DataDriven Stays: Crafting Strategies for Success in the European Airbnb Market

Team A4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Julia Ungvary

Anna Vasconcelos

Cole Whittington

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Business Decision-Making with Data

QST BA 305

November 29, 2023

**1) Introduction**

1.1) Selecting A Dataset

For our term project, our team decided to analyze data on Airbnbs in different cities in order to see what factors influence pricing, and ultimately create algorithms that predict the price and guest satisfaction of an Airbnb. Before settling on Airbnb data, our team had a variety of ideas based on different datasets. These datasets included data on airline satisfaction, Boston-specific real estate data, and customer personality data. We decided that the Airbnb data was most interesting to us; therefore, we should use this dataset. Our team had a shared interest in Airbnb and felt there were multiple approaches we could explore with this dataset. There were a few datasets related to Airbnb on Kaggle, and we decided to go with the one with the most relevant features for analysis. This ended up being the dataset of Airbnb’s in Europe.

1.2) The European Airbnb Market Data

Our team had several different ideas for how to approach this dataset. Originally, our team thought the best approach would be to create a price predictor for Airbnb. However, we found it would be more helpful to create both a price predictor and a guest satisfaction predictor, along with suggestions on how Airbnb owners can increase both to maximize their own profit. We thought this would be a more interesting idea to explore, leading to more creative analysis for our team.

Our dataset had 19 features. These features included categorical information such as room type, the city the Airbnb is located in, whether the rental is a weekday or weekend, shared room, private room available, superhost or not, multiple rooms or not, whether the Airbnb is a business or not, cleanliness rating, guest satisfaction rating, attraction index, restaurant index, normalized attraction index, and normalized restaurant index. The dataset also included quantitative information such as price, person capacity, number of bedrooms, distance to the city center, and distance to the metro. Overall, there were 14 categorical features and five numerical features. The dataset described 41,000 Airbnbs, and the data was collected in 2023.

1.3) Objective Questions and Goals

Our objective was to find ways to improve Airbnb owners' business and help them choose better investment properties. As well as to help Airbnb owners strategically position themselves in an established market. We will look for solutions to this problem by exploring the following questions:

1. What are the most important features when classifying Airbnb listings with high guest satisfaction?
2. Which features should be focused on when deciding on the appropriate price for Airbnb listings?
3. How can Airbnb owners use this analysis to improve their business and choose investment properties?

**2) Data Preprocessing**

2.1) Removing and Cleaning Data Entries

After we established what dataset to use and defined our goals, we set out to clean and organize our dataset. The dataset we chose was already cleaned, so our next step was dropping unnecessary features from the dataset. We decided to drop the non-normalized forms of the restaurant index and attraction index. We decided to drop these because the impact of these features are captured in a normalized form of scale 0-100, making the non-normalized forms obsolete. Our team wanted to normalize the column names. We replaced the spaces in column names with underscores and made all column names lowercase. For the respective models, each continued a more thorough preprocessing procedure for the different outcomes we are trying to predict.

2.2) Checking Correlations

After we had this information, we decided to create a correlation matrix to understand the relationship between our different variables. We were looking to see what variables are of interest for our price and guest satisfaction models.

