net penalty, 5-fold CV, metric: balanced accuracy).

rank_test_score	mean_test_score	std_test_score	params
1	0.713588	0.009099	model__C = 0.1 model__l1_ratio = 0.5
2	0.713587	0.009521	model__C = 0.1 model__l1_ratio = 0.9
3	0.713308	0.007853	model__C = 0.1 model__l1_ratio = 0.1
4	0.713167	0.006586	model__C = 1.0 model__l1_ratio = 0.9
5	0.712886	0.006065	model__C = 1.0 model__l1_ratio = 0.5

Table 3: Top 15 Logistic Regression coefficients (elastic-net) for predicting top-tier ratings (≥ 4.8). Coefficients are in log-odds; odds ratios are $\exp(\beta)$.

feature	coef_log_odds	odds_ratio	abs_coef
host_is_superhost	0.912646	2.490904	0.912646
estimated_revenue_l365d	0.487274	1.627873	0.487274
estimated_occupancy_l365d	-0.322908	0.724041	0.322908
instant_bookable	-0.275475	0.759211	0.275475
has_availability__missing	-0.195560	0.822374	0.195560
host_acceptance_rate__missing	0.167430	1.182263	0.167430
availability_30	-0.157859	0.853970	0.157859
host_acceptance_rate	-0.154929	0.856476	0.154929
minimum_nights	0.142823	1.153526	0.142823
bathrooms	0.130371	1.139251	0.130371
reviews_per_month	-0.121647	0.885461	0.121647
number_of_reviews_l30d	0.112788	1.119395	0.112788
neighbourhood_cleansed_Meidling	-0.109035	0.896699	0.109035
host_response_time_within an hour	-0.107309	0.898248	0.107309
host_is_superhost__missing	0.107192	1.113148	0.107192

4.2 Predicting popularity

We tune ElasticNet regression for reviews_per_month (our proxy for popularity), after splitting the dataset by 80/20 into train and test set. Primary metric is R2. For the alpha parameter (strength of the regularization) we specify the grid range as [0.01, 0.1, 1.0, 10.0], for the l1_ratio parameter (weighting between l1 and l2 penalty) we use [0.1, 0.5, 0.9]. This lets us cover a large range of parameters without letting the grid explode in size. The best combination of hyperparameters was identified with alpha at 0.01 and penalty ratio at 0.1 achieving an average R2 score on the CV folds of 64.8% (see table 4). GridSearchCV was run with refit=True, hence after selecting the best hyperparameters via 5-fold CV on the training set, scikit-learn automatically refits the best configuration on the full training set. Thus, we did not need an additional validation dataset and a refitting on the training data.

The variables with the largest positive coefficients (factors where an increase of the variable led to an increase of predicted popularity) were estimated_revenue_l365d, review_scores_value, instant_bookable, host_acceptance_rate__missing, review_scores_communication (see table 5).

Table 4: Top 5 GridSearchCV configurations for ElasticNet regression predicting reviews_per_month (5-fold CV, metric: R^2).

rank_test_score	mean_test_score	std_test_score	params
1	0.647510	0.013921	model__alpha = 0.01 model__l1_ratio = 0.1
2	0.645528	0.013822	model__alpha = 0.01 model__l1_ratio = 0.5
3	0.639954	0.013778	model__alpha = 0.01 model__l1_ratio = 0.9
4	0.633364	0.011033	model__alpha = 0.1 model__l1_ratio = 0.1
5	0.603342	0.008884	model__alpha = 0.1 model__l1_ratio = 0.5

Table 5: Top 15 ElasticNet regression coefficients for predicting reviews_per_month. Coefficients are standardized due to feature scaling.

feature	coefficient	abs_coefficient
estimated_occupancy_l365d	0.348113	0.348113
minimum_nights	-0.111780	0.111780
review_scores_value	0.092190	0.092190
last_review_days_since	-0.065410	0.065410
review_scores_rating	-0.054873	0.054873
instant_bookable	0.041479	0.041479
host_acceptance_rate__missing	0.037856	0.037856
review_scores_location	-0.033722	0.033722
review_scores_communication	0.030415	0.030415
host_response_time_within an hour	0.025552	0.025552
calculated_host_listings_count	0.023906	0.023906
review_scores_cleanliness__missing	-0.023608	0.023608
review_scores_location__missing	-0.023252	0.023252
review_scores_rating__missing	-0.022758	0.022758
review_scores_communication__missing	0.021895	0.021895

4.3 Predicting price

We try three regression models for nightly price prediction using GridSearchCV with 5-fold cross-validation on the training set. GridSearchCV selects the parameter configuration that minimizes RMSE. We restrict the hyperparameter grids to a small but meaningful set of values to cover a range of regularization strengths and model complexities. For ElasticNet, we tune the regularization strength alpha = [0.01, 0.1, 1.0], L1 and L2 penalties l1_ratio = [0.1, 0.5, 0.9]. The best configuration was alpha=0.01 and l1_ratio=0.1, achieving a mean CV score of 0.4943 (std 0.0324), corresponding to a CV RMSE of approximately 0.494 in the optimized (log) space. For HistGradientBoostingRegressor, we tune the learning rate = [0.03, 0.05, 0.1], maximum tree depth = [4,6,8], minimum samples per leaf = [10,20,50], and L2 regularization = [104, 103, 102]. The best configuration was learning_rate = 0.1, max_depth = 8, min_samples_leaf = 10, and l2_regularization = 0.001, achieving the best mean CV score of 0.2759 (std 0.0118), CV RMSE of approximately 0.276. For RandomForestRegressor, we tune the number of trees = [200, 400] and the minimum samples per leaf = [10, 20, 50]. The best configuration was n_estimators = 400 and min_samples_leaf = 10, with a mean CV score of 0.3936 (std 0.0246), corresponding to a CV RMSE of approximately 0.394. While Random Forest outperforms ElasticNet, it remains notably worse than HistGradientBoosting in our setting (Table 6, 7, and 8).

