Predicting Prices in Short-term rentals in Latin American Markets

I. Problem Definition

Digital platforms (e.g: Airbnb) have enabled a massive supply expansion in short term rental accommodation (listings) around the world. This has garnered attention and posed challenges for both the private and public sectors.

On the private sector side, hosts are interested in maximizing revenue to profit from their properties. To this end, many companies have enabled analytical tools (e.g: BeyondPricing) to optimize pricing. This requires understanding what predicts both price and bookings. On the public sector side, cities want to implement and enforce adequate rules regarding short term rentals. Otherwise, they will suffer from reduced tax revenues, reduced access to housing, and increased conflicts within neighborhoods. The most relevant listings for regulatory enforcement are the ones with high revenues, which is composed of price and bookings quantity.

This project seeks to develop some analytic tools for Rentur piloting them with preprocessed data from InsideAirbnb for Mexico City and Santiago de Chile. We are interested in explaining what determines the price of listings. The explanatory features are mostly defined through our domain-specific knowledge and entail a combination of location and other listing characteristics. The central questions we try to answer are: Can we accurately explain the variation of prices through machine learning modeling techniques? This is a regression type problem as the price is a continuous variable. A second question linked to the central one is: Does the explanatory model vary across Latin American capitals?

This is a four-week part-time project. To meet this deadline, we will constrain the analysis to two cities (Mexico City and Santiago de Chile). Initially, we proposed two regression problems instead of one (price and bookings) but we had to constrain the analysis to price due to the project complexity and time constraints. We will consider the project successful if we can develop a model that predicts a large part (50%+) of the variation in prices and bookings. If results confirm pre-analysis intuitions (e.g: location being the main driver of higher prices), that is also a valuable insight.

II. Data

Data Acquisition

Digital accommodation platforms do not share data publicly for independent analysis or public regulators. InsideAirbnb[1] is an independent oversight website launched in 2016 by housing activist Murray Cox. It provides cross-sectional data obtained through web scraping on Airbnb listings for 80+ cities around the world.

Given time constraints, we will work with the .csv files provided by InsideAirbnb instead of doing our own web scraping and integrating data from other sources. We utilize the March 2019 dataset for both Mexico

[1] www.insideairbnb.com

City and Santiago de Chile as this is the one that overlaps (Santiago has only one web scraped dataset). A geoson neighborhood file for geospatial analysis was also obtained from InsideAirbnb.

Main Features/Variables of Interest

The unit of analysis is a listing. Both datasets provide a large number of observations and features. Santiago has 15,790 observations and 106 features. Mexico City has 17,229 observations with 106 features. After a quick assessment of all the features, we determined that many features are not useful for the analysis as they do not have a logical relationship to price. We selected a preliminary list of the 36 most relevant features that would be kept for further analysis. The full list is presented in Annex 1.

Our dependent variable of interest is the price feature, defined as the price per night during a stay. There are some limitations to both. Price is the observed price in March 2019 (could change in high/low season) and is in string format and local currency units.

The relevant feature for price is:
‘price’ = price per night in local currency

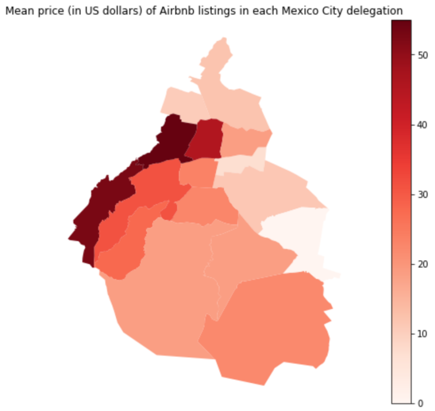
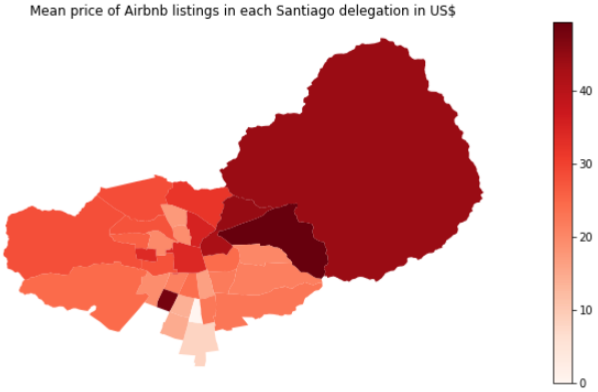
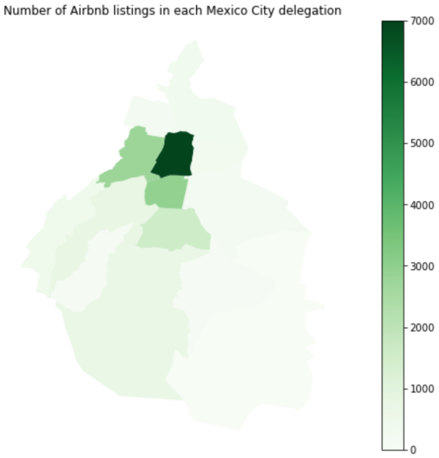
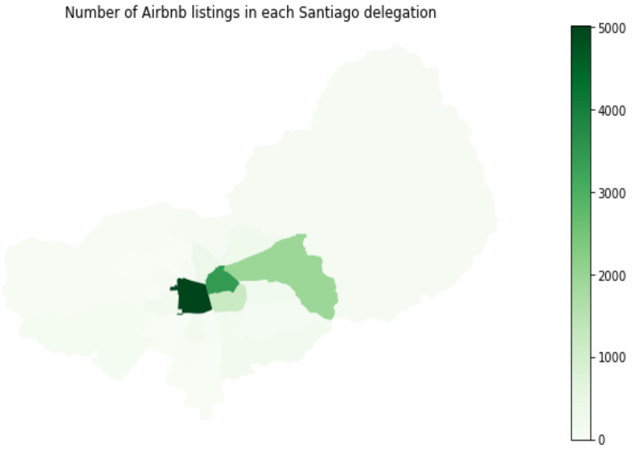
Examples of features for exploratory analyses are presented below (full list in Annex):
‘Id’ = ID for the listing
‘host_listings_count’ = number of listings the host has registered (if more than 1, it is a multi listing)
'neighbourhood_cleansed' = neighborhood where the listing is located (county/municipality)
'property_type' = many different categories [house, cave, etc]
'room_type' = room type refers to categories [entire home/apt, private room, shared room]
'accommodates' = number of people the listing can accommodate
'beds' = number of beds (or equivalent)
‘number of reviews’ = number of reviews in the listing
'first_review' = date/month/year of the first review
‘review_score_rating’ = review score (0 to 100)
‘'reviews_per_month' = reviews per month since started

Exploratory Data Analysis

We anchor our first parts of the EDA along with pricing and its distribution. By first understanding our dependent variables we obtain a better notion of what we are trying to predict. The price was transformed from string to numeric variable (float). Additionally, prices were transformed from the local currency to US Dollars. We used the average exchange rate per dollar in March 2019, which is 19.1343 Mexican pesos and 688 CLP respectively.

Price distribution presented quite a few outliers. Some prices were USD$0 which does not make any economic sense, and a few data points were as high as US$10,000. As these outliers would distort the overall analyses, we decided to use the outlier criteria of 1.5 times the Interquartile Range to classify them. According to these criteria, outliers accounted for 5.0% of the observations in Santiago and 6.9% for Mexico City. Figures A.1 and A.2 show the price distribution for Mexico City and Santiago.

Using geopandas, each city was divided into their respective neighborhoods and the mean price was computed for each neighborhood. We also plotted the distribution of listings across both cities. Figures 1 and 2 depict the price distribution across different neighborhoods in Santiago and Mexico City. Price variations vary strongly across neighborhoods with the highest values located in richer neighborhoods (e.g: Las Condes in Santiago and Polanco in Mexico City. The number of listings is heavily concentrated in middle-high income neighborhoods and downtown areas (Santiago, Providencia and Las Condes for Santiago, and Miguel Hidalgo, Cuajimalpa and Cuauhtemoc for Mexico City) as Figures 3 and 4 show. Taken together, this suggests that location plays a key role and that including neighborhood specific variables (dummies) is important for explaining variation.

	Mexico City	**Santiago**
	 Mean price (in US dollars) of Airbnb listings in each Mexico City delegation Figure 1	 Mean price of Airbnb listings in each Santiago delegation in US$ Figure 2
	 Number of Airbnb listings in each Mexico City delegation Figure 3	 Number of Airbnb listings in each Santiago delegation Figure 4

Feature Engineering and Data Preparation

There is relevant feature engineering and data preparation to be done before running regressions. A few features must be type processed (e.g: they are in string format instead of integer/float). Some features have information that is irrelevant for the analysis (e.g: room type shows 27 hotel rooms in Mexico City). There are some missing values that are unusual (e.g: listings with no bathrooms). Let's explore the changes we added to further improve the data.

The 'room_type' is one of the key features of interest. Initially, we decided to consider the standard distribution from Airbnb, categorizing them in: (1) Shared Room, (2) Private Room and (3) Entire home/apartment. As Shared Room accounted for less than 5% of all the listings, we pooled shared and private room together. In the case of Mexico City we also detected Hotel Rooms as an extra category and deleted said observations (Traditional accommodation like hotels can use platforms like Airbnb but they are not our analysis target). Figures A.3 and A.4 (Annex) show the supply for both listing types with Santiago having more "Entire Home/Apt" and Mexico City offering more rooms. Both cities are consistent with the notion of entire properties taking nearly half of the listings.

Location is intuitively one of the most important features for deciding where to book a listing. The variable 'neighbourhood_cleansed' was in string format for each jurisdiction within the city. Dummies were created for each neighbourhood through one hot encoding. As previously shown, there is a large concentration of listing in the neighborhoods close to central areas and with high incomes.

Some variables with NaN values were subject to data imputation using always conservative approaches. 'bathrooms' were rounded to 1 in case there were none, since we considered that Airbnbs should have at least 1 bathroom. For 'bedrooms' and 'beds' we imputed the median value as to not bias our results.