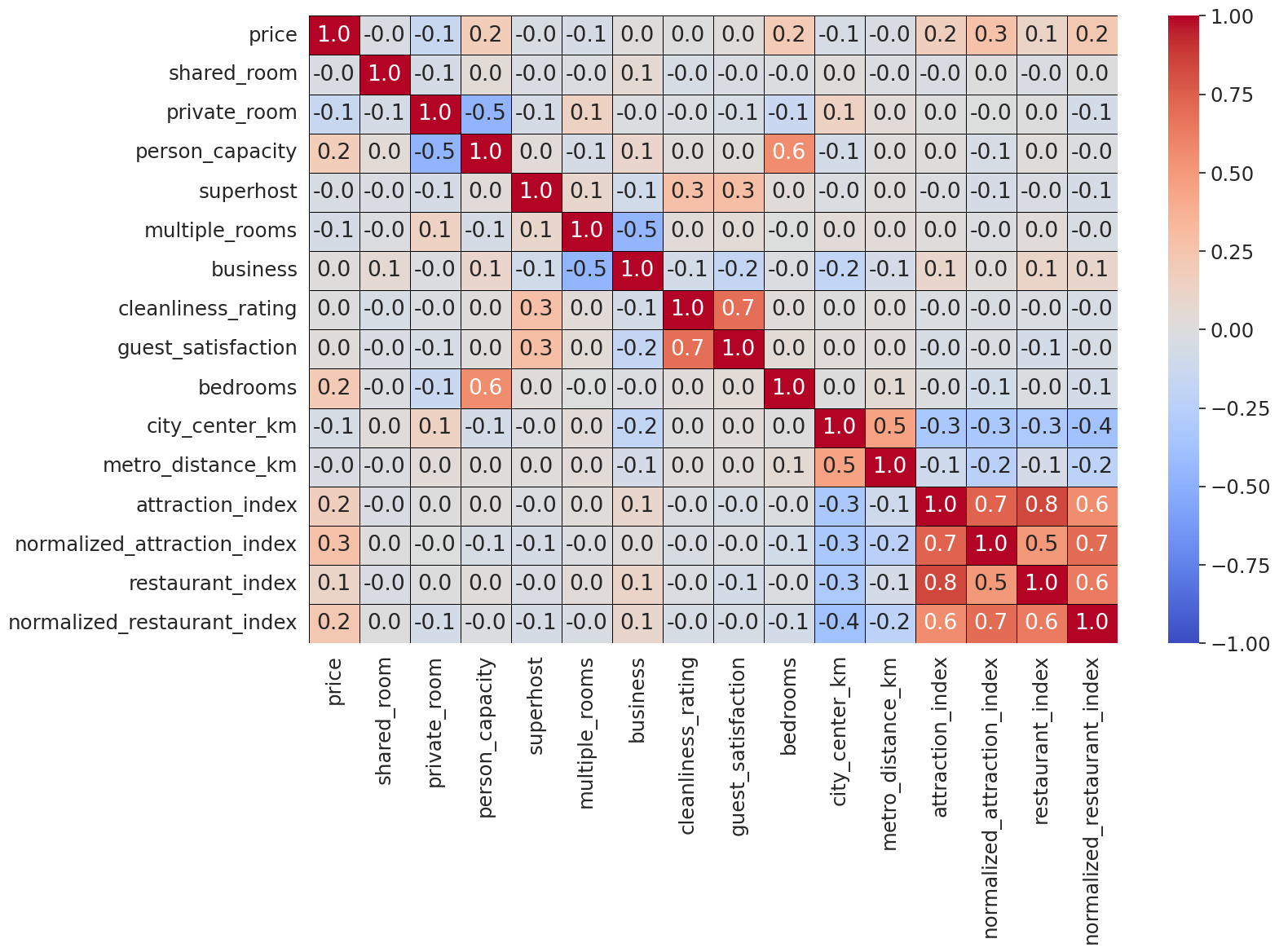


Figure 1: Seaborn Heatmap Indicating Correlations Between Features

To our surprise, all features had a correlation that was less than or equal to 0.3, indicating a weak correlation. Additionally, we noticed that the cleanliness rating was highly correlated (0.7) with guest satisfaction. This seems to indicate that cleanliness will be an important factor in determining price and guest satisfaction. Other initial dependencies are more intuitive, such as a high correlation between person capacity and bedrooms, and attraction index and restaurant index.

**3) Methodology and Analysis - Predicting Guest Satisfaction Algorithm**

3.1) Purpose

It is important for both investors and customers of Airbnb to understand guest satisfaction. For potential guests, the guest satisfaction rating is often a very significant, if not the most significant, factor in choosing an Airbnb listing. The primary goal of this classification analysis is to answer objective question 1: what are the most important features in listings when guest satisfaction ratings are high? Classification Decision Trees can provide insight into the listings’ primary reasons for receiving a high guest satisfaction rating. In addition to classification decision trees, the predictive power of the decision tree was compared to three other algorithms: random forest classifier, logistic regression, and K-Nearest Neighbors (KNN). A random forest classifier is chosen because it combines multiple decision trees to improve accuracy and reduce overfitting. Logistic regression is one of the simpler models and is used for efficiency in classifying. K-Nearest Neighbors is the best option when local patterns in datasets are important to cluster features together.

3.2) Initial Implementation and Configuration

The first decision to make during this analysis was the classification of the guest satisfaction variable. Within the dataset, this variable consisted of values from 0 to 100, with the latter being the best score. We chose to use classification in this analysis as opposed to regression because guest satisfaction is an ordinal scale, even though it initially represents a numerical value, meaning there are no consistent intervals between values, and classification can preserve the inherent ranking characteristic. Initially, there were many possible values we could take when binning the values into classes. We considered ten, four, three, and two different classes. We quickly found that splitting the variable into four or more classes would make our analysis more difficult without granting us more predictive power or interpretability. At this point, we chose to graphically represent the variable through a boxplot, discovering that the data was highly skewed left, as seen in Figure 2.