Table 6: GridSearchCV results for RandomForest price prediction.

rank_test_score	mean_test_score	std_test_score	params
1	-0.393626	0.024604	{model__min_samples_leaf: 10, model__n_estimators: 400}
2	-0.394213	0.024761	{model__min_samples_leaf: 10, model__n_estimators: 200}
3	-0.424080	0.020492	{model__min_samples_leaf: 20, model__n_estimators: 400}
4	-0.424620	0.020062	{model__min_samples_leaf: 20, model__n_estimators: 200}
5	-0.460339	0.019309	{model__min_samples_leaf: 50, model__n_estimators: 400}
6	-0.460482	0.018831	{model__min_samples_leaf: 50, model__n_estimators: 200}

Table 7: Top GridSearchCV configurations for HistGradientBoosting price prediction.

rank_test_score	mean_test_score	std_test_score	params
1	-0.275887	0.011800	{model__l2_regularization: 0.001, model__learning_rate: 0.1, model__max_depth: 8, model__min_samples_leaf: 10}
2	-0.275962	0.011895	{model__l2_regularization: 0.0001, model__learning_rate: 0.1, model__max_depth: 8, model__min_samples_leaf: 10}
3	-0.277791	0.012418	{model__l2_regularization: 0.01, model__learning_rate: 0.1, model__max_depth: 8, model__min_samples_leaf: 10}
4	-0.285510	0.012797	{model__l2_regularization: 0.0001, model__learning_rate: 0.1, model__max_depth: 8, model__min_samples_leaf: 20}
5	-0.285590	0.012783	{model__l2_regularization: 0.001, model__learning_rate: 0.1, model__max_depth: 8, model__min_samples_leaf: 20}
6	-0.285709	0.011765	{model__l2_regularization: 0.01, model__learning_rate: 0.1, model__max_depth: 8, model__min_samples_leaf: 20}
7	-0.288880	0.015117	{model__l2_regularization: 0.0001, model__learning_rate: 0.1, model__max_depth: 6, model__min_samples_leaf: 10}
8	-0.289079	0.015062	{model__l2_regularization: 0.01, model__learning_rate: 0.1, model__max_depth: 6, model__min_samples_leaf: 10}
9	-0.289450	0.015450	{model__l2_regularization: 0.001, model__learning_rate: 0.1, model__max_depth: 6, model__min_samples_leaf: 10}
10	-0.295396	0.012183	{model__l2_regularization: 0.001, model__learning_rate: 0.1, model__max_depth: 6, model__min_samples_leaf: 20}

Table 8: GridSearchCV results for ElasticNet price prediction.

rank_test_score	mean_test_score	std_test_score	params
1	-0.494347	0.032428	{model__alpha: 0.01, model__l1_ratio: 0.1}
2	-0.508404	0.030996	{model__alpha: 0.01, model__l1_ratio: 0.5}
3	-0.515935	0.030515	{model__alpha: 0.01, model__l1_ratio: 0.9}
4	-0.529617	0.029086	{model__alpha: 0.1, model__l1_ratio: 0.1}
5	-0.562787	0.027307	{model__alpha: 0.1, model__l1_ratio: 0.5}
6	-0.565697	0.027143	{model__alpha: 0.1, model__l1_ratio: 0.9}
7	-0.569596	0.027006	{model__alpha: 1.0, model__l1_ratio: 0.1}
8	-0.598117	0.024385	{model__alpha: 1.0, model__l1_ratio: 0.5}
9	-0.599241	0.023331	{model__alpha: 1.0, model__l1_ratio: 0.9}

4.4 Identifying high opportunity neighborhoods

To identify high-opportunity neighbourhoods, we analyze the imbalance between demand and supply at the district level. Supply is defined as the number of active Airbnb listings per district, while demand is proxied in two ways: (1) an observed demand proxy based on the total number of reviews per month, and (2) a modeled demand estimate obtained from the trained popularity prediction model.

For the observed approach, district-level demand is computed as the sum of reviews per month across all listings within each district. This demand measure is normalized by supply to obtain an *observed opportunity score*, defined as the average number of monthly reviews per listing. Districts with high values of this score are interpreted as having strong demand relative to existing supply.

Both observed and modeled scores are used to rank Vienna's districts and identify neighbourhoods that exhibit consistently high demand relative to supply.

Based on the results of the previous modeling tasks, we used Random Forest, HistGradientBoosting, and ElasticNet to model reviews_per_month. The logic for the most efficient hyperparameters is exactly the same as the one for the previous investigations. HistGradientBoosing turned out to be the most accurate option for us.

5 Evaluation

5.1 Rating prediction

5.1.1 Test set performance. On the test set, the model achieved a balanced accuracy score of 70.2%, almost matching the performance for the training data. Hence, for both classes on average 70% of the listings were correctly classified (see also confusion matrix in figure 1). This performance supports the validity of taking the largest identified variable coefficients as indicators for high listing rating scores.

5.1.2 SOTA and baseline performance. We could not identify any state-of-the-art performance for this kind of prediction task on the Inside Airbnb dataset. Still, we calculated performance measures for trivial baseline models. As expected, a majority model (always predicting the majority class of the training set) achieves a balanced accuracy of 50% on the test set (which contains approximately an equal number of both classes) similarly to the a random model (randomly classifying the test instances into one of the classes) with 48.6% bacc. Hence, our trained model achieves significant improvement compared to these baselines.

5.1.3 Group biases. Exploratively using the "instant_bookable" variable as a subgroup attribute, the model shows consistently good performance across both subgroups. Balanced accuracy is slightly higher for instant-bookable listings (0.68) compared to non-instant-bookable listings (0.65), but the difference is moderate and both groups are well above random performance. This indicates no substantial subgroup bias with respect to booking mode, but rather a slightly easier prediction task for instant-bookable listings.

5.1.4 Conclusion. Our initial regression model heavily underperformed the data mining success criteria. In order to still get a valid idea of which variables contribute to a higher rating predictions in the dataset, we transformed the task to binary classification into rating-tiers. This model performed quite well. Though, this performance is probably due to leakage of rating information through the variable "host_is_superhost" since a superhost label is (probably) obtained on the Airbnb platform by having high ratings.

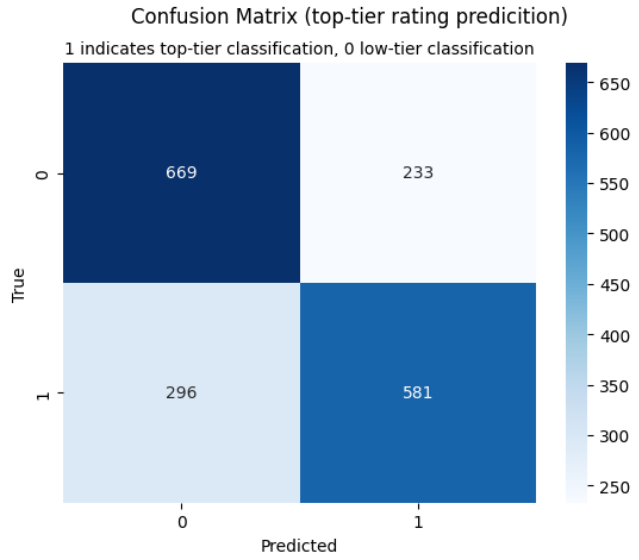

Figure 1: Confusion matrix of the logistic regression model for predicting top-tier ratings (≥ 4.8) on the test set. Rows correspond to true labels, columns to predicted labels.