The feature 'zipcode' was dropped from our preliminary list of 36 features because of a large percentage of missing values (90%+, see Figure A.5 in Annex) and no reasonable approach for missing those gaps. In consequence, our spatial analysis can go as granular as the neighborhood level.

'host_since' is a column specifying when the host joins Airbnb. This might be important for prices as the older the host, more reviews will be available that can impact positively or negatively the listing price. This variable was converted to date the "host_since" variable was in string format and also created a number of days hosting variable.

Ratings ('review_scores_rating' 'review_scores_value') will be grouped into bins. Figures A.6 and A.7 in Annex show the used histogram to decide on useful bins. The majority of ratings are 9 or 10 out of 10, which does not allow much distinction between listings. Therefore for these columns, 9/10 and 10/10 will be kept as separate groups, and 1-8/10 will be binned together. We imputed missing values as the median.

Multi-listings are also an interesting variable ('calculated_host_listing_count'). Hosts with more than one listing might be commercial operators with higher managerial expertise and use different pricing strategies than single listing hosts. Both cities show a similar pattern of multi-listings with most hosts (around 8,000 in each dataset) having only one listing (Figures A.8 and A.9).

For the regression we decided to use one hot encoding by utilizing the Pandas function get_dummies(). It is necessary to turn the strings containing the neighbourhood names into integers to be able to utilize the regressor to do the modeling.

After all the feature engineering and data cleaning, correlation matrices were used to identify multicollinearity in the data along with low correlation features that we expect to have low predictive power (Figures A.10 and A.11).

III. Analyses Approach

III.A Santiago Models

For the Santiago notebook we used one hot encoding on the top 5 most occurring neighbourhoods, which makes 97.5% of the dataset, the rest of the neighbourhoods were put into a “nb_Other” neighbourhood column. This column was then dropped, much like the drop_first parameter one would regularly use. This allows us to present the same amount of data, but with one less feature column, as the information is already given by the one-hot-encoding. Right before the regression was to begin, we dropped columns that either were directly dependent on price, were strings or had no value to our model. Here we dropped features like the host_name, host_id and estimated income (which was deduced from the price). Since there are many data types representing numbers, we decided to include all numeric columns, and excluding all others.

We used the StandardScaler to scale our dataset, making the mean equal to 0 and the variance equal to 1 unit. By scaling, we allow our model to be as objective as possible, and reduce the distance between values of different features. After scaling we dropped the label column (price) from the train_data and made the features sub dataset, and stored the dropped value as the labels series.

After differentiating these sets we used the train_test_split() function provided by sci-kit learn. For our experiments, we used a train-test-split of 0.8/0.2, which resulted in 9,518 instances in our training set and 2,378 in our test set. We also used a fixed random state to ensure consistent results and reproducibility for our experiments.

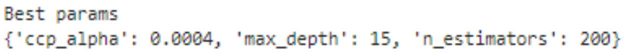
In the first iteration of training our regression model, we had allowed features such as host_id, however, we figured out that this did nothing but confuse the model, so we dropped it for the next iterations. Furthermore, we had coded the feature “availability” as a string, not an integer, so we had to change this.

Initially, we wanted to try different regression models to scope in on the one that was the most accurate from the get go. The contesting models were MLPRegressor, KNeighborsRegressor and RandomForestRegressor. Their representative score() function outputs are listed in the table below. The R^2 score is a score with respect to y.

Score R^2	MLPRegressor	KNeighborsRegressor	RandomForestRegressor
Train	0.6619	0.7208	0.9445
Test	0.6450	0.5594	0.6585

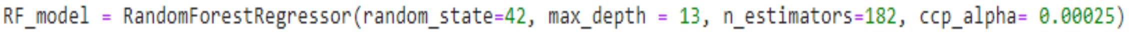
From this table outputs, we choose to pursue the RandomForrestRegressor (RFR) model. RandomForest uses decision trees (DT) as the base learner. DTs have low bias, thus the high train score, and high variance, however, since we aggregate over many trees and take the mean, we end up with low variance. The RFR was very fast to overfit the training set. So we reduced the train set size to 0.5 in this model. We tested with the MinMaxScaler as well and found out that we got better results with this scaler.

For hyperparameter tuning, we used GridSearchCV, which uses cross validation to get a more complete view of the model by testing it on unseen data during training. Using cross-validation is important, because we have to refrain from using the test set in any way during training. The grid search is an exhaustive search through values we specify for some of the parameters of the individual models. This helps us to find the best hyperparameters for our model, see Figure A.12 in the appendix. Then we can re-evaluate the model using our results from the hyperparameter tuning. After using the grid-search, we could manually test for different values around the best parameters, all in the hunt for the best possible test score. From the grid search, we found these to be the best hyperparameters, and it gave a score of 66.44%.



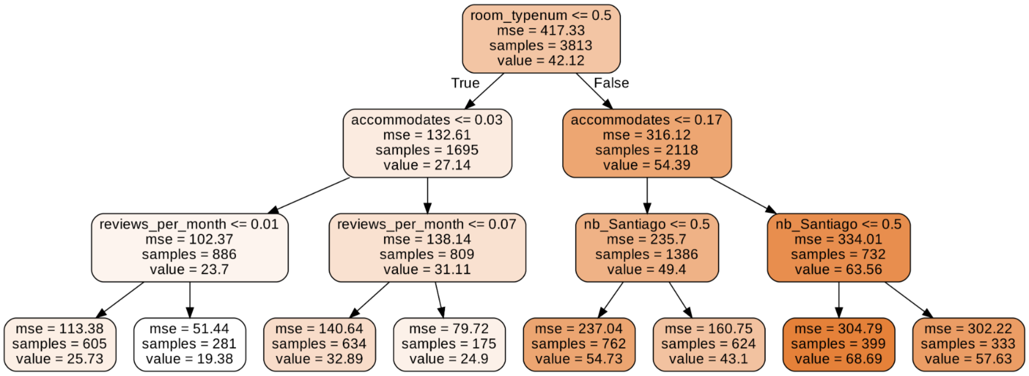
```
Best params
{'ccp_alpha': 0.0004, 'max_depth': 15, 'n_estimators': 200}
```

It was interesting to see that the gridsearch chose a relatively small number in max_depth. With a small number here, there is less chance of overfitting, and provide a model that is more explainable from a user's point of view. Then, we manually went up and down from these until we found these hyperparameters as the best ones (Santiago):



```
RF_model = RandomForestRegressor(random_state=42, max_depth = 13, n_estimators=182, ccp_alpha= 0.00025)
```

This resulted in the score of 66.92%. We were not able to increase the test score by that much, so we have understood that one has to fight for each decimal of score in the models. The evaluation of the model came next, and we wanted to firstly try to visualize the tree and the nodes. Therefore, we produce a small tree of the RFR model with max_depth set to 3 (Santiago):



In this tree we see that the 'room_typenum' is the root node. This means that by splitting on this node we gain the most information. We see that the mse is highest here. The full tree was too extensive to visualize in this report.

We computed the feature importance plot for this model and saved for comparison to Mexico (Figure 5 below). Some of the numeric values of the ranking is shown in Figure A.12 in the appendix.

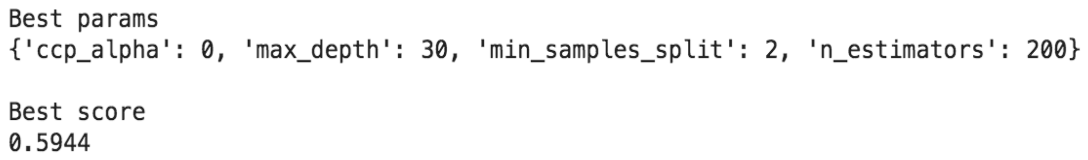
III.B Mexico City Models

In the case of Mexico City, we decided to run an Ordinary Least Squares (OLS) linear regression model without splitting the data, just to test a baseline model and see how it performs on our data. String variables such as 'id', 'host_id', 'host_total_listings_count', 'host_since', 'room_Shared room', 'room_type_num' were dropped and the dependent variable, price, was created, as well as an independent set of relevant variables that have been previously mentioned. The set was scaled to be standardized with mean equal to 0 and the variance equal to 1. From there, the linear regression was carried out with the presence of outliers and, as expected, accuracy results were made on the model. The OLS regression had an accuracy of 0.055 (Graphic results shown on Figure A.13) .

On a second approach we decided to run an OLS splitting data into train and test sets with the train_test_split() with 80% to train and validate the model and 20% to test the final model and without outliers. The result for splitting the data was 12829 train and 3208 test observations. We also used a fixed random state to ensure consistent results. The linear regression was made for the standardized data (scaled and fitted data) testing the dependent variable against the independent variable on their trained partition. By taking for this second approach a cleaner data, the test accuracy showed an improvement although not a in a grand manner with a 0.46 result (Graph of this model shown in Figure A.14).

For a third approach into modeling, we decided to run a polynomial regression of second degree, since this is an effective tool in feature engineering which can identify features that should be transformed to a higher degree or potentially important feature interactions. The polynomial features were split once more into train and test sets and strandarized. For this model a linear regression was run with the fitted values obtaining the following array: array([0.40454362, 0.4950521, 0.36513675, 0.44042075, 0.12028832]). For this third approach, our model had a better predictability shown in Figure A.15.