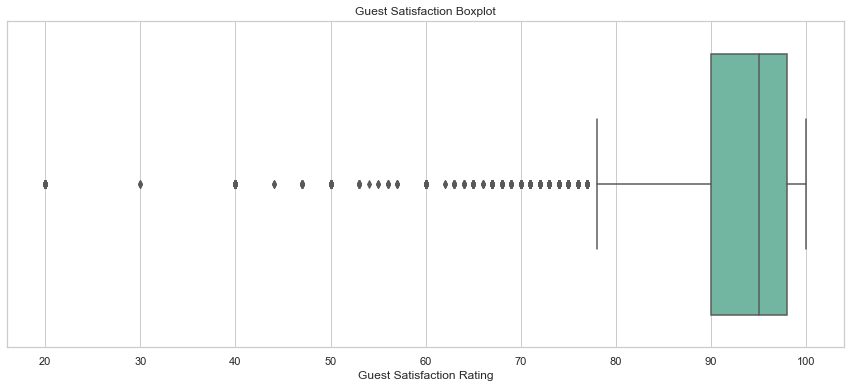
****

Figure 2: Guest Satisfaction Variable Boxplot

Using this boxplot, we found that the mean guest satisfaction rating was 93, and the 25th percentile was 90. This led us to divide the variable into three classes: low satisfaction below 90, high satisfaction at 98 and above, and moderately satisfied for every value in between. However, using this three-class approach led to poor performance of the K-Nearest Neighbor’s (KNN) model, with an accuracy score of 66.94%. We decided to compare this to the accuracy of a binary classification, and the best model had a significantly higher accuracy at 77.16%. Therefore, we chose to continue producing models based on the binary classification of the guest satisfaction variable, with highly satisfied being at 98 or above and every value below 98 being classified as not highly satisfied.

To accurately compare the predictive power of the four different algorithms, the data used for each algorithm was identical. Since KNN models only use metric variables in their prediction, the data used to train all four models only consisted of numerical features. This involved dropping the following variables from the dataset: city, day, and room\_type. One-hot encoding was not used in this analysis. The metric dataset was split into training and testing sets and standardized using the StandardScaler function from the Python package scikit-learn.

The only exception to this rule of identical datasets throughout each model was the implementation of principal component analysis. This primarily served as a method to resolve collinearity issues that could arise when implementing the KNN or Logistic Regression model. In addition, this method was implemented to reduce noise in the data and improve computational efficiency. The principal component analysis was used to reduce the standardized data from 13 features to five features. Five core components were chosen because their corresponding eigenvalues were greater than 1. This can be seen in Figure 3.1. These five components explain 62.4% of the variance. These five components can be seen in Figure 3.2. However, this reduced dataset was only used to observe the difference in accuracy scores when implementing the KNN and Logistic Regression models. This PCA data was not used to train any Classification Decision Tree or Random Forest models.

|  |  |
| --- | --- |
| Figure 3.1: PCA Scree Plot | Figure 3.2: Core Components Heatmap from PCA |

3.3) Naive Rule and Our Model

With each predictive algorithm implemented, using the Naive Rule as a benchmark to compare the accuracy score to prove how much better the specific algorithm performed. The benchmark for predicting guest satisfaction was an accuracy score of 67.44%. This was computed by assigning every observation to the majority class, ‘Not High Satisfaction.’

3.4) Model Implementation

3.4.1) K-Nearest Neighbors Model

Using the standardized data, the model was run numerous times using k values between one and 14 to find the k value with the highest accuracy score. As seen in Figure 4, a k value of 1 had the highest accuracy score at 77.16%. Although the accuracy on the testing data was high, the accuracy score on the training data was 100%, making it clear the model overfit the training data. By plotting the training accuracy versus the testing accuracy in Figure 4, it is clear that the best k value is 11 since it has a high accuracy score on the testing data and is similar to the training accuracy score. Following the same methodology, the k value of 11 also yielded the highest accuracy score for the PCA data without overfitting. When both models used a k value of 11, the model trained on the PCA dataset performed slightly worse than the original model, at 72.38%, achieving an accuracy score of 71.56%.