5.2 Popularity prediction

5.2.1 Test set performance. On the test set, the model achieved an R2 score of 67.5% and a RMSE of 0.35 which is even better than on the training data. This performance supports the validity of taking the largest identified variable coefficients as indicators for a high amount of monthly reviews.

5.2.2 SOTA and baseline performance. We could not identify any state-of-the-art performance for this kind of predicition task on the Inside Airbnb dataset. Still, we calculated performance measures for trivial baseline models. We predict the mean target value estimated from the training set for all test instances. As expected, this approach yields an R2 close to zero, confirming that it provides no explanatory power beyond the global average. Hence, our trained model achieves significant improvement compared to this baseline.

5.2.3 Group biases. Exploratively using the "instant_bookable" variable as a subgroup attribute, the model shows highly consistent performance across both groups (instantly bookable and not instantly bookable listings). RMSE and MAE are nearly identical, and the mean prediction error is close to zero for both subgroups. This indicates no systematic bias with respect to booking mode and suggests that the model generalizes robustly across this subgroup.

5.2.4 Conclusion. The regression model for predicting monthly reviews of listings almost achieved the data mining success criteria. With an R2 score on the test set of 67.5% it crossed the success threshold of > 50%. On the other hand, the model did not achieve the goal of having an RMSE below 0.3.

5.3 Price prediction

The goal is to Identify which factors enable higher nightly prices, to support pricing and investment strategies. The target variable is the log-transformed nightly price, chosen to address the strong right skew and heavy-tailed distribution of raw prices. We evaluate models of increasing complexity: ElasticNet, Random Forest Regressor, and HistGradientBoostingRegressor, which is selected as the final model due to its strong performance and computational efficiency. Hyperparameters are selected using cross-validation, optimizing negative RMSE in log space. For HistGradientBoosting, tuning focuses on learning rate, tree depth, minimum samples per leaf, and L2 regularization to balance bias and variance. On the test set, HistGradientBoosting achieves a log-RMSE of 0.085, log-MAE of 0.153, and an R2 of 0.804. In Euro space, this corresponds to an RMSE of approximately €130 and an MAE of approximately €47, with a MAPE of 15.8% and a median absolute percentage error of 8.4%. Performance varies strongly by price segment. Listings priced below €156 exhibit MAE values between €9 and €15, while high-priced listings (€156–€8000) show much larger errors, with an MAE of €196 and RMSE exceeding €800. This systematic underprediction of expensive listings highlights the role of unobserved factors. Given the results, we define the following feature importance (see table 6).

Table 9: Permutation based feature importance list for the price prediction model

feature	importance_mean	importance_std
estimated_revenue_l365d	0.890134	0.015389
estimated_occupancy_l365d	0.884898	0.010057
host_total_listings_count	0.174321	0.010952
host_response_rate	0.098684	0.006429
minimum_nights	0.030677	0.004393
bedrooms	0.029450	0.002272
accommodates	0.020814	0.002181
host_acceptance_rate	0.019487	0.003006
calculated_host_listings_count	0.018115	0.001624
availability_30	0.017144	0.002340
property_type_Private room in rental unit	0.016836	0.002507
neighbourhood_cleansed_Innere Stadt	0.015457	0.002220
reviews_per_month	0.013799	0.001092
latitude	0.013590	0.001614
longitude	0.013501	0.002123
number_of_reviews	0.006986	0.000924
instant_bookable	0.006634	0.001205
bathrooms	0.005707	0.001746
room_type_Entire home/apt	0.005579	0.001258
room_type_Shared room	0.005307	0.000796

5.3.1 Test set performance. The target variable for price prediction is the log-transformed nightly price, which is used to address the strong right skew and heavy-tailed distribution of raw prices. Among the evaluated models, HistGradientBoostingRegressor is selected as the final model due to its strong predictive performance and favorable computational efficiency.

On the test set, the final HistGradientBoosting model achieves a log-RMSE of 0.085, a log-MAE of 0.153, and an R2 of 0.804, indicating that a substantial portion of the variance in nightly prices can be explained by the available listing level features. When transforming predictions back into euro space, this corresponds to an

RMSE of approximately €130 and an MAE of approximately €47. Relative error metrics further show a MAPE of 15.8% and a median absolute percentage error (MdAPE) of 8.4%, suggesting that typical prediction errors are moderate for the majority of listings.

5.3.2 SOTA and baseline performance. We compare the models against simple baseline predictors as well as more expressive regression models. Two naive baselines are implemented that ignore all listing characteristics and predict a constant value in log-price space. Specifically, the baselines predict either the *mean* or the *median* of the training target $y_{\text{train,log}}$ for all evaluation samples.

The baseline results highlight the limitations of central-tendency predictors. The mean baseline achieves a log-RMSE of approximately 0.435 with an R^2 close to zero, while the median baseline performs slightly worse with a log-RMSE of approximately 0.443 and a negative R^2. In Euro space, both baselines yield very large absolute errors, with RMSE values exceeding 280,000 and MAE values above 90, as well as high relative errors (MAPE > 45%). These results confirm that predicting a global average price is insufficient for capturing the structure of Airbnb pricing.
Furthermore, our results clearly outperform external ("SOTA") studies (see Airbnb Explorer (bluewallumich), Cornell INFO2950 Price Prediction Analysis): while their Random Forest models achieve log-RMSE values of approximately 0.42–0.52 (corresponding to 50–70% multiplicative error) and R^2 values around 0.55, our final HistGradientBoosting model reaches a log-RMSE of 0.085 and an R^2 of 0.80. This implies that our predicted prices deviate by only about 9% from the true prices on average, substantially improving upon Random Forest, k-NN, and regularized linear models.

5.3.3 Group biases. Model performance is not uniform across the price distribution. When stratifying prediction errors by price segment, clear group-level biases emerge. Listings priced below 156 exhibit relatively small absolute errors, with mean absolute error (MAE) values ranging between approximately 9 and 15. This indicates strong predictive accuracy in the low- and mid-price segments, where data density is high and pricing behavior is more homogeneous.

In contrast, high-priced listings in the range of 156 to 8,000 show substantially larger errors. For this segment, the MAE increases to approximately 196, and the root mean squared error (RMSE) exceeds 800. The model systematically underpredicts these expensive listings, reflecting both the limited number of high-end observations in the dataset and the influence of unobserved factors.

5.3.4 Conclusion. Overall, the price prediction results demonstrate that HistGradientBoosting provides a strong and robust model for estimating nightly prices at scale, particularly for low- and mid-priced listings. The model substantially outperforms linear methods and naive baselines, validating the choice of non-linear gradient boosting and cross-validated hyperparameter tuning. However, the presence of systematic underprediction for high-priced listings indicates that the model captures only part of the pricing mechanism. The most significant factors that define the price of an apartment have been successfully identified.