For a final and fourth approach, it was decided to run the model undel a Random Forest Regressor where we took the split data. By utilizing a Random Forest Regressor with the GridSearch tool it was found that the best hyperparameters with the test parameters (Figure A.16) and best score as follows:



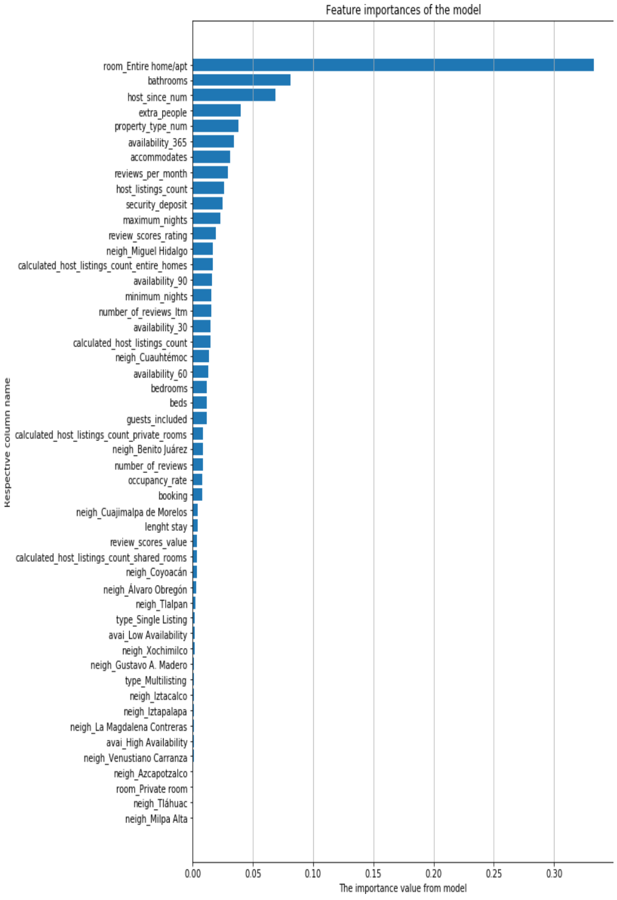
```
Best params
{'ccp_alpha': 0, 'max_depth': 30, 'min_samples_split': 2, 'n_estimators': 200}

Best score
0.5944
```

Then, we manually went up and down from these parameters in order to see if it was possible to increase the accuracy, but attempts didn’t get a better fit. In conclusion, the GridSearch tool with RandomForest proved to be the best approach for the Mexico City dataset with an improvement of 0.006 under a score of 0.5944.

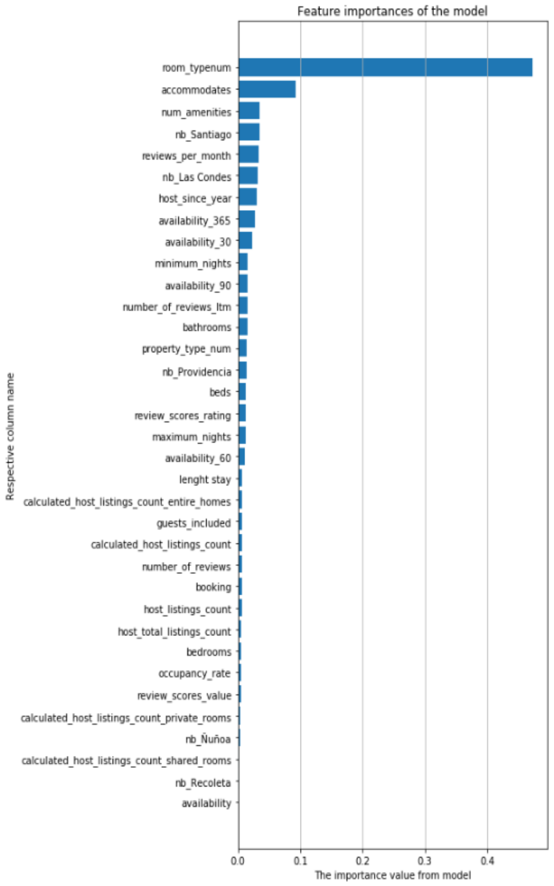
III.C Features Importance Comparison

Figure 5

Mexico City



Santiago



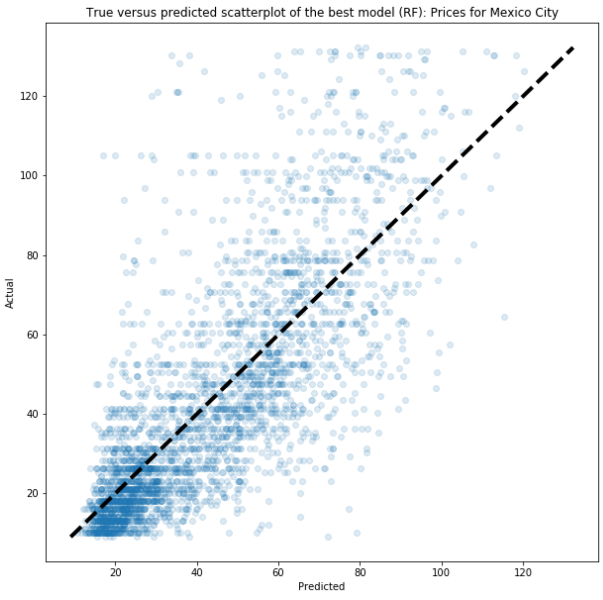
III.D Challenges and Solutions

Many of the challenges and solutions were described in the previous subsection (feature engineering, iteration in the model, etc.). A more general challenge refers to working with two datasets independently. This was interesting as it enabled us to discover different processes and methods of exploring and cleaning data. For Santiago, more sophisticated models were executed and for Mexico a more basic approach was made. We explored the richness of different models and could compare their flaws and richness. Nevertheless, sometimes the homologation of both results was a challenge when we wanted to compare our results. In the end we were able to sort out the differences between notebooks and coding, but we could have developed a different methodology for digging in the dataset.

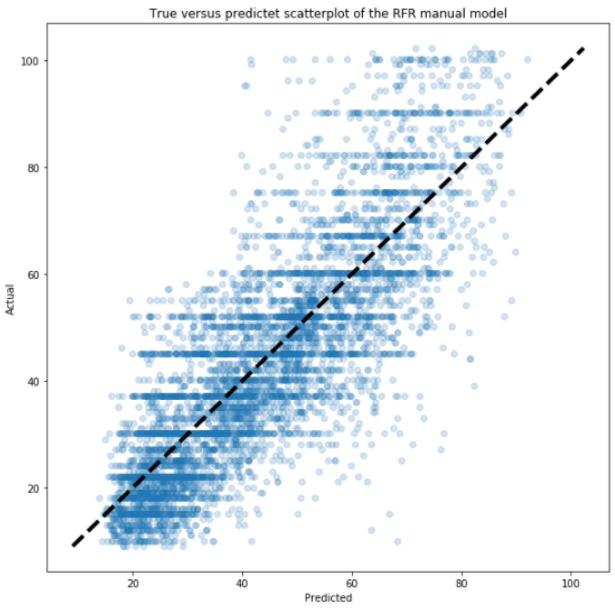
IV. Analysis Insights

Both cities came to the conclusion that the listing type was the most important factor with respect to the price. The type of listing was 35-47 % of the prediction our model came up with. For Santiago, the number of accommodations was the second most important feature with 9% of the prediction. For Mexico City it was the type of listing, in which an entire property was 35% of the model's prediction. Other important features for Mexico City's market are the number of bathrooms (10%) and the host's experience in Airbnb (7%). One hypothesis is that since Santiago has more entire homes listings than rooms (Figure A10), and Mexico City the opposite (Figure A9), their feature importances are a bit different. For Santiago, as mentioned above the second most important feature was the number of accommodations, which in an entire home listing setting makes sense. For Mexico City, where there are more stand alone rooms, features like bathrooms, the host's experience and the cost of extra people are more important.

Predictions for Mexico City:



Predictions for Santiago:



Upon looking at the prediction plot, there are horizontal lines made up by many data points at certain intervals. Our hypothesis is that this may come from the fact that prices often are for example set just under intervals of round numbers, like for example 29, 39, 49 US$. Further, the lines don't line up to the round numbers of the scale since the currency is in another currency. Also in Figure A.18, we see the heatmap of the predictions versus the actual values of the Santiago data set. It is here easy to see that the model was proficient at predicting prices of US$ 25 and US$ 50. This is interesting since we know from Figure A.2 that these prices are very close to the means of the 'room' and 'entire home' listing types respectively. The actual prices of the mean are 26.9 and 54.6 US$.

V. Future Work

Further work could go in multiple directions using these same datasets. For example, splitting the current dataset and working with the two listing types (full property versus private/shared room). Some of the evidence found in this study points that also the specific property type (e.g: Bungalow, Hotel, Bed &

Breakfast) might be interesting to explore (see Figure A.10). One could add features that we did not explore in this study such as amenities from a listing.

A complementary dependent variable that is crucial for both the private and public sector is bookings. The total revenue generated by a listing is depending on price per night and the number of nights. By proxying bookings with the number of reviews, we could run a similar exercise to those we did in this study and get better insights. It may be the case that what drives price is different than what drives bookings.

Improvements in the existing dataset could also help better understand the drivers of listing prices. For example, there is a large amount of missing values in interesting variables like zip codes (see Figure A.5). By leveraging latitude and longitude, a researcher could find and input the zipcodes to add a new level of granularity to the spatial analysis. We do consider that location is really important when predicting the price and it might not be fully captured by the neighborhood variable. .

Technically, more experiments and use of other models in depth (like the MLPRegressor) could be interesting. It might be the case that MLPRegressor can yield better scores after tuning parameters than our preferred RandomForestRegressor.

On that note, the utilization of RandomizedSearchCV could have been implemented, or used in the very start, to try to find “hidden” or special combinations of hyperparameters that could have resulted in even better results. Aslo, doing an extensive booking analysis would prove interesting in this data set. Booking and the actual activity would give a broader picture of the listings, yielding the opportunity to do more extensive analysis.

Annex

Annex 1 - Preliminary list of 36 features selected for the data analysis

Features appear in both datasets.