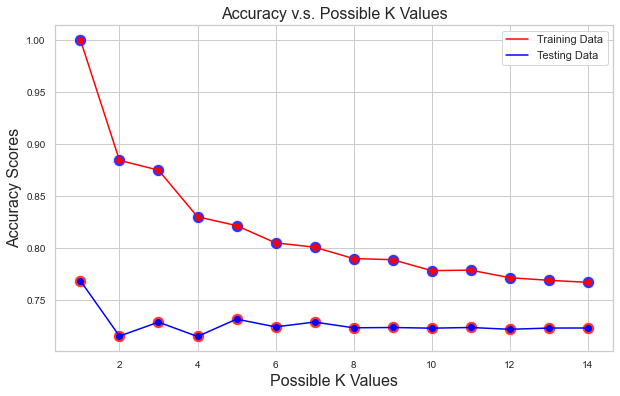


Figure 4: Accuracy v.s. Possible K Values

3.4.2) Logistic Regression Model

The Logistic Regression model was also trained on both the standardized data and the PCA data. When trained on the standardized data, the accuracy score of the single logistic regression model was 72.44%, failing to perform as well as the KNN model. When trained on the PCA data, the model performed slightly worse, with an accuracy score of just 70.29%.

3.4.3) Classification Decision Tree Model

The Classification Decision Tree Model used standardized data to train three different models. The first decision tree model had no parameters and built a full tree based on the training data. When the model was used to predict the test data, it achieved an accuracy score of 77.5%. Although the accuracy on the testing data was high, the accuracy score on the training data was 100%, making it clear the model overfit the training data. After choosing arbitrary values for the second model, the accuracy scores on the training and testing data sets were identical at 67.44%. These parameter values can be seen in Figure 5. This is equal to the Naive Rule benchmark, meaning the model has no predictive power. The final model’s parameters were determined by using a parameter grid search that compared multiple values of maximum tree depth, minimum sample leaf size, and minimum impurity decrease to produce the combination with the highest accuracy score. This grid search resulted in an accuracy score of 72.84%, with parameters that can be seen in Figure 5. Most importantly, the training set accuracy score of the final model was 75.38%. This small decrease in accuracy proves that this final model did not overfit the training data and, therefore, will generalize well to unknown data.

| **Guest Satisfaction** | **Model Type** | **Parameters** | **Accuracy** |
| --- | --- | --- | --- |
| Binary | Decision Tree | none/full tree | 77.5% |
| Binary | Decision Tree | Max\_depth =10, Minimum\_sample\_split = 40,  Minimum\_sample\_leaf = 20,  Minimum\_impurity\_decrease = 0.01 | 67.44% |
| Binary | Decision Tree | Max\_depth =20,  Minimum\_sample\_leaf = 100,  Minimum\_impurity\_decrease = 0 | 72.77% |

Figure 5: Classification Decision Tree Model Accuracy Table

3.4.4) Random Forest Model

The Random Forest model used standardized data to train multiple models with seven distinct values for the parameter associated with the number of estimators as seen in figure 6. Although all seven models had high accuracy scores, the model with 2000 estimators performed the best, yielding a score of 83.08%.

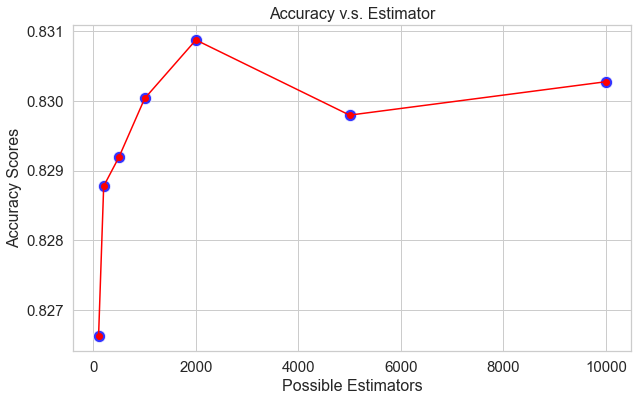


Figure 6: Accuracy v.s. Possible Estimators

3.5) Evaluating Prediction Performance

Across all four algorithms, the top-performing model was the Random Forest model with 2000 estimators. The following accuracy scores can be observed in Figure 7.Although the accuracy score for the fully grown decision tree is high, this model overfits the training data

| **Model Type** | **Parameters** | **Accuracy** |
| --- | --- | --- |
| Random Forest | N\_estimators = 2000 | 83.08% |
| Random Forest | N\_estimators = 500 | 82.92% |
| Decision Tree | none/full tree | 77.50% |
| KNN | n=11 | 72.38% |
| KNN (PCA) | n=11 | 71.56% |
| Decision Tree | Max\_depth =20, Minimum\_sample\_leaf = 100, Minimum\_impurity\_decrease = 0 | 72.84% |
| Logistic Regression | none | 72.44% |
| Logistic Regression (PCA) | none | 70.19% |
| Naive Rule | n/a | 67.44% |
| Decision Tree | Max\_depth =10, Minimum\_sample\_split = 40,  Minimum\_sample\_leaf = 20, Minimum\_impurity\_decrease = 0.01 | 67.44% |

Figure 7: Accuracy Metric Comparison for Different Models

3.6) Simulation of Recommendation Algorithm in Practice

To show how the guest satisfaction model would be implemented in practice, two potential properties were created. The first property, named property1, has features such as a shared room, close proximity to a metro station, a high normalized restaurant index, two-person capacity, and a 10.0 cleanliness rating. Using the Random Forest recommendation algorithm, the predicted satisfaction\_category is ‘High Satisfaction.’