5.4 Neighborhood classification

5.4.1 Test set performance. The HistGradientBoostingRegressor trained to predict `reviews_per_month` showed strong and stable performance in the popularity prediction task and was therefore applied to all listings to estimate expected demand.

When aggregating the predicted demand at the district level and normalizing by supply, the resulting opportunity scores exhibit consistent patterns across neighbourhoods. This stability indicates that the model generalizes well for the purpose of neighbourhood-level comparison.

5.4.2 SOTA and baseline performance. For identifying high opportunity neighbourhoods we compare the model-based approach against a transparent baseline using *observed demand*, defined as the total number of reviews per month per district normalized by the number of listings. Both approaches yield consistent rankings among the top districts. In particular, neighbourhoods such as Favoriten, Meidling, and Landstraße rank among the highest-opportunity areas under both observed and modeled demand, with opportunity scores in a similar range.

5.4.3 Group biases. Exploratory analysis across districts shows that variation in opportunity scores is primarily driven by *supply saturation* rather than systematic model bias. Central districts with very high listing density exhibit lower opportunity scores despite high absolute demand, whereas districts with moderate supply and strong demand consistently rank higher.

5.4.4 Conclusion. Overall, the updated analysis confirms that the model-based approach provides a robust and interpretable framework for identifying high-opportunity neighbourhoods. By combining observed demand with model-predicted demand, the method balances transparency and stability. The full ranked list can be seen in Table 10.

Table 10: Observed district-level opportunity scores.

district	supply_listings	demand_reviews_per_month	avg_reviews_per_listing	median_reviews_per_listing	opportunity_score_observed
Meidling	677	1106.69	1.859983	1.150	1.634697
Favoriten	1328	2149.31	1.877127	1.140	1.618456
Landstraße	1206	1797.46	1.776146	1.090	1.490431
Brigittenau	601	835.64	1.654733	1.000	1.390416
Rudolfsheim-Fünfhaus	1021	1394.73	1.631263	1.000	1.366043
Simmering	226	302.23	1.574115	1.000	1.337301
Mariahilf	558	740.86	1.549916	0.750	1.327706
Wieden	487	622.78	1.560852	0.750	1.278809
Neubau	679	857.11	1.470172	0.750	1.262312
Leopoldstadt	1540	1908.85	1.453808	0.700	1.239513
Hernals	450	556.51	1.476154	0.780	1.236689
Innere Stadt	578	714.28	1.526239	0.995	1.235779
Ottakring	791	976.84	1.477821	0.880	1.234943
Penzing	455	533.88	1.408654	0.590	1.173363
Margareten	759	848.13	1.354840	0.670	1.117431
Floridsdorf	257	280.64	1.368976	1.000	1.091984
Währing	396	398.15	1.221319	0.775	1.005429
Donaustadt	457	454.25	1.234375	0.680	0.993982
Alsergrund	697	686.70	1.150251	0.580	0.985222
Josefstadt	335	272.61	1.028717	0.420	0.813761

6 Deployment

6.1 Have we achieved the business objectives?

6.1.1 Objective a. We could identify a list of variables which contributed positively to a top-tier rating prediction. Still, none of the top 3 strongest variables / predictors give particular insight about which characteristics of a listing could be modified in order to achieve higher ratings. The strongest predictor "host_is_superhost" is most likely an effect of high ratings rather than a reason for them. The same holds for estimated_revenue_l365 (indicating the achieved revenue over the last 365 days with this listing). Also,

while it is interesting that host_acceptance_rate__missing (a missing host acceptance rate) helps to explain the data, we cannot derive any reasonable decision from it. The next two strongest and positive predictors were "minimum_nights" and "bathrooms", where a higher minimum requirement of nights to stay and more bathrooms increased the predicted probability of top-tier ratings.
In general, one has to be aware that predictors cannot give generalizable insight about cause and effect. Predictors themselves could be influenced by extraneous variables potentially being the true effect of the observed behavior.

6.1.2 Objective b. We could identify a list of variables which contributed positively to a top-tier rating prediction. Still, the strongest predictor estimated_revenue_l365 (indicates the achieved revenue over the last 365 days with this listing) seems to be rather an effect of high listing popularity, since more customers also lead to higher revenue. For review_scores_value (the second strongest positive predictor for popularity) one could hypothesize that higher ratings also encourage others to leave a rating, eventually increasing monthly review frequency (our proxy for popularity). This could then lead to the business recommendation of encouraging people to post high ratings in order to increase popularity of the listing. On the other hand, high average ratings could also be an effect popularity or both variables could be influenced by a lot of different factors. For the third strongest predictor "instant_bookable" one could hypothesize that a listing, which can be booked directly and thus has fewer barriers of rental, also attracts more people who might then post a rating. A recommendation could therefore be to enable the instant booking feature for a listing.
Still, one has to be aware that predictors cannot give generalizable insight about cause and effect. Predictors themselves could be influenced by extraneous variables potentially being the true effect of the observed behavior.

6.1.3 Objective c. We were able to build a strong predictive model for nightly price and to extract practically relevant pricing drivers. We modeled `price` in log-space to reduce the strong right-skew and heavy tails of the raw Euro distribution, and evaluated three regression models (ElasticNet, RandomForest, HistGradientBoostingRegressor). HistGradientBoosting achieved the best generalization performance and was selected as the final model. On the test set, it reached a log-RMSE of about 0.085, a log-MAE of about 0.153, and $R^2 \approx 0.804$, corresponding in Euro space to an RMSE of roughly 130 and an MAE of roughly 47, with MAPE $\approx$ 15.8% and MdAPE $\approx$ 8.4%. From the model analysis, the strongest predictors of price are `estimated_revenue_l365d` and `estimated_occupancy_l365d`. A key limitation for the model is that price errors increase substantially for the most expensive listings. For listings priced below 156, the model achieves relatively low absolute errors (MAE roughly 9–15), while the high-price segment (156–8000) shows much larger uncertainty (MAE around 196 and very large RMSE). This systematic underprediction of premium listings indicates the presence of unobserved factors not captured in our feature set. However, overall, the objective has been achieved.

6.1.4 Objective d. We identified high-opportunity districts by comparing demand and supply at the neighbourhood level. Supply is defined as the number of active listings per district, while demand is proxied by `reviews_per_month`. The observed opportunity score is computed as aggregated district-level demand normalized by supply, reflecting expected demand pressure per listing. Using this metric, the highest observed opportunity scores are found in Meidling, Favoriten and Landstraße. We additionally compute a model-based opportunity score, use the predicted demand from the popularity model and normalizing by supply. For deployment, districts can be grouped into high, mid, and low opportunity tiers based on their opportunity scores, providing a practical decision-support tool for identifying promising investment areas.