‘price’ = price per night in local currency
‘id’ = ID for the listing
‘host_id’ = ID for the host
‘host_name’ = name of the host
‘host_since’ = day/month/year when the host registered (not necessarily this listing)
‘host_listings_count’ = number of listings the host has registered (if more than 1, it is a multi listing)
‘host_total_listings_count’ = number of listings the host has registered (if more than 1, it is a multi listing) [ended up being the same as the previous one
'neighbourhood_cleansed' = neighborhood where the listing is located (county/municipality)
'city' = city where the listing is located
‘zipcode' = zipcode of the listing (more granular than neighborhood)
'property_type' = many different categories [house, cave, etc]
'room_type' = room type refers to categories [entire home/apt, private room, shared room]
'accommodates' = number of people the listing can accommodate
'bathrooms' = number of bathrooms
'bedrooms' = number of bedrooms
'beds' = number of beds (or equivalent)
'amenities' = string text with all the amenities included (e.g: Wifi, breakfast, washer)
'guests_included' = number of guests included in the base price
‘extra_people’ = number of extra people allowed
‘minimum_nights’ = number of minimum nights per stay
‘maximum_nights’ = number of maximum nights per stay
‘availability_30’ = # days available for booking in the next 30 days
‘availability_60’ = # days available for booking in the next 60 days
‘availability_90’ = # days available for booking in the next 90 days
‘availability_365’ = # days available for booking in the next 365 days
number of reviews’ = number of reviews in the listing
'number_of_reviews_ltm' = number of reviews last month
'first_review' = date/month/year of the first review
‘last_review' = date/month/year of the last review
‘review_score_rating’ = review score (0 to 100)
‘review_score_value’ = review score (1 to 9)
'calculated_host_listings_count' = number of accommodations registered by the host
'calculated_host_listings_count_entire_homes' = number of accommodations registered by the host that are entire homes
'calculated_host_listings_count_private_rooms' = number of accommodations registered by the host that are private rooms
'calculated_host_listings_count_shared_rooms' = number of accommodations registered by the host that are shared rooms
‘'reviews_per_month' = reviews per month since started

Figure A.1 Price Distribution in Mexico

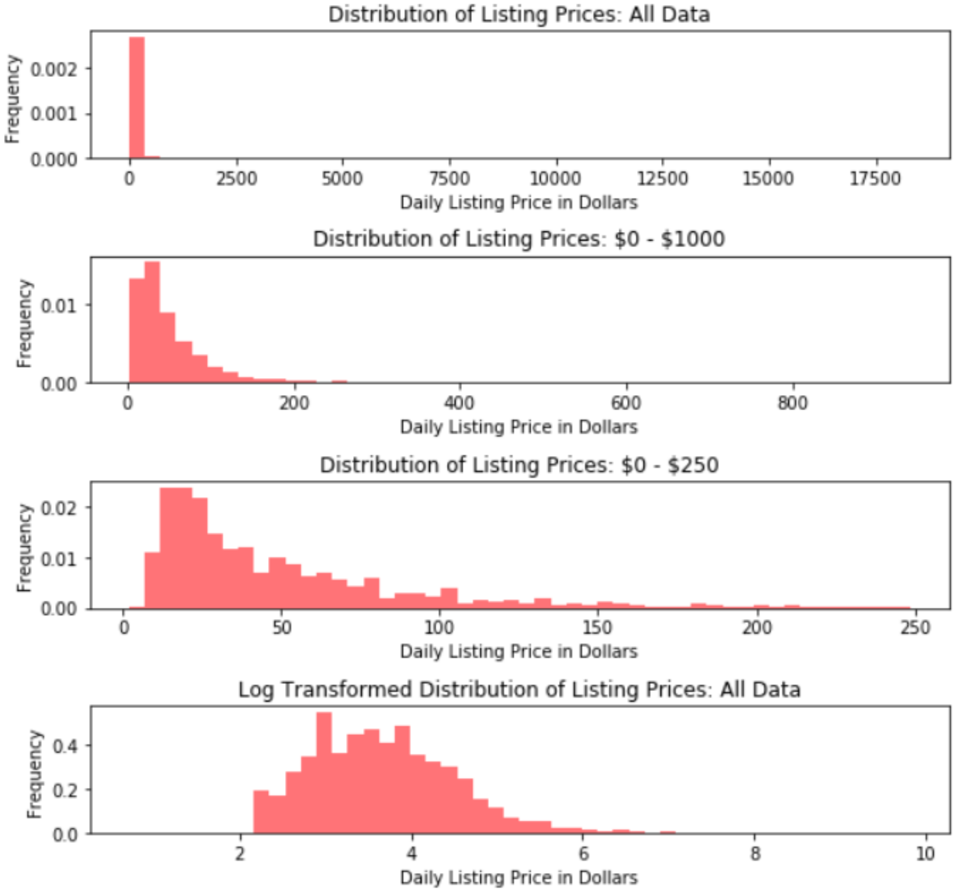
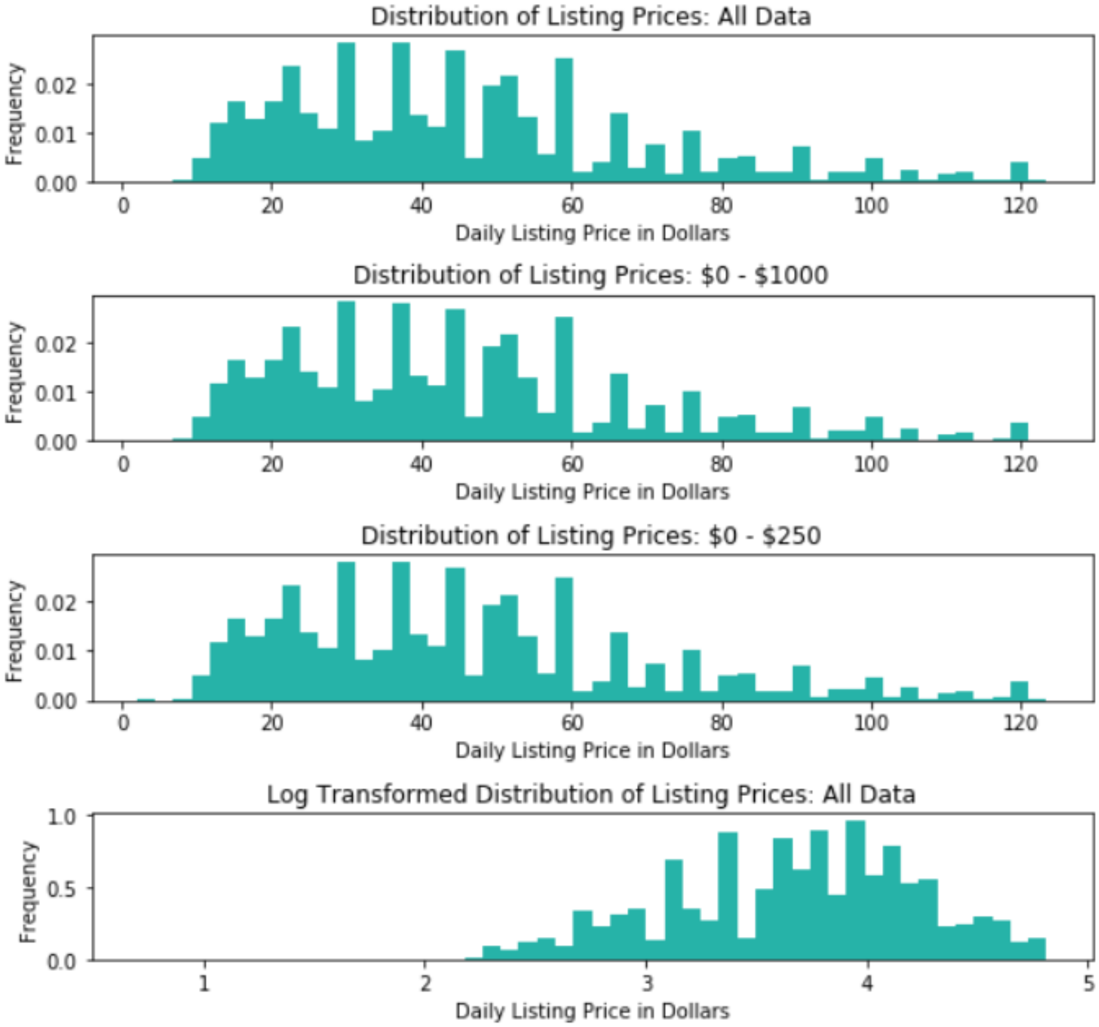


Figure A.2 Price Distribution in Santiago



Distribution of Listing Prices: All Data

Distribution of Listing Prices: $0 - $1000

Distribution of Listing Prices: $0 - $250

Log Transformed Distribution of Listing Prices: All Data

Figure A.3 Mexico City Room Type Distribution

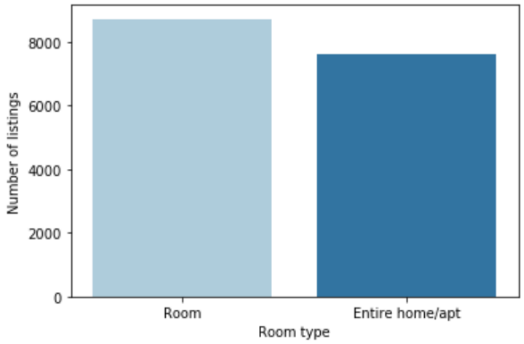
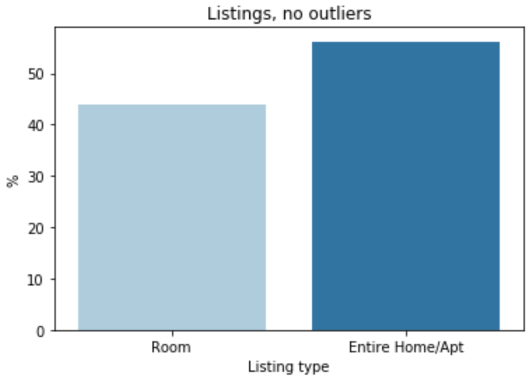


