Munich's Airbnb Data Analysis

by Selmir Kalender
17. Aug. 2020

1. Introduction

1.1 Background

Airbnb is a internet marketplace for short-term home and apartment rentals. It allows you to, for example, rent out your home for a week while you're away, or rent out your empty bedroom.
One challenge that Airbnb hosts face is determining the optimal rent price. In many areas, renters are presented with a good selection of listings and can filter by criteria like price, number of bedrooms, room type, and more. Since Airbnb is a market, the amount a host can charge is ultimately tied to market prices.

1.2 Problem

Although Airbnb provides hosts with general guidance, there are no "easy to access" methods to determine the best price to rent out a space. There is third-party software available, but for a hefty price (see an example on available software, click here).
One method could be to find a few listings that are similar to the place that will be up for rent, average the listed prices and set our price to this calculated average price. However, with the market being so dynamic, we would probably be looking to update the price regularly and this method can become tedious.
Moreover, this may not be very accurate, as we are not taking into account other important factors that may give us a comparative advantage over other listings around us. This could be property characteristics such as number of rooms, bathrooms and extra services on offer.
The aim of this project is to propose a data-driven solution, by using machine learning to predict rental price.
For this project, a predictor based on space will be introduced to the model: the property's proximity to certain venues. This will allow the model to put an implicit price on things such as living close to a bar or a supermarket.

1.3 Interest

Who will be beneficial from this project:

- **Hosts** who are interested to identify the market value for their apartments or rooms
- **Tourists** who would want to compare different hosts to find the best solution for themselves
- **Data Analysts/Scientists** who can view the analysis for this project and can get inspired to implement in other different problems using the techniques and ideas used here.

2. Data Overview

Airbnb does not release any data on the listings in its marketplace, a but separate group named Inside Airbnb scrapes and compiles publicly available information about many cities Airbnb's listings from the Airbnb web-site. For this project, their data set scraped on June 20, 2020, on the city of Munich, Germany, is used. It contains information on all Munich Airbnb listings that were live on the site on that date (almost 10,000). Here's a direct link.
The data has certain limitations. The most noticeable one is that it scrapes the advertised price rather than the actual price paid by previous customers. More accurate data is available for a fee in sites like AirDNA.

Each row in the dataset is a listing available for rental in Airbnb's site for the specific city (observations). The columns describe different characteristics of each listing (features).

Some of the more important features this project will look into are the following:

- accommodates: the number of guests the rental can accommodate
- bedrooms: number of bedrooms included in the rental
- bathrooms: number of bathrooms included in the rental
- beds: number of beds included in the rental
- price: nightly price for the rental
- minimum_nights: minimum number of nights a guest can stay for the rental
- maximum_nights: maximum number of nights a guest can stay for the rental
- number_of_reviews: number of reviews that previous guests have left

To model the spatial relationship between Airbnb rental prices and property proximity to certain venues, we use the Foursquare API to access the city's venues and the street network, available through OpenStreepMap (OSM).