6.2 Monitoring requirements

Since we did not development the models to make further predictions but rather for analytical purposes of the underlying data generation process, no specific monitoring of the models is required.

7 Findings and lessons learned

7.1 Recommended business actions

- One could try to increase the required minimum nights to stay for a listing and provide listings / invest in apartments with tendentially more bathrooms to increase rating scores.
- One could try to enable the instantly bookable feature or encourage guests to post a good rating for increasing the popularity of a listing.
- One could try
- To increase achievable nightly prices, one could try to prioritize listings with higher accommodation capacity and bedroom.
- For the most promising areas, a business should focus on districts with the highest modeled opportunity scores such as Favoriten, Meidling, and Landstraße.

7.2 Reflections

- In this project the clear goal was to build models (for several variables) which should be explainable. This motivated the use of simpler models with potentially lower performance but therefore better interpretability and clearer business conclusions.
- Having to document all processes in the provenance graph seemed cumbersome at first but on the other hand motivated to reflect a lot and double check what we were actually doing.

A Appendix: Tables and Figures

Table 11: Summary statistics for selected variables.

variable	mean	mode	median	var	min	max	skew	unit
host_response_rate	0.936	1.000	1.000	0.030	0.000	1.000	-4.011	percent
host_acceptance_rate	0.893	1.000	0.990	0.051	0.000	1.000	-2.726	percent
host_total_listings_count	45.206	1.000	5.000	22341.473	1.000	8769.000	19.664	count
latitude	48.204	48.164	48.203	0.001	48.126	48.297	0.206	degree
longitude	16.361	16.316	16.360	0.001	16.198	16.543	0.280	degree
accommodates	3.477	2.000	3.000	3.692	1.000	16.000	1.939	count
bathrooms	1.185	1.000	1.000	0.237	0.000	12.000	5.914	count
bedrooms	1.342	1.000	1.000	0.925	0.000	50.000	12.657	count
beds	2.031	1.000	2.000	1.852	0.000	20.000	2.872	count
price	156.728	72.000	93.000	284583.583	13.000	10000.000	14.168	EUR
minimum_nights	7.718	1.000	2.000	794.853	1.000	1125.000	21.181	night
maximum_nights	456.687	365.000	365.000	866114.430	1.000	99999.000	86.724	night
availability_30	8.200	0.000	5.000	84.892	0.000	30.000	0.904	day
number_of_reviews	43.364	0.000	10.000	6692.908	0.000	1347.000	3.860	count
number_of_reviews_l30d	0.905	0.000	0.000	3.502	0.000	37.000	4.153	count
number_of_reviews_ly	9.002	0.000	0.000	308.239	0.000	299.000	3.289	count
estimated_occupancy_l365d	63.766	0.000	18.000	7439.825	0.000	255.000	1.241	percent
estimated_revenue_l365d	9062.259	0.000	3900.000	516319347.021	0.000	1073550.000	26.488	EUR
review_scores_rating	4.685	5.000	4.800	0.209	1.000	5.000	-4.032	rating-1-5
review_scores_accuracy	4.739	5.000	4.860	0.190	0.000	5.000	-4.636	rating-1-5
review_scores_cleanliness	4.656	5.000	4.800	0.246	0.000	5.000	-3.720	rating-1-5
review_scores_checkin	4.794	5.000	4.910	0.173	0.000	5.000	-5.411	rating-1-5
review_scores_communication	4.778	5.000	4.910	0.193	0.000	5.000	-5.139	rating-1-5
review_scores_location	4.679	5.000	4.770	0.163	0.000	5.000	-4.012	rating-1-5
review_scores_value	4.629	5.000	4.740	0.215	0.000	5.000	-3.909	rating-1-5
calculated_host_listings_count	27.992	1.000	4.000	5170.066	1.000	396.000	4.106	count
reviews_per_month	1.510	0.010	0.840	3.420	0.010	26.450	2.850	count-per-month
host_response_time	NaN	within an hour	NaN	NaN	NaN	NaN	NaN	string
host_is_superhost	NaN	f	NaN	NaN	NaN	NaN	NaN	boolean
host_has_profile_pic	NaN	t	NaN	NaN	NaN	NaN	NaN	boolean
host_identity_verified	NaN	t	NaN	NaN	NaN	NaN	NaN	boolean
neighbourhood_cleansed	NaN	Leopoldstadt	NaN	NaN	NaN	NaN	NaN	string
property_type	NaN	Entire rental unit	NaN	NaN	NaN	NaN	NaN	string
room_type	NaN	Entire home/apt	NaN	NaN	NaN	NaN	NaN	string
has_availability	NaN	t	NaN	NaN	NaN	NaN	NaN	boolean
last_review	NaN	2025-08-31 00:00:00	NaN	NaN	NaN	NaN	NaN	date
instant_bookable	NaN	t	NaN	NaN	NaN	NaN	NaN	boolean

Table 12: Top Pearson correlations for response variable reviews_per_month.

variable	pearson_corr
reviews_per_month	1.0000
number_of_reviews_l30d	0.7869
estimated_occupancy_l365d	0.6389
number_of_reviews_ly	0.6327
number_of_reviews	0.5184
host_acceptance_rate	0.2240

Continued on next page

Table 12: Top Pearson correlations for response variable reviews_per_month.

variable	pearson_corr
instant_bookable	0.2163
estimated_revenue_l365d	0.2082
host_is_superhost	0.1994
host_response_rate	0.1648
host_identity_verified	0.1393
accommodates	0.1021
availability_30	0.0788
review_scores_value	0.0561
review_scores_communication	0.0491
review_scores_checkin	0.0451
review_scores_cleanliness	0.0448
longitude	0.0281
host_total_listings_count	0.0260
review_scores_accuracy	0.0246
beds	0.0225
review_scores_rating	0.0192
calculated_host_listings_count	0.0153
host_has_profile_pic	-0.0143
review_scores_location	-0.0163
bedrooms	-0.0233
bathrooms	-0.0303
price	-0.0531
latitude	-0.0569
maximum_nights	-0.0577

Table 13: Top Pearson correlations for response variable price.