Figure A.4 Santiago Room Type Distribution



Listings, no outliers

Figure A.5 - Missing Values Santiago All Dataset

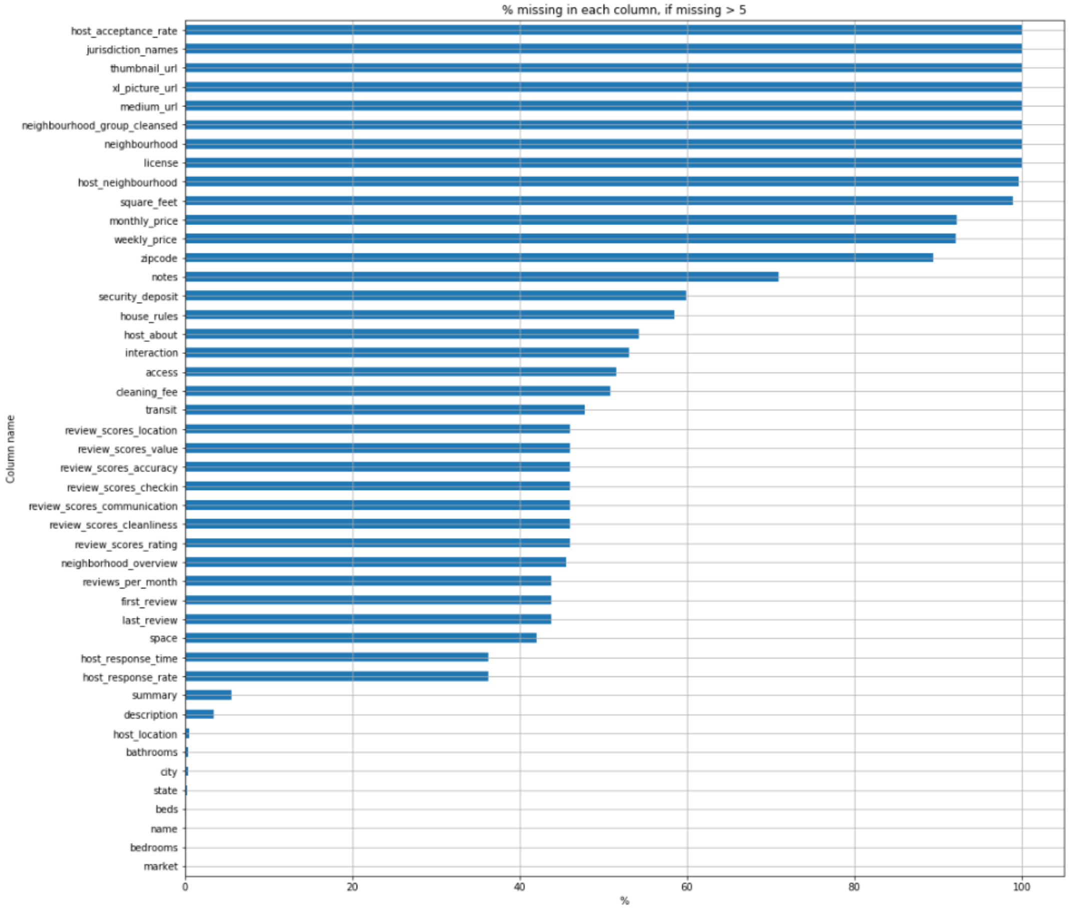


Figure A.6 Rating for Mexico City

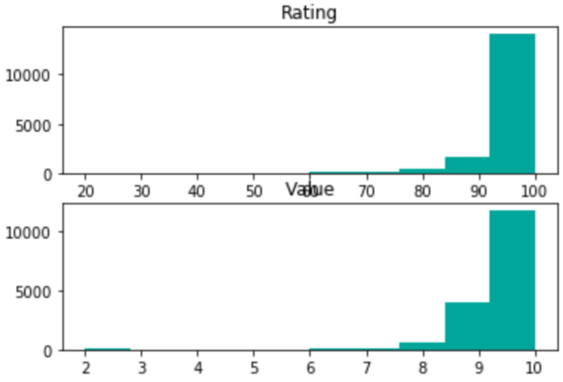


Figure A.7 Rating for Santiago

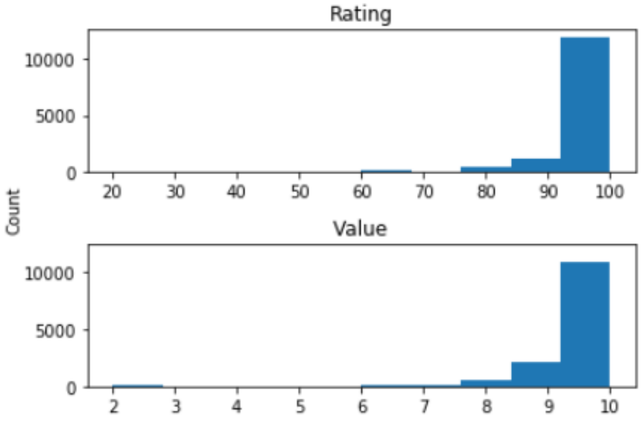


Figure A.8 Multi Listing for Mexico