The second property, named property2, has features such as eight bedrooms, a low normalized restaurant index, a 10-person capacity, and is located far from the city center. Using the guest satisfaction recommendation algorithm, the predicted satisfaction\_category is ‘Not High Satisfaction.’

These different predicted outcomes showcase the importance of features like the cleanliness\_rating, the distance to the city center, and the normalized\_restaurant\_index. Therefore, if an investor’s goal is to attain a high satisfaction rating, the recommendation algorithm suggests investing in property1.

**4) Methodology and Analysis - Predicting Price Algorithm**

4.1) Purpose

The significance of predicting Airbnb listing prices lies in its impact on affordability for the general population and in positioning listings competitive for revenue in densely populated metropolitan areas. For investors seeking pricing strategies, Regression Decision Trees serve as a recommendation algorithm, predicting prices based on listing characteristics. The interpretability of feature importance in decision trees aids in identifying the most significant factors. The primary goal is to answer objective question 2: Which features should be prioritized when deciding on the appropriate price for Airbnb listings? The algorithm’s important features provide valuable insight into crafting effective pricing.

4.2) Initial Implementation and Configuration

The Decision Regression tree was implemented by dummy coding categorical variables into numeric ones. The model was split into training and testing sets for evaluation without additional preprocessing. However, using the full tree resulted in a high Root Mean Square Error (RMSE) score of 300.10 and a low value of 0.14. This indicates that the model’s predictive accuracy is low, with only 14% of the variance in the target variable (price) explained by features in the decision tree.

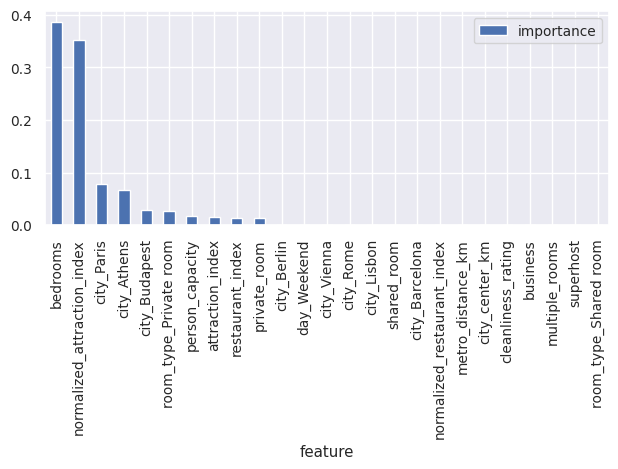


Figure 8: Full Price Decision Tree Feature Importance

As shown in this feature importance chart, the city that the listing is in has a significant effect on predicting the price. Further analysis would lead to more exploratory data analysis of the impact of specific cities on price.

4.3) Revision of Approach to Predicting Price

Further exploratory data analysis revealed significant variations in prices across different cities, showing diverse price ranges. To visually represent these disparities, a box plot was used (figure 8.1), highlighting the spread and identifying potential outliers. This visualization prompted the consideration of reorganizing the data by removing outliers.

|  |  | **Mean Price** | **Price Range** | **High Price** | **Low Price** |
| --- | --- | --- | --- | --- | --- |
| Amsterdam | 573.11 | 8001.78 | 8130.67 | 128.89 |
| Athens | 151.74 | 18502.57 | 18545.45 | 42.88 |
| Barcelona | 293.75 | 6874.11 | 6943.70 | 69.59 |
| Berlin | 244.58 | 5792.51 | 5857.48 | 64.97 |
| Budapest | 176.51 | 3716.45 | 3751.23 | 34.78 |
| Lisbon | 238.21 | 1610.46 | 1681.05 | 70.59 |
| Paris | 392.53 | 16352.88 | 16445.61 | 92.74 |
| Rome | 205.39 | 2372.29 | 2418.35 | 46.06 |
| Vienna | 241.58 | 13600.73 | 13664.31 | 63.58 |

Figures 9.1: Boxplot for Outlier Detection Figure 9.2: Overall Price Statistics

As shown in the box plot, with taking into account 40,000 different observations, there is a minority of outliers that are completely skewing the price data upwards. Using the properties of the Interquartile Range, and removing outliers that do not fit in that range, the dataset is reformatted, and the boxplot is rerun to ensure the range of prices is more feasible for prediction.