variable	pearson_corr
price	1.0000
host_total_listings_count	0.3201
estimated_revenue_l365d	0.2224
maximum_nights	0.1233
bedrooms	0.0983
accommodates	0.0804
beds	0.0782
instant_bookable	0.0658
latitude	0.0481
bathrooms	0.0475
review_scores_location	0.0408
review_scores_cleanliness	0.0278
host_response_rate	0.0154
host_has_profile_pic	0.0144
longitude	0.0109
review_scores_rating	0.0075
review_scores_accuracy	0.0062
availability_30	0.0047
host_identity_verified	0.0043
calculated_host_listings_count	-0.0022
review_scores_value	-0.0032
review_scores_checkin	-0.0072

Continued on next page

Table 13: Top Pearson correlations for response variable price.

variable	pearson_corr
host_acceptance_rate	-0.0090
review_scores_communication	-0.0147
minimum_nights	-0.0217
host_is_superhost	-0.0379
number_of_reviews	-0.0450
number_of_reviews_ly	-0.0475
reviews_per_month	-0.0531
number_of_reviews_l30d	-0.0585

Table 14: Top Pearson correlations for response variable review_scores_rating.

variable	pearson_corr
review_scores_rating	1.0000
review_scores_value	0.8368
review_scores_accuracy	0.8215
review_scores_cleanliness	0.7885
review_scores_communication	0.7144
review_scores_checkin	0.6716
review_scores_location	0.6292
host_is_superhost	0.2390
estimated_occupancy_l365d	0.1218
number_of_reviews	0.1114
number_of_reviews_ly	0.1042
estimated_revenue_l365d	0.0957
host_response_rate	0.0436
bathrooms	0.0406
minimum_nights	0.0330
host_has_profile_pic	0.0275
latitude	0.0245
host_identity_verified	0.0235
reviews_per_month	0.0192
number_of_reviews_l30d	0.0103
price	0.0075
longitude	0.0004
bedrooms	-0.0019
maximum_nights	-0.0043
beds	-0.0107
accommodates	-0.0307
host_acceptance_rate	-0.0460
host_total_listings_count	-0.1234
instant_bookable	-0.1630
calculated_host_listings_count	-0.1688

Table 15: Missing values per variable (absolute and relative).

variable	abs_number	rel_fraction
estimated_revenue_l365d	3817.0000	0.2703
price	3817.0000	0.2703

Continued on next page

Table 15: Missing values per variable (absolute and relative).

variable	abs_number	rel_fraction
beds	3811.0000	0.2698
bathrooms	3794.0000	0.2686
host_response_time	3710.0000	0.2627
host_response_rate	3710.0000	0.2627
host_acceptance_rate	3181.0000	0.2252
review_scores_value	2290.0000	0.1621
review_scores_location	2289.0000	0.1621
review_scores_communication	2289.0000	0.1621
review_scores_checkin	2289.0000	0.1621
review_scores_cleanliness	2289.0000	0.1621
review_scores_accuracy	2289.0000	0.1621
review_scores_rating	2289.0000	0.1621
last_review	2289.0000	0.1621
reviews_per_month	2289.0000	0.1621
bedrooms	1369.0000	0.0969
has_availability	987.0000	0.0699
host_is_superhost	354.0000	0.0251
host_total_listings_count	3.0000	0.0002
host_has_profile_pic	3.0000	0.0002
host_identity_verified	3.0000	0.0002
longitude	0.0000	0.0000
calculated_host_listings_count	0.0000	0.0000
instant_bookable	0.0000	0.0000
neighbourhood_cleansed	0.0000	0.0000
latitude	0.0000	0.0000
property_type	0.0000	0.0000
minimum_nights	0.0000	0.0000
room_type	0.0000	0.0000
estimated_occupancy_l365d	0.0000	0.0000
number_of_reviews_ly	0.0000	0.0000
number_of_reviews_l30d	0.0000	0.0000
number_of_reviews	0.0000	0.0000
availability_30	0.0000	0.0000
accommodates	0.0000	0.0000
maximum_nights	0.0000	0.0000

Table 16: Outliers per variable (IQR rule summary).

variable	abs_amount	rel_fraction	upper_out_bound	lower_out_bound
bathrooms	2682.0000	0.1899	1.0000	1.0000
host_total_listings_count	2379.0000	0.1684	62.0000	0.0000
minimum_nights	2359.0000	0.1670	6.0000	0.0000
calculated_host_listings_count	2157.0000	0.1527	38.5000	0.0000
host_response_rate	1888.0000	0.1337	1.0450	0.9250
number_of_reviews_l30d	1880.0000	0.1331	2.5000	0.0000
number_of_reviews_ly	1794.0000	0.1270	25.0000	0.0000
host_acceptance_rate	1777.0000	0.1258	1.1050	0.8250
number_of_reviews	1657.0000	0.1173	112.0000	0.0000
beds	1067.0000	0.0756	3.5000	0.0000
review_scores_checkin	950.0000	0.0673	5.3300	4.4500

Continued on next page

Table 16: Outliers per variable (IQR rule summary).

variable	abs_amount	rel_fraction	upper_out_bound	lower_out_bound
review_scores_value	937.0000	0.0663	5.3900	4.0300
review_scores_communication	926.0000	0.0656	5.3750	4.3750
review_scores_rating	849.0000	0.0601	5.5000	4.0600
review_scores_accuracy	838.0000	0.0593	5.4400	4.2400
review_scores_location	764.0000	0.0541	5.4950	4.0150
price	711.0000	0.0503	251.0000	0.0000
review_scores_cleanliness	604.0000	0.0428	5.5600	3.9600
reviews_per_month	595.0000	0.0421	5.0700	0.0000
accommodates	573.0000	0.0406	7.0000	0.0000
estimated_revenue_l365d	496.0000	0.0351	30724.5000	0.0000
longitude	342.0000	0.0242	16.4507	16.2694
latitude	281.0000	0.0199	48.2636	48.1443
bedrooms	242.0000	0.0171	3.5000	0.0000
maximum_nights	1.0000	0.0001	1641.2500	0.0000
estimated_occupancy_l365d	0.0000	0.0000	255.0000	0.0000
availability_30	0.0000	0.0000	35.0000	0.0000

Table 17: Modeled district-level opportunity scores based on predicted demand.

district	supply_listings	predicted_demand	opportunity_score_modeled
Favoriten	1328	2134.814602	1.607541
Meidling	677	1087.882095	1.606916
Landstraße	1206	1795.536652	1.488836

Table 18: Modeled district-level opportunity scores based on predicted demand.

district	supply_listings	predicted_demand	avg_predicted_demand	opportunity_score_modeled
Favoriten	1328	2134.814602	1.607541	1.607541
Meidling	677	1087.882095	1.606916	1.606916
Landstraße	1206	1795.536652	1.488836	1.488836
Brigittenau	601	850.343117	1.414880	1.414880
Rudolfsheim-Fünfhaus	1021	1391.005179	1.362395	1.362395
Mariahilf	558	749.853825	1.343824	1.343824
Simmering	226	300.358131	1.329018	1.329018
Neubau	679	880.071475	1.296129	1.296129
Wieden	487	619.314720	1.271693	1.271693
Innere Stadt	578	733.475985	1.268990	1.268990
Leopoldstadt	1540	1917.038398	1.244830	1.244830
Hernals	450	558.060507	1.240134	1.240134
Ottakring	791	967.848926	1.223576	1.223576
Penzing	455	551.242888	1.211523	1.211523
Margareten	759	864.109875	1.138485	1.138485
Floridsdorf	257	270.967853	1.054350	1.054350
Alsergrund	697	702.039837	1.007231	1.007231
Donaustadt	457	457.434364	1.000950	1.000950
Währing	396	395.229225	0.998054	0.998054
Josefstadt	335	269.550367	0.804628	0.804628

Table 19: Model performance comparison for nightly price prediction.