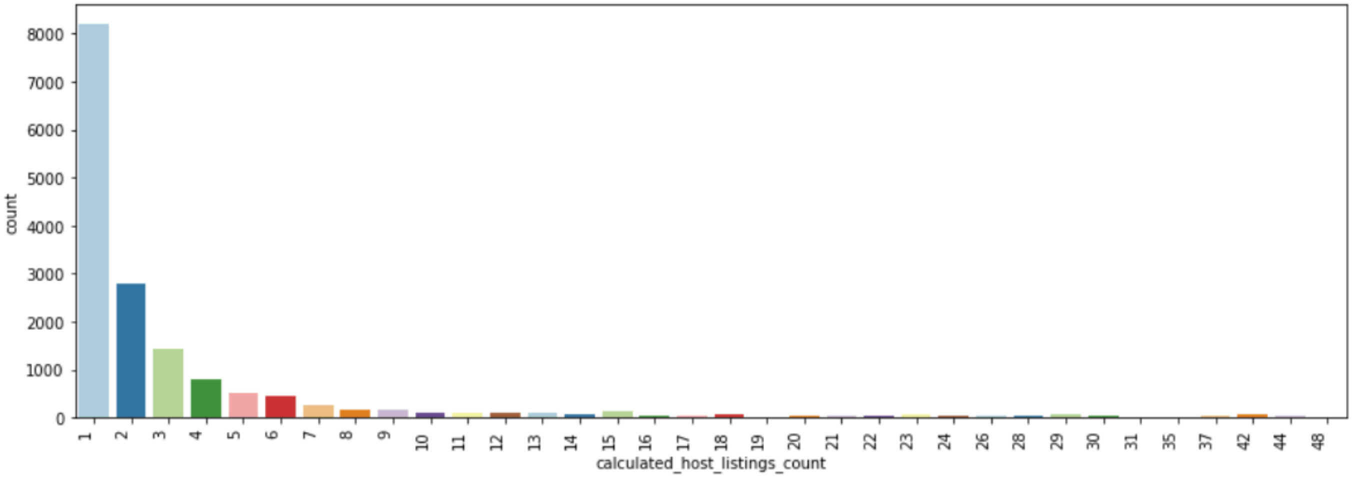


Figure A.9 Multi Listing for Santiago

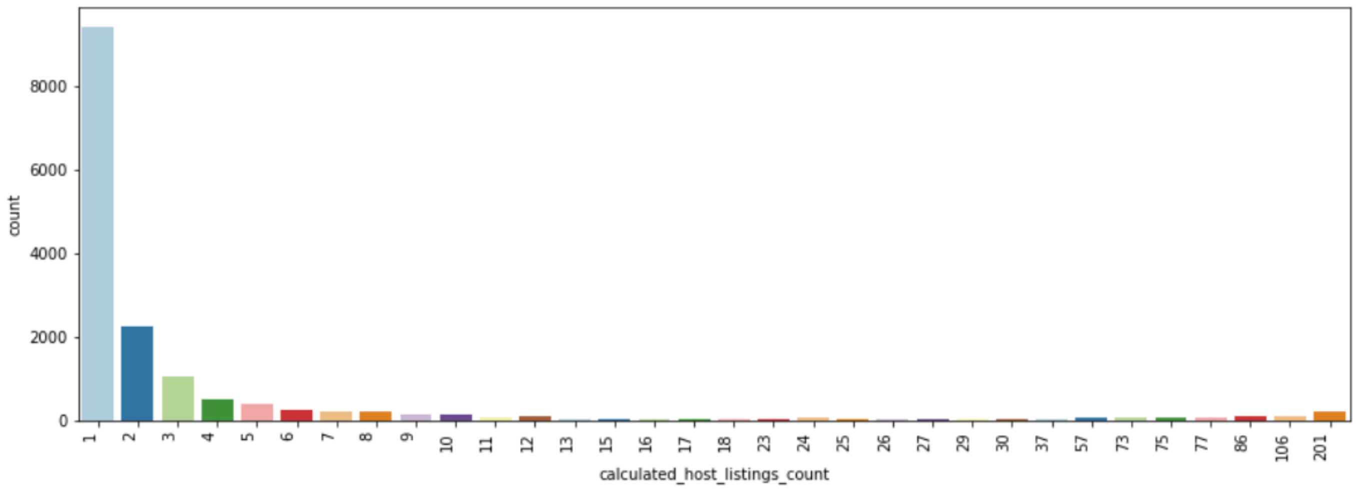


Figure A.10 Correlation Matrix for Mexico City

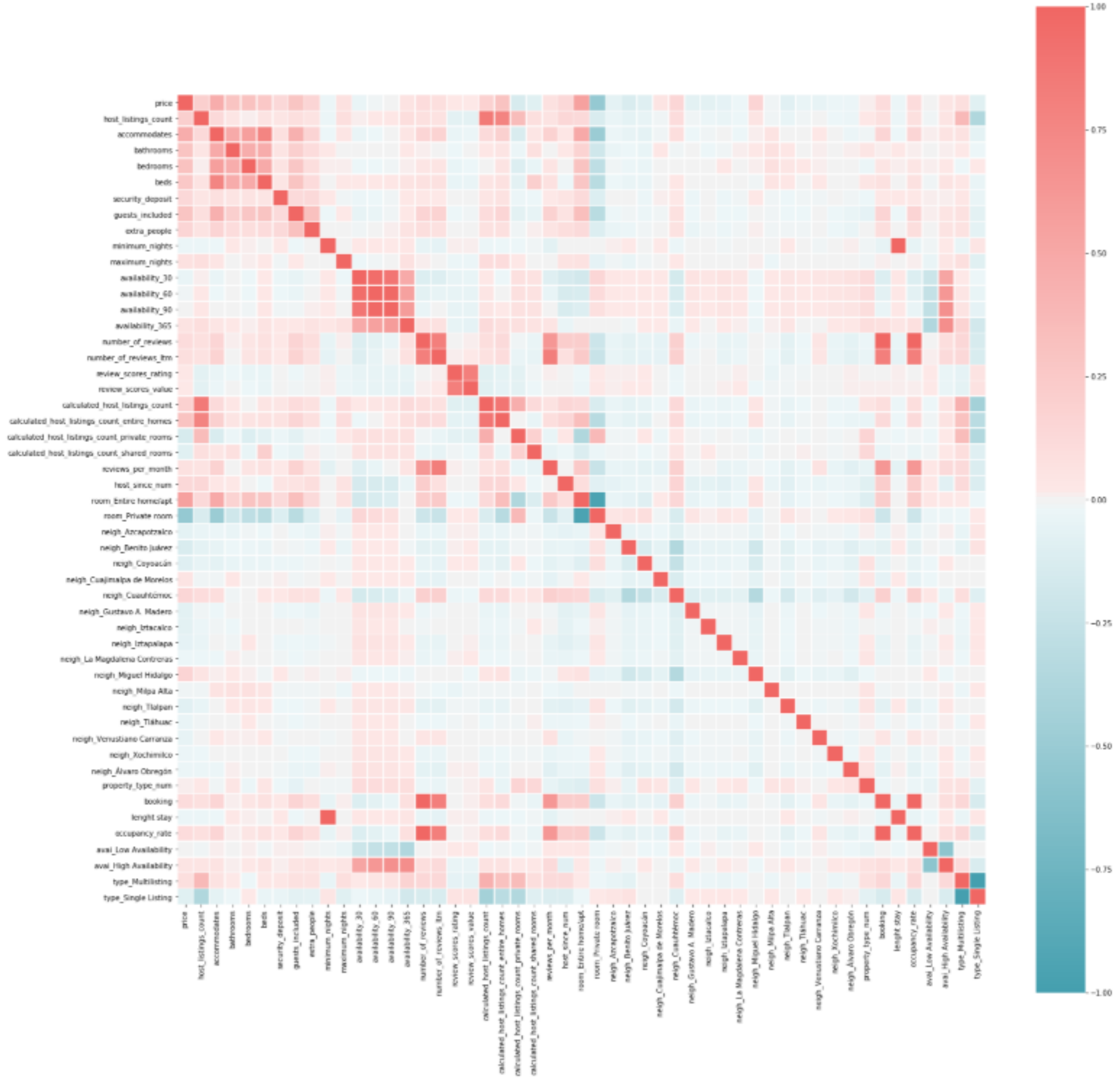


Figure A.11 Correlation Matrix Santiago

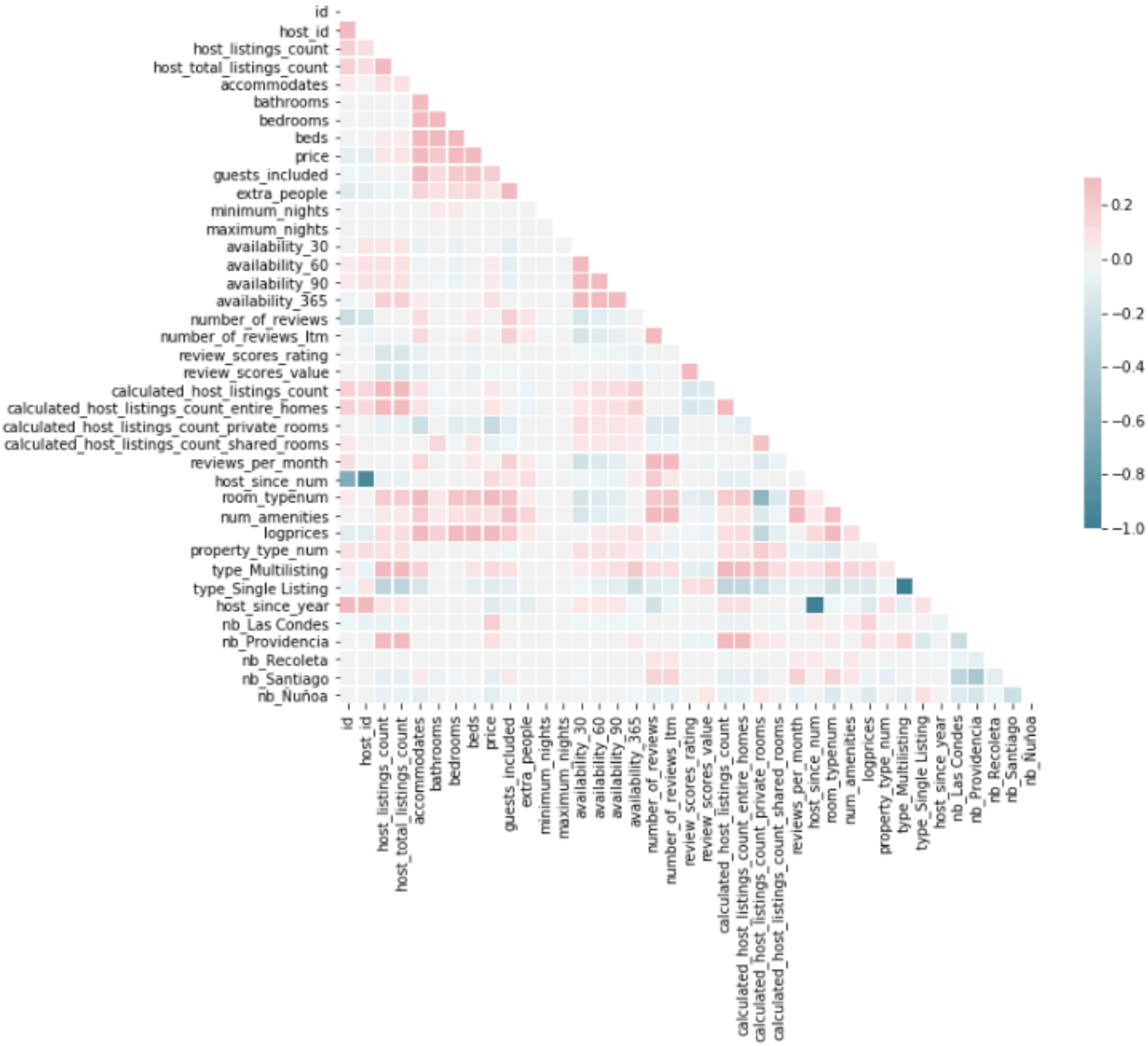
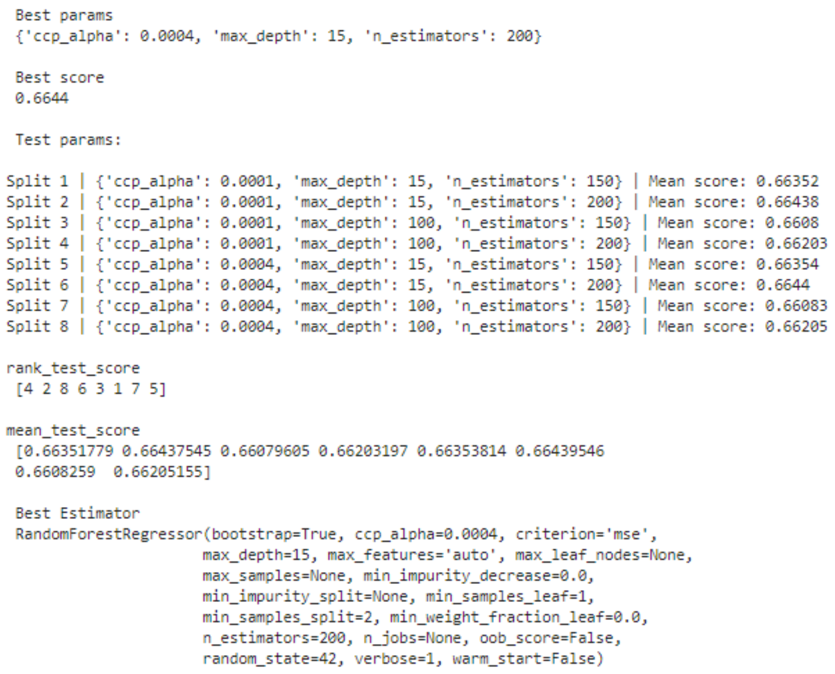