|  |  | **Mean Price** | **Price Range** | **High Price** | **Low Price** |
| --- | --- | --- | --- | --- | --- |
| Amsterdam | 354.11 | 397.21 | 526.09 | 128.89 |
| Athens | 144.35 | 479.76 | 521.64 | 42.88 |
| Barcelona | 227.11 | 456.86 | 526.45 | 69.59 |
| Berlin | 211.99 | 456.90 | 521.88 | 64.97 |
| Budapest | 167.49 | 481.04 | 515.82 | 34.78 |
| Lisbon | 230.25 | 455.68 | 526.27 | 70.59 |
| Paris | 299.22 | 434.57 | 527.31 | 92.74 |
| Rome | 197.23 | 490.91 | 526.97 | 46.06 |
| Vienna | 221.75 | 461.63 | 525.21 | 63.58 |

Figure 10.1: Revised Boxplot Figure 10.2: Revised Overall Price Statistics

Once the outliers are taken into consideration, the revised box plot and the associated table show a much more representative scale and range of the Airbnb dataset prices to then be used in a price recommendation algorithm.

4.4) Revised Implementation and Configuration

Specific to the price prediction algorithm, preprocessing involves dropping the guest satisfaction variable, as it was previously tested independently. Additionally, non-normalized values of restaurant\_index and attraction\_index are dropped, and the normalized versions are kept to maintain a consistent scale. Categorical variables are dummy coded, and the dataset is divided into training and testing sets for further analysis.

4.5) Naive Rule and Our Model

To accurately determine Airbnb listing prices, various prediction algorithms are tested for all of the nine cities. The Naive Rule serves as a benchmark for accuracy metrics, setting all prices to the average and then computing RMSE. The Regression decision tree provides interpretability by visualizing important features but may sacrifice prediction accuracy and is prone to overfitting. The Random Forest Regression Decision Tree improves accuracy through ensemble methods but loses interpretability. A ‘for loop’ efficiently assesses accuracy measures, visual representation, and feature importance for each city and each model, streamlining the computational process and maintaining consistency across cities.

4.6) Evaluating Predictor Performance in Competitor Analysis

The price prediction algorithm is crucial for Airbnb investors to identify key features influencing appropriate pricing. Segmenting the analysis into cities allows observations of how different locations prioritize certain features over others in pricing decisions. Although most cities share similar feature importance, discrepancies exist. The figure below illustrates algorithm performance across cities, highlighting the least error-prone algorithm and the top three feature importances for each city and algorithm. Additionally, it provides a graphical representation of RMSE variations between models.

| **City** | **Naive Rule (RMSE)** | **Decision Tree Regression (RMSE)** | **Random Forest Regression (RMSE)** | **Top 3 Important Features DTree Regression** | **Top 3 Important Features Random Forest Regression** |
| --- | --- | --- | --- | --- | --- |
| Amsterdam | 73.497 | 64.338 | 67.604 | Restaurant Index  Private Room  City Center (km) | Restaurant Index  Attraction Index  City Center (km) |
| Athens | 64.155 | 52.161 | 50.717 | Attraction Index  Bedrooms  Person Capacity | Attraction Index  Bedrooms  Person Capacity |
| Barcelona | 83.375 | 68.536 | 73.027 | Private Room  Restaurant Index  Business | Person Capacity  Restaurant Index  City Center (km) |
| Berlin | 81.993 | 63.389 | 63.638 | Private Room  Restaurant Index  Attraction Index | Private Room  Restaurant Index  City Center (km) |
| Budapest | 64.275 | 56.252 | 55.910 | Bedrooms  Private Room  Attraction Index | Bedrooms  Attraction Index  Private Room |
| Lisbon | 82.612 | 56.766 | 56.509 | Private Room  Bedrooms  City Center (km) | Private Room  Bedrooms  Restaurant Index |
| Paris | 81.727 | 69.124 | 68.601 | Attraction Index  Person Capacity  Cleanliness Rating | Attraction Index  Person Capacity  Cleanliness Rating |
| Rome | 76.382 | 51.012 | 50.605 | Restaurant Index  Private Room  Person Capacity | Restaurant Index  Private Room  Person Capacity |
| Vienna | 77.407 | 61.171 | 60.064 | Private Room  Attraction Index  Person Capacity | Private Room  Person Capacity  Attraction Index |