Model	rmse_log	mae_log	r2_log	rmse_euro	mae_euro	MAPE	MdAPE
HistGBR	0.085004	0.153145	0.804355	130279.030911	47.387487	0.158308	0.083768
RandomForest	0.175711	0.231467	0.595586	261088.828873	67.156243	0.230149	0.129975
ElasticNet	0.226708	0.308181	0.478213	266192.124138	73.971674	0.309913	0.221092
baseline_mean	0.434696	0.473917	-0.000490	279267.955870	92.525363	0.493937	0.380783
baseline_median	0.443334	0.471210	-0.020371	280217.195536	92.257157	0.453252	0.361111

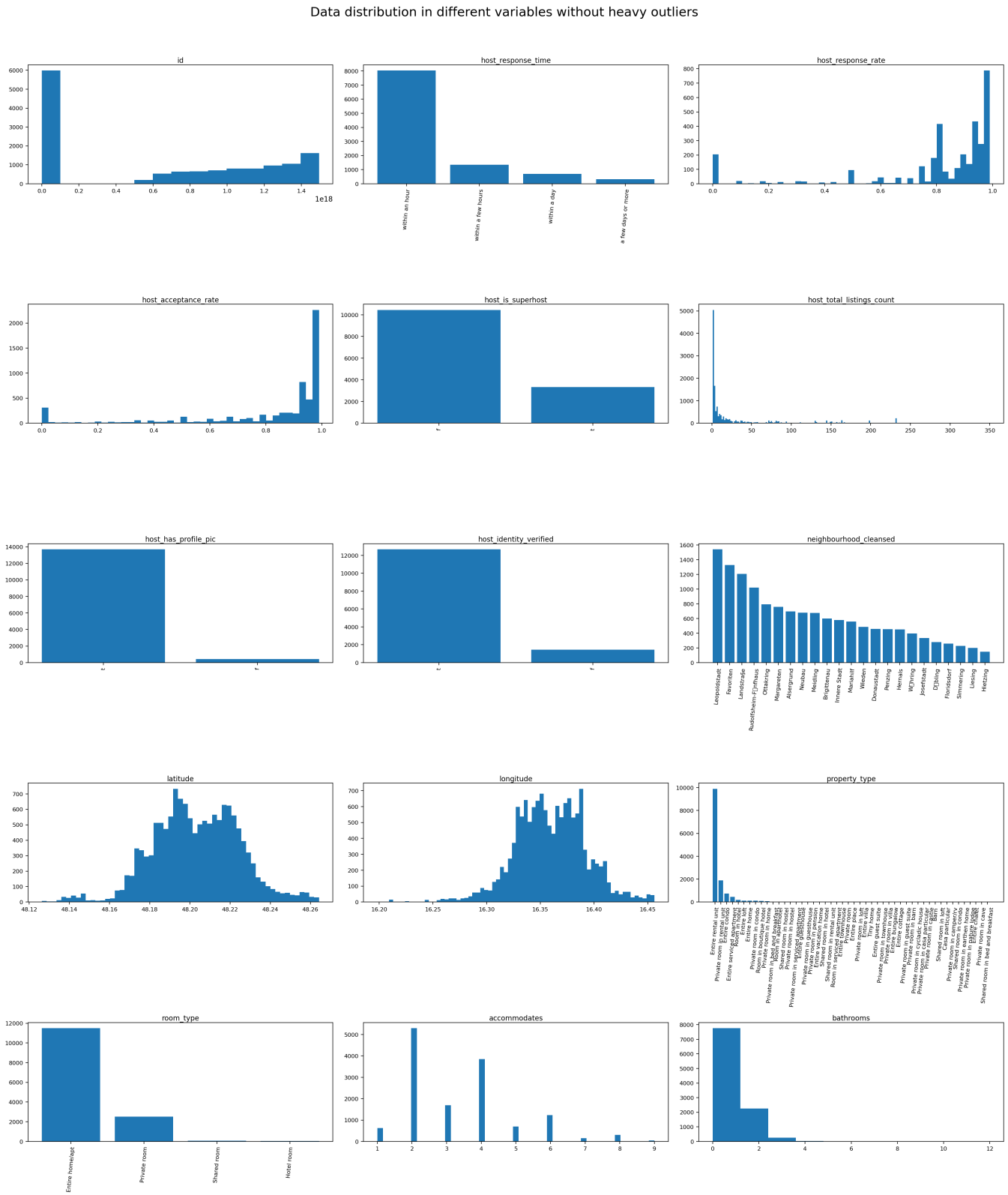

Figure 2: Outlier-robust distributions (part 1/3).