Figure A.12 (Santiago):



```
 Best params
 {'ccp_alpha': 0.0004, 'max_depth': 15, 'n_estimators': 200}

 Best score
 0.6644

 Test params:

Split 1 | {'ccp_alpha': 0.0001, 'max_depth': 15, 'n_estimators': 150} | Mean score: 0.66352
Split 2 | {'ccp_alpha': 0.0001, 'max_depth': 15, 'n_estimators': 200} | Mean score: 0.66438
Split 3 | {'ccp_alpha': 0.0001, 'max_depth': 100, 'n_estimators': 150} | Mean score: 0.6608
Split 4 | {'ccp_alpha': 0.0001, 'max_depth': 100, 'n_estimators': 200} | Mean score: 0.66203
Split 5 | {'ccp_alpha': 0.0004, 'max_depth': 15, 'n_estimators': 150} | Mean score: 0.66354
Split 6 | {'ccp_alpha': 0.0004, 'max_depth': 15, 'n_estimators': 200} | Mean score: 0.6644
Split 7 | {'ccp_alpha': 0.0004, 'max_depth': 100, 'n_estimators': 150} | Mean score: 0.66083
Split 8 | {'ccp_alpha': 0.0004, 'max_depth': 100, 'n_estimators': 200} | Mean score: 0.66205

rank_test_score
 [4 2 8 6 3 1 7 5]

mean_test_score
 [0.66351779 0.66437545 0.66079605 0.66203197 0.66353814 0.66439546
 0.6608259  0.66205155]

 Best Estimator
 RandomForestRegressor(bootstrap=True, ccp_alpha=0.0004, criterion='mse',
                       max_depth=15, max_features='auto', max_leaf_nodes=None,
                       max_samples=None, min_impurity_decrease=0.0,
                       min_impurity_split=None, min_samples_leaf=1,
                       min_samples_split=2, min_weight_fraction_leaf=0.0,
                       n_estimators=200, n_jobs=None, oob_score=False,
                       random_state=42, verbose=1, warm_start=False)
```

Figure A.13 (Mexico)

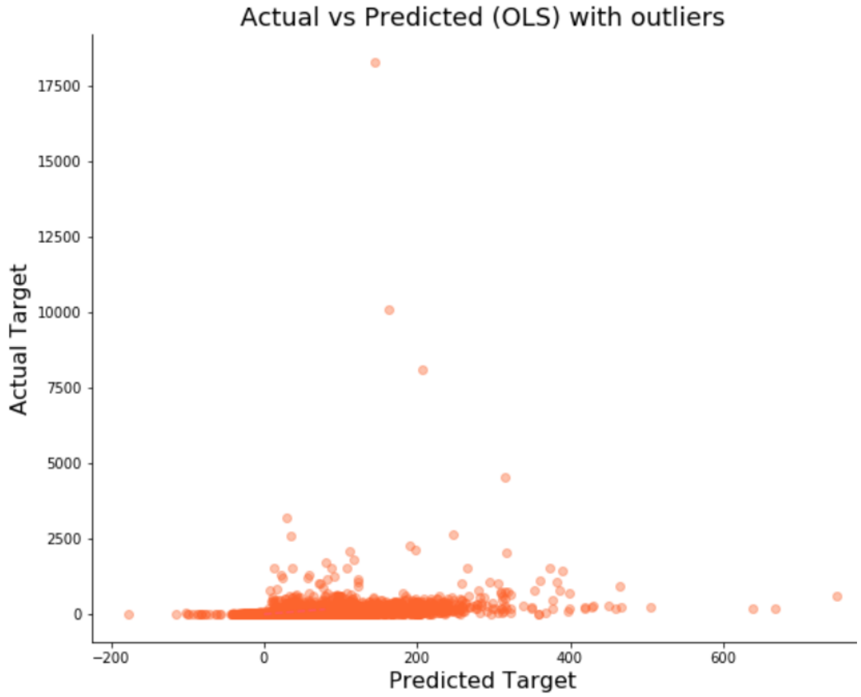


Figure A.14 (Mexico)

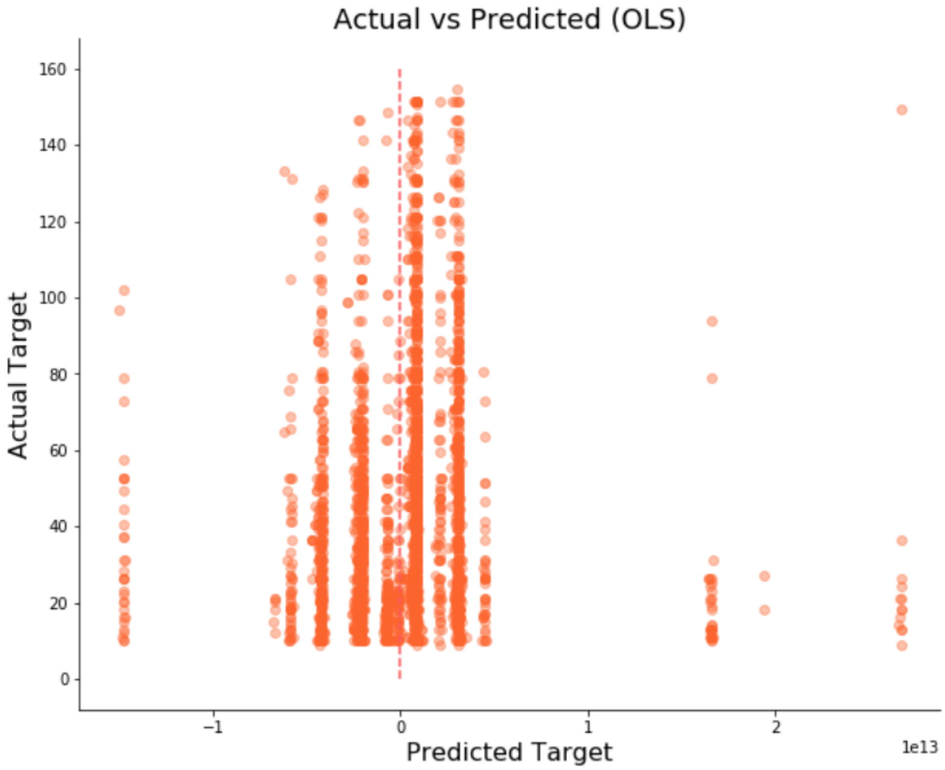


Figure A.15 (Mexico)

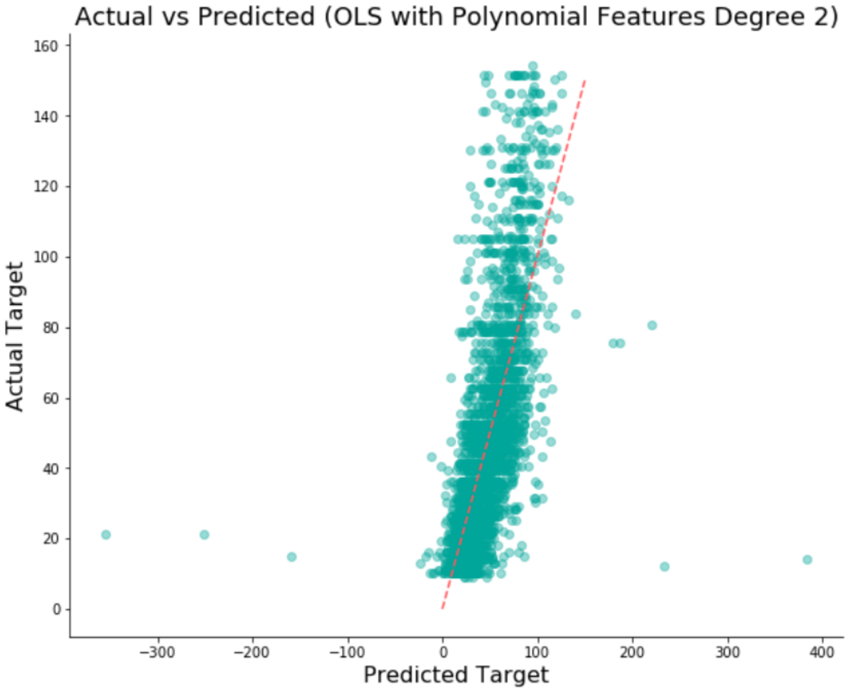
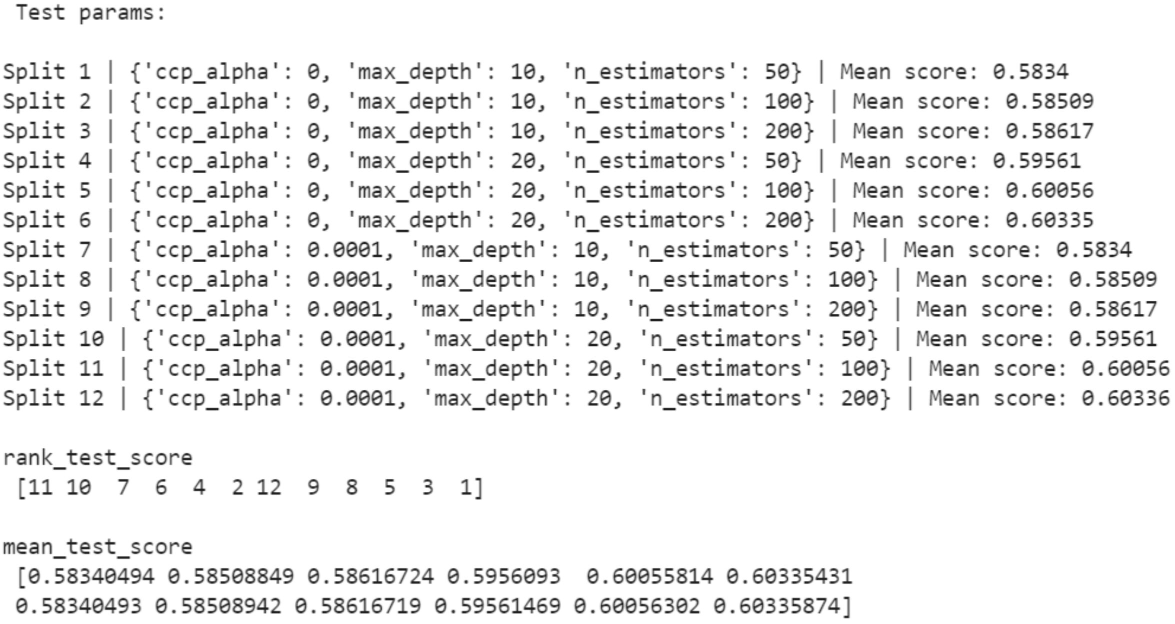