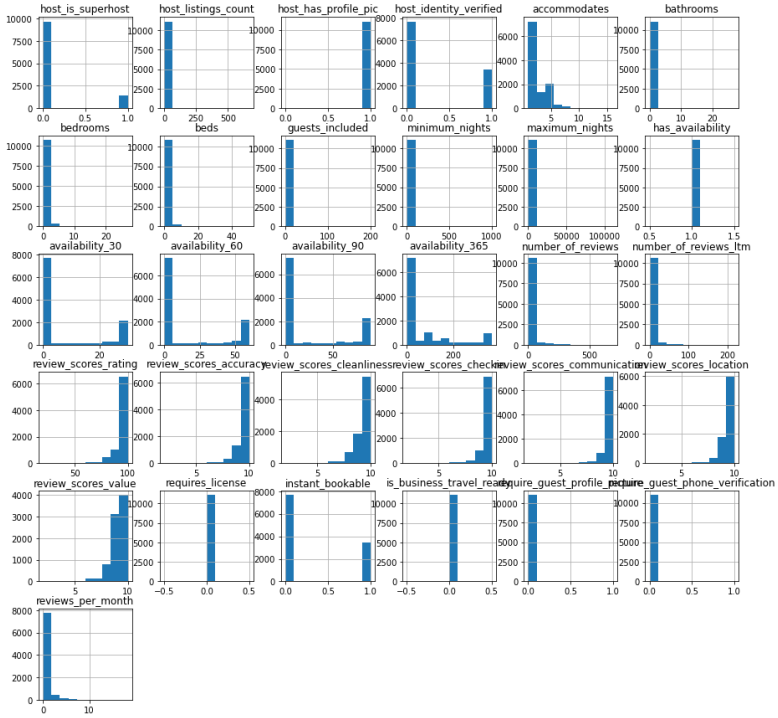
3. Methodology

3.1 Cleaning and pre-processing

The original dataset contained 11172 Airbnb listings and 106 features. Natural Language processing was not used in for the model. Therefore, free text columns were dropped as well as other features not useful for predicting price (e.g. url, host name and other host-related features that are unrelated to the property). Multiple columns for property location were found. Due to all of the listings being located in Munich, the country and city columns were dropped. There were multiple columns for minimum and maximum night stays, but the two main ones were used as there were few differences between them (e.g. *minimum_nights* and *minimum_minimum_nights*).

A visual inspection of whether boolean and categorical features contained sufficient numbers of instances in each category to make them worth including led to the realisation that several columns only contained one category and could be dropped. There were *has_availability*, *host_has_profile_pic*, *is_business_travel_ready*, *require_guest_phone_verification*, *require_guest_profile_picture,* and *requires_license*.

Visual inspection of Boolean and categorical features:



The *experiences offered* feature was also dropped, due do most listings not offering it. About two thirds of rows did not have a value for *host_response_time*, and a lot of these have also had not yet been reviewed. Therefore this section of the data set consisted primarily of properties which had not yet had a completed stay (most likely properties which had not yet had a booking, although may also include properties that had a booking currently occurring). Although this is a considerable proportion of the dataset, these listings were retained in the data because they were still legitimate properties with advertised prices and part of the comparative market when considering the price for which to advertise an Airbnb listing. However, if the dataset being used had the actual average price paid as its target, it would be necessary to drop these rows because they would not have a value, as they had not yet been booked.
Because *host_response_time* is unknown for so many listings, it was being retained as its own category, '*unknown*'.
The same applied to *host_response_rate*, with about two thirds of values being null. The feature was kept as its own category, after grouping other values into meaningful groups (i.e. transforming this into a categorical feature, rather than a numerical one). Because about 70% of hosts responded 100% of the time, this was kept as its own category, and other values were grouped into bins.
There were 7 rows lacking values for each of five different host-related features, so those rows were dropped.
Some cleaning of property types features was required as there were a large number of categories with only a few listings. The categories *Apartment*, *House* and *Other* were used, as most properties could be classified as either *apartment* or *house*.
For *bathroom*, *bedroom* and *beds* missing values were replaced with the median. Most listings have the same bed type, so it was considered that it would not constitute a comparative advantage to have this feature. Hence, it was dropped.
The *amenities* feature was a list of additional features in the property, i.e. whether it has a *TV* or *parking*. It is understood that some amenities are more important than others (i.e. a *balcony* is more likely to increase price than a *fax machine*), and some are likely to be fairly uncommon (i.e.. *'Electric profiling bed'*). For the purpose of this project, *amenities* were extracted based on quick research into which *amenities* were considered by guests as important, as well as personal experience. This was further investigated in the EDA section. One way to reduce the number of features was to remove the *amenities* that added relatively little information, or were relatively unhelpful in differentiating between different listings. *Amenity* features where either the true or the false category contained fewer than 10% of listings were removed.
There were 89 unique categories for *calendar_updated*, and it was not entirely clear what this feature was adding to the model (a host might update their calendar for multiple different reasons). Therefore this column was dropped.
There were also multiple different measures of *availability*, which would have resulted in a multicollinearity problem. Thus, only one feature was retained, *availability for 90 days* (*availability_90*).
Almost 25% of listings had not had a review written for them. This was too large a proportion of the dataset to drop, and dropping the columns would have resulted in the loss of a lot of useful information because reviews are very important in people's decisions to book, and therefore price.
This was also too large a proportion of the dataset to simply replace with median/mean values, as this would have skewed the distribution substantially. Also, the missing values here were not really missing values, as the fact that were *NaNs* was meaningful (it tells us that these are new or previously unbooked listings that have not had reviews yet). In order to make the resulting model able to predict prices for any Airbnb listing, including brand new listings, is actually beneficial to keep them in. Therefore, these were kept as an unknown category, and the feature was treated as categorical (and therefore one-hot encoded) rather than numerical.
As above, listings without reviews were kept and replaced with unknown. Other ratings were grouped into bins. The majority of ratings were 9 or 10 out of 10. Therefore for these columns, 9/10 and 10/10 were kept as separate groups, and 1-8/10 were binned together (as this is, by Airbnb standards, a 'low' rating).