Figure 11.1: RMSE and Feature Importance Based on City and Model

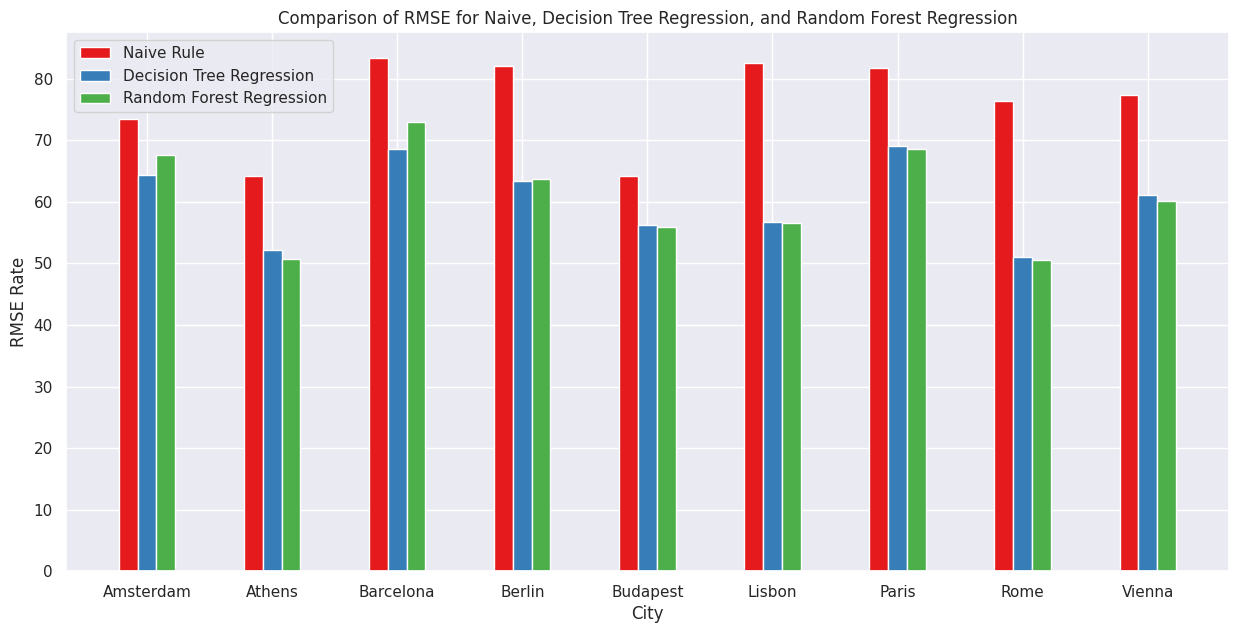


Figure 11.2: RMSE Visualization for Naive Rule, Decision Tree Regression, Random Forest Regression

The Random Forest Algorithm outperforms the Regression Decision Tree in 6 out of 9 city cases, indicating its overall superior performance. Thus, our model is established using the Random Forest Decision Tree algorithms as the primary model choice the majority of the time. Notably, two recurring predictors with a substantial impact on price are personal capacity and private room status. Additionally, the restaurant index proves to be crucial. This recommendation does not predict the highest price for a new owner but aims to suggest a competitive market price based on the features offered by the listing.

4.7) Simulation of Recommendation Algorithm in Practice

To demonstrate the real-world application of the price recommendation model, a hypothetical Airbnb owner instance, ‘new\_investor’ was created. This instance represents an entire apartment for rent in Rome, with features such as two bedrooms, a sleeping capacity of 4, a central location (1.2km from the city center and 0.9km from the nearest metro), and availability on the weekend. Utilizing the lower error-prone Random Forest Regression trees algorithm, the predicted competitive price for this listing is $210.14 per night.

For comparison, a similar listing for another hypothetical was evaluated with a changed value for the priority feature of the normalized restaurant index in Rome, considering all the same feature values except for the normalized restaurant index being changed to 50. This results in a predicted price of $233.62, which demonstrates the impact of feature importance. These results provide a valuable price point for Airbnb owners and investors to strategically position themselves in the market.

**5) Potential Areas for Improvement**

5.1) Feature Engineering

In the price prediction recommendation algorithm, all variables were used except for the unnormalized attraction and restaurant indexes. For future use, more informative feature engineering could include transforming the existing features or combining them to get more informative variables. An interesting feature to continue this study of predicting price would be the effect of weekends or weekdays in each of the local markets; this could be considered further as the primary model choice exploration.

5.2) Dataset Selection with Updating

The current dataset on Kaggle is not self-updating and was last updated eight months ago. A possible improvement would be to find a dataset that updates more often to have current Airbnbs to track how the market of Airbnbs changes to make the recommendation and prediction algorithms more up to date. This creates longevity of the algorithms to predict prices and guest satisfaction further into the future, rather than just the historical data that this current dataset provides.