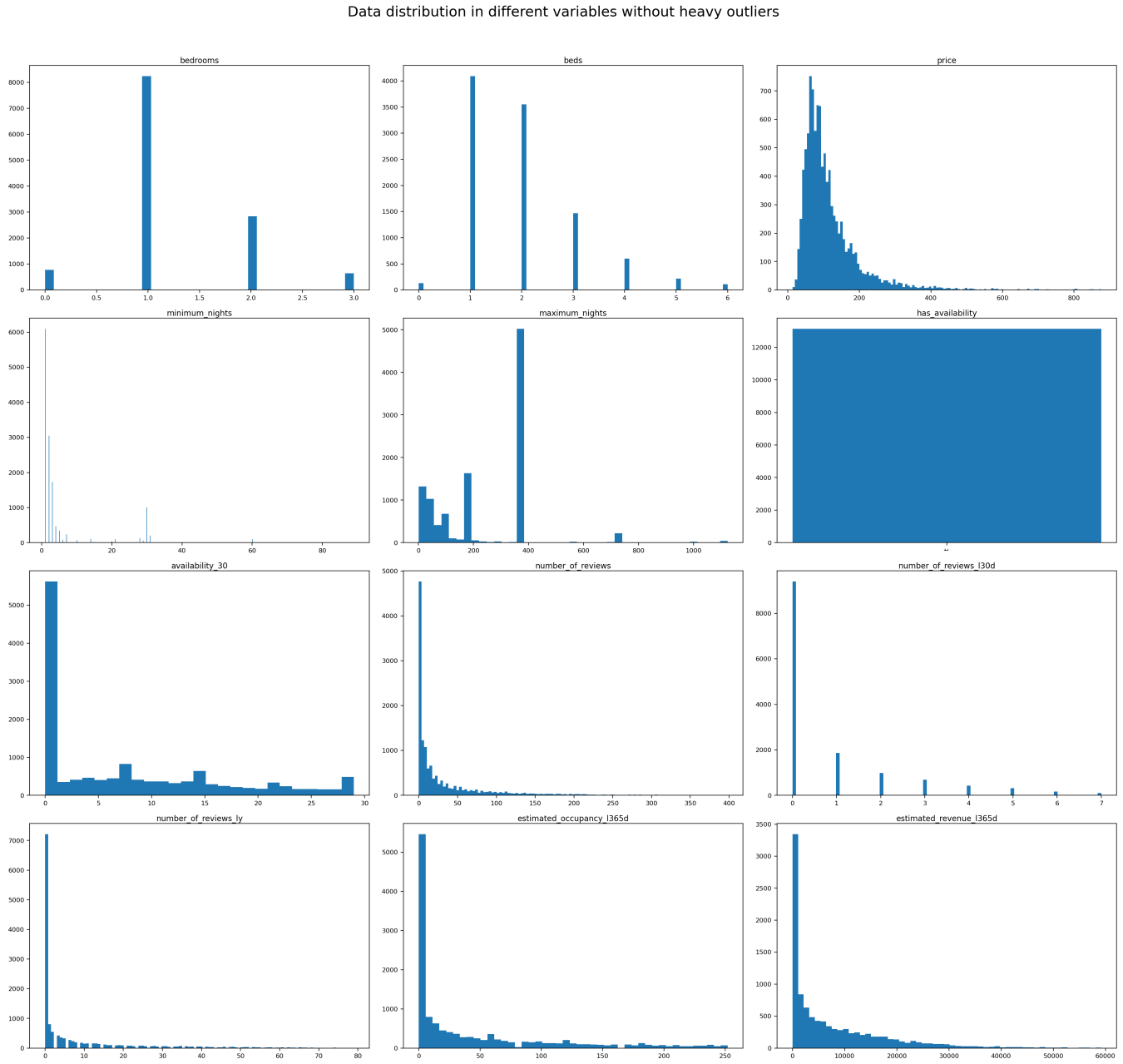

Figure 3: Outlier-robust distributions (part 2/3).

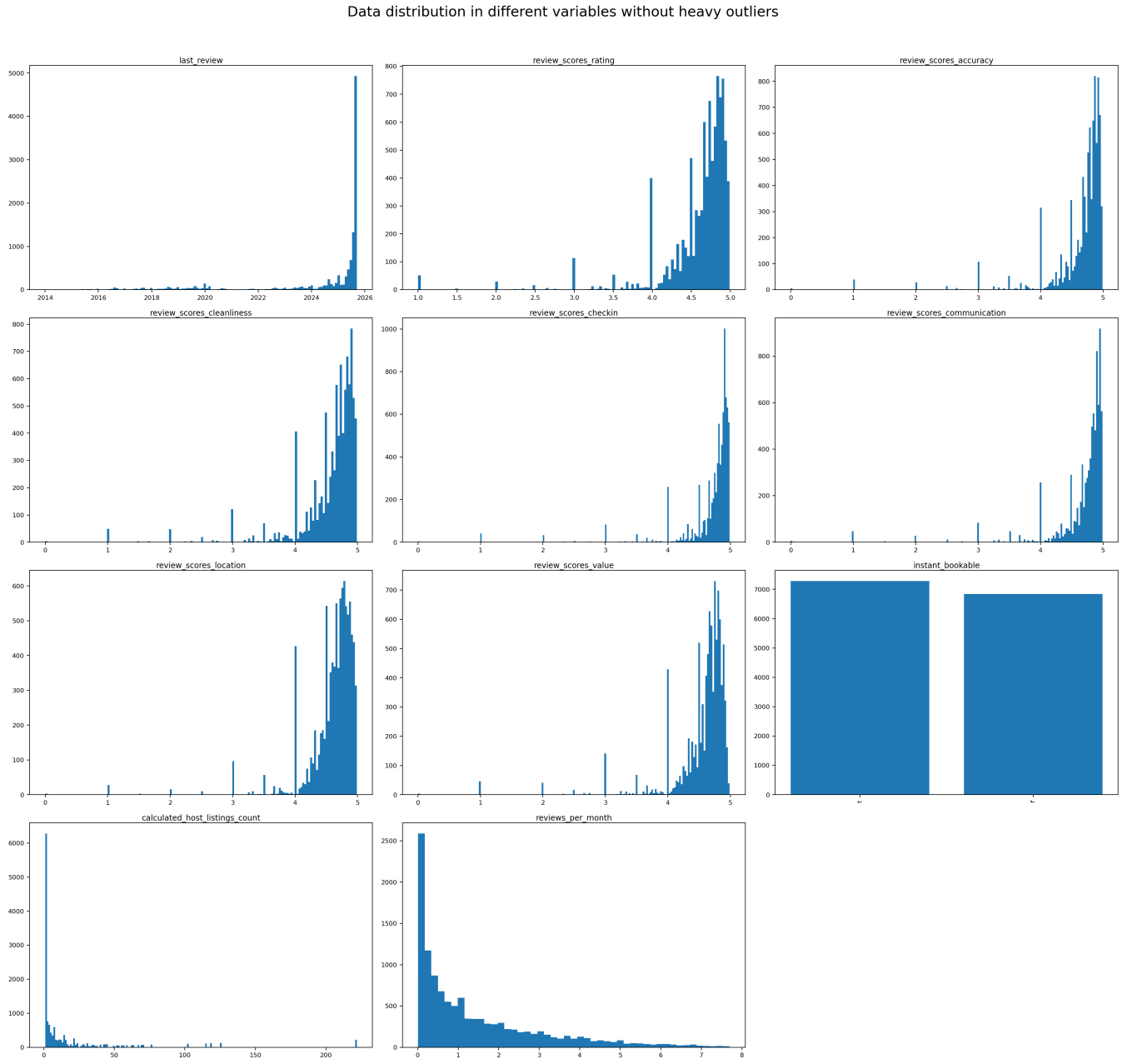

Figure 4: Outlier-robust distributions (part 3/3).