Figure A.16 (Mexico)



```
 Test params:

Split 1 | {'ccp_alpha': 0, 'max_depth': 10, 'n_estimators': 50} | Mean score: 0.5834
Split 2 | {'ccp_alpha': 0, 'max_depth': 10, 'n_estimators': 100} | Mean score: 0.58509
Split 3 | {'ccp_alpha': 0, 'max_depth': 10, 'n_estimators': 200} | Mean score: 0.58617
Split 4 | {'ccp_alpha': 0, 'max_depth': 20, 'n_estimators': 50} | Mean score: 0.59561
Split 5 | {'ccp_alpha': 0, 'max_depth': 20, 'n_estimators': 100} | Mean score: 0.60056
Split 6 | {'ccp_alpha': 0, 'max_depth': 20, 'n_estimators': 200} | Mean score: 0.60335
Split 7 | {'ccp_alpha': 0.0001, 'max_depth': 10, 'n_estimators': 50} | Mean score: 0.5834
Split 8 | {'ccp_alpha': 0.0001, 'max_depth': 10, 'n_estimators': 100} | Mean score: 0.58509
Split 9 | {'ccp_alpha': 0.0001, 'max_depth': 10, 'n_estimators': 200} | Mean score: 0.58617
Split 10 | {'ccp_alpha': 0.0001, 'max_depth': 20, 'n_estimators': 50} | Mean score: 0.59561
Split 11 | {'ccp_alpha': 0.0001, 'max_depth': 20, 'n_estimators': 100} | Mean score: 0.60056
Split 12 | {'ccp_alpha': 0.0001, 'max_depth': 20, 'n_estimators': 200} | Mean score: 0.60336

rank_test_score
 [11 10  7  6  4  2 12  9  8  5  3  1]

mean_test_score
 [0.58340494 0.58508849 0.58616724 0.5956093  0.60055814 0.60335431
 0.58340493 0.58508942 0.58616719 0.59561469 0.60056302 0.60335874]
```

Figure A.17 (Santiago)

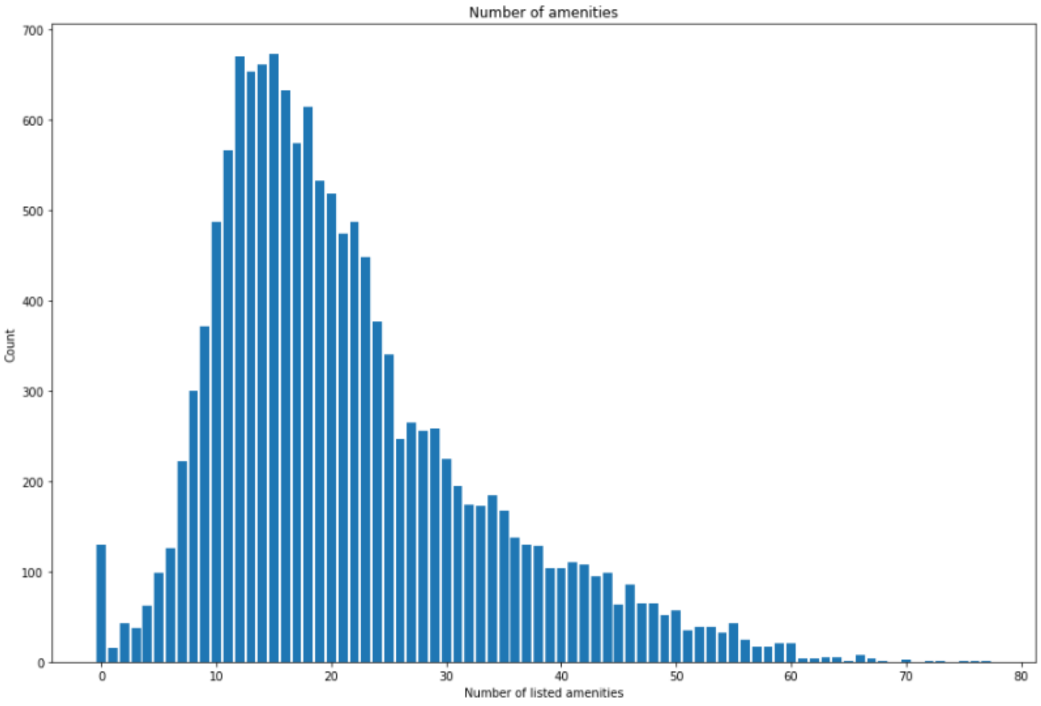


Figure A.18 (Santiago)

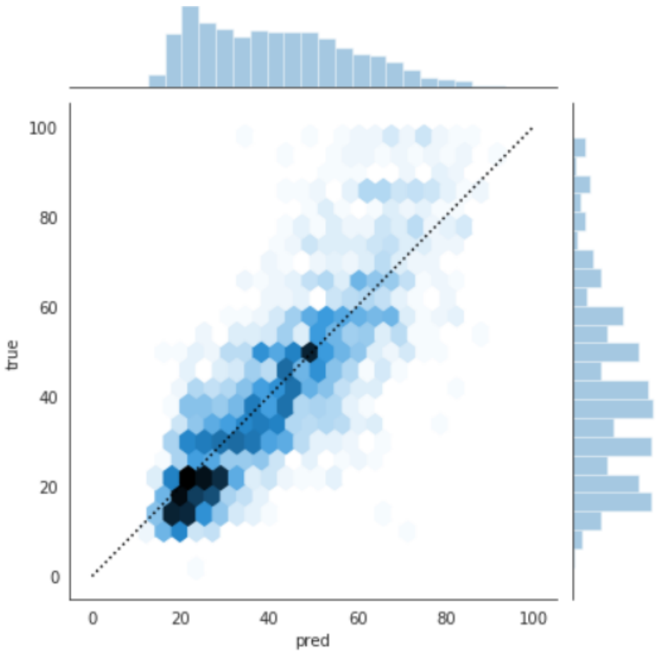
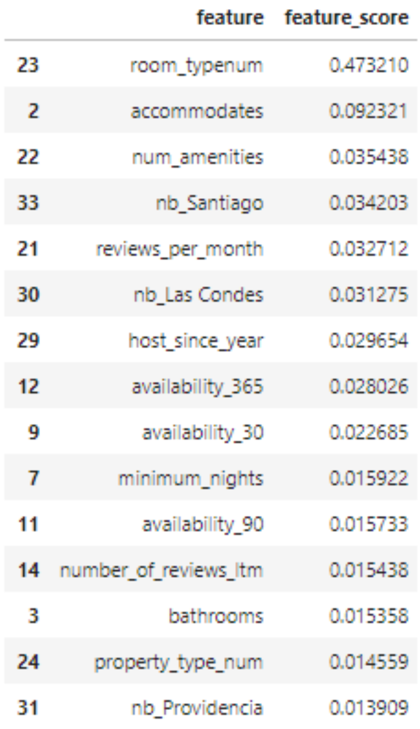
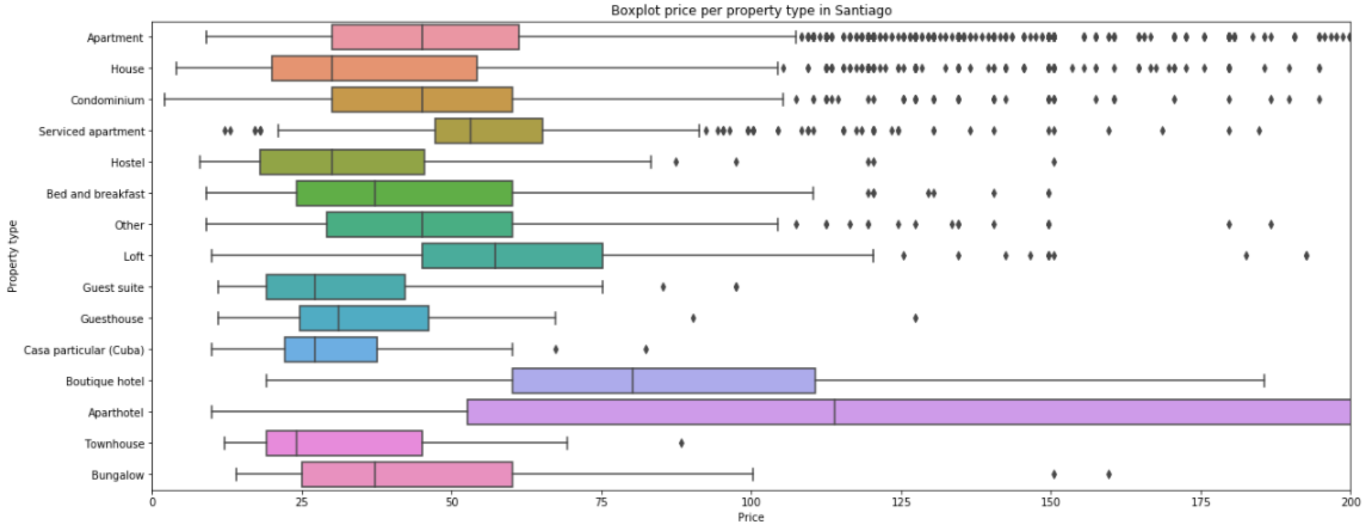


Figure A.19 (Santiago)
This table shows the ranking of the features score for Santiago:



	feature	feature_score
23	room_typenum	0.473210
2	accommodates	0.092321
22	num_amenities	0.035438
33	nb_Santiago	0.034203
21	reviews_per_month	0.032712
30	nb_Las Condes	0.031275
29	host_since_year	0.029654
12	availability_365	0.028026
9	availability_30	0.022685
7	minimum_nights	0.015922
11	availability_90	0.015733
14	number_of_reviews_ltm	0.015438
3	bathrooms	0.015358
24	property_type_num	0.014559
31	nb_Providencia	0.013909

Figure A.19 (Santiago)
This boxplot showcase the prices of the most occuring property types


Boxplot price per property type in Santiago
Property type
Apartment
House
Condominium
Serviced apartment
Hostel
Bed and breakfast
Other
Loft
Guest suite
Guesthouse
Casa particular (Cuba)
Boutique hotel
Aparthotel
Townhouse
Bungalow
0
25
50
75
100
125
150
175
200
Price