**6) Recommended Strategies**

6.1) Strategies for Increasing Guest Satisfaction and Determining Price

For potential Airbnb owners, we first suggest looking at the city where the Airbnb will be located in order to determine what features are most important for that particular city. Features that were important across the board for guest satisfaction were the restaurant index, attraction index, city center (km), person capacity, private room, and bedrooms. Although a lot of these features are not in the hands of the Airbnb owner, there are still ways an owner can increase their price and guest satisfaction rating. If an owner is looking to invest in a property and rent it out on Airbnb, they should look to see what restaurants and attractions are in the area, in addition to evaluating the property’s distance to the city center. If an area has many good restaurants and attractions, in addition to being close to the city center, the owner can charge a higher price and receive better reviews for a stay in their Airbnb. Moreover, investors should look to invest in properties that have private rooms and can house more than two people, specifically for price. Lastly, cleanliness rating is the most important feature to ensure high guest satisfaction. We suggest keeping your listing as clean as possible. Whether it is choosing an easy-to-clean property or maintaining a regular cleaning schedule, that is essential to holding a high guest satisfaction rating. If investors look to accomplish these strategies, they can charge a competitive price for their rental and receive higher guest satisfaction.

**7) Conclusion**

7.1) Final Implications

The two features we focused on have a deeper implication to building a presence as an Airbnb owner and investor. Price has a more intuitive application as predicting a competitive price directly relates to earnings and revenue as an owner. Starting off as an owner, this model would give you a sense of where to position your listing in the market. Furthermore, guest satisfaction has ramifications that connect deeper to the business model of Airbnb upon further research. It is important to have high guest satisfaction because it means, as an owner, you can reach Superhost status. This is described as “provide outstanding hospitality, highly rated, experienced, reliable and responsive. And while each Superhost has their own unique style, they’ve earned their status by meeting, and often exceeding, guest expectations” (Airbnb). The average Superhost earns 64% more than the average Airbnb host, so as a result, if the host is a Superhost, then in turn, as a respected Airbnb host, a new owner will earn more revenue as a whole. By going above and beyond and creating a positive experience, hosts can grow and repeatedly have more consistent business with renters.

7.2) Closing Thoughts

Although our two prediction algorithms are not necessarily perfect, this project acted as an experience to use the skills from class in real-world practice. It was interesting to study a topic that has connections to a topic we find interesting both as consumers of the Airbnb app and as renters but also to better inform us as potential future Airbnb owners.

**8) Appendix and References**

8.1) Dataset Dimension Description

| **Name of Column** | **Column Description** | **Data Type and Range** |
| --- | --- | --- |
| City | Name of city | String |
| Price | Price of Airbnb | Float |
| Day | If it is a weekday or weekend for the booking | String |
| Room Type | Type of Airbnb - Entire apartment, private room, shared room | String |
| Shared Room | If the room in the Airbnb is shared by anyone | Boolean |
| Private Room | If the listing has private rooms available | Boolean |
| Person Capacity | The maximum number of people that can stay in the room | Integer |
| Superhost | If the Airbnb host is a Superhost or not | Boolean |
| Multiple Rooms | If the Airbnb has multiple rooms available | Boolean (1 true, 0 false) |
| Business | If the owner has more than 4 offers for their stay | Boolean (1 true, 0 false) |
| Cleanliness Rating | How clean the room/stay is rated | Integer (range: 2-10) |
| Guest Satisfaction | How satisfied was the guest after staying at that offering | Integer (range: 20-100) |
| Bedrooms | How many bedrooms are offered in the listing | Integer (range: 0-10) |
| City Center (km) | Listing distance from the center of the city in kilometers | Float (range: 0.02-25.3) |
| Metro Distance (km) | Listing distance from the nearest public transit in kilometers | Float (range: 0-14.3) |
| Attraction Index | Attraction index of the listing location | Float (range: 15 - 4513) |
| Normalized Attraction Index | Normalized attraction index | Float (range: 0-100) |
| Restaurant Index | Restaurant index of the listing location | Float (range: 20 - 6696) |
| Normalized Restaurant Index | Normalized restaurant index | Float (range: 0-100) |

8.2) Outside References

[https://www.Airbnb.com/d/superhost-guest#:~:text=Being%20an%20Airbnb%20Superhost%20is,and%20often%20exceeding%2C%20guest%20expectations](https://www.airbnb.com/d/superhost-guest#:~:text=Being%20an%20Airbnb%20Superhost%20is,and%20often%20exceeding%2C%20guest%20expectations).