The *cancellation policy* feature was categorised into three larger categories: *super-strict*, *strict* and *moderate* (e.g. the super strict options were only available to long-term Airbnb hosts, and is invitation only).
The features *number_of_reviews*_ltm and *reviews_per_month* were deemed to be likely highly correlated with *number_of_reviews* and so were dropped.
Numerical features *price*, *security deposit*, *cleaning fee* and *extra-people* were converted to integers (e.g. price was a string, as it contained the currency sign). Having a missing value for *security deposit* was considered to be functionally the same as having a *security deposit* of €0, so missing values were replaced with 0. As with security deposit, having a missing value for *cleaning fee* and *extra people* was also replaced with 0.

3.2 Exploratory Data Analysis (EDA)

3.2.1 Time Series

Time is an important factor to consider in a model when we would like to predict prices or trends. A time series is a series of data points ordered in time. In a time series, time is often the independent variable and the goal is usually to make a forecast for the future. There are also other questions that come to play when dealing with time series. For example: Is there any seasonality to the price? Is it stationary? Even though we may not be able to include this aspect into our model, it is good to explore it to be aware of it and be able to make recommendations for future research. Thus, in this section, we will explore this aspect of the data.
For Airbnb prices, a high level of seasonality is expected due to the characteristics of the market. In October, during the *Oktoberfest*, room rental price rises considerably. It is an extremely popular event and much of the rented property available will have been taken up due to the number of people who attend each year. Munich has lots of other events throughout the year, mainly focused between March and December.

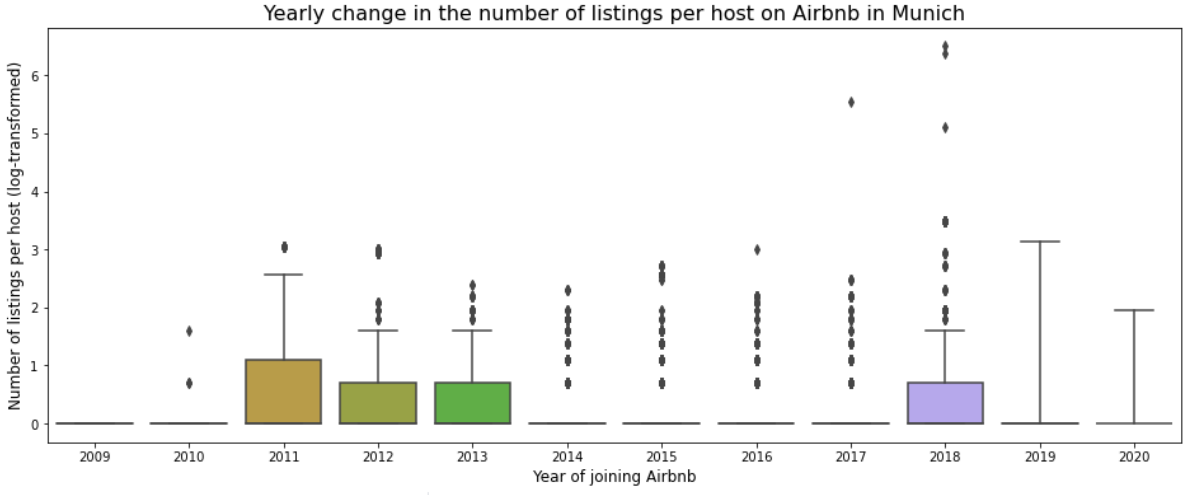
Munich's hosts joining Airbnb each month and receiving their first review:

Seasonality and Trend of Airbnb each month:

We can see above that there is a clear seasonality. Every year, you see a peak towards hosts joining around the middle of the second half of the year (summer/fall), and the lowest points are the beginning and the end of each year. There is a big peak in the number of hosts joining Airbnb between 2014-2015 and 2015-2016. Indeed, there has been a fast growth of Airbnb since middle 2014, with clear peaks during Munich's *Oktoberfest*. This was around the time when Airbnb became increasingly popular for short-term lease, as a way to get around local legislation and taxation.

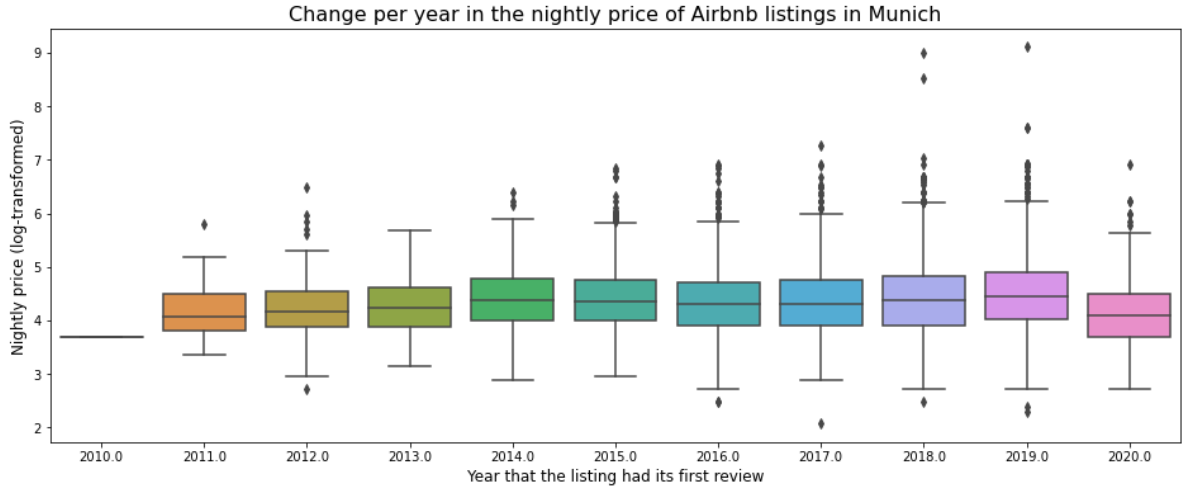
Another important pattern to observe is the *number of listings per host*. There are a number of professional Airbnb management companies which host a large number of listings under a single host profile. However, there is no consistent upwards trend in the average number of properties managed by each host.

Change per year in number of listings per host in Airbnb Munich:



In term of changes in prices over time, the *average price per night* for Airbnb listings in Munich has increased slightly over the last 10 years. In particular, the top end of property prices has increased, resulting in a larger increase in the mean price compared to the median. The mean price in 2011 was €76.92 and the median €59.0, whereas the mean price in 2019 (the last complete year of data) was €120.39 and the median €79.0.

Change per year in nightly price of Airbnb listings in Munich:



3.2.2 Numerical features

Looking at price distribution, advertised prices range from 8 to €12.000. The extreme ends of the range are due to hosts not understanding how to use Airbnb advertised prices correctly. The advertised prices can be set to any arbitrary amount, and these are the prices that show when dates are not entered on the site. Once you enter the dates you want to occupy the property, prices can vary a lot.

Unfortunately, this model will predict advertised prices rather than the actually paid prices. Nevertheless, cleaning of the particularly unhelpful values will be done. Values under €10 will be increased to €10.

There are notable drop-offs in prices at €200, €400-500 (second graph, green line) and €900-1.000. Values above €1.000 will be reduced to €1.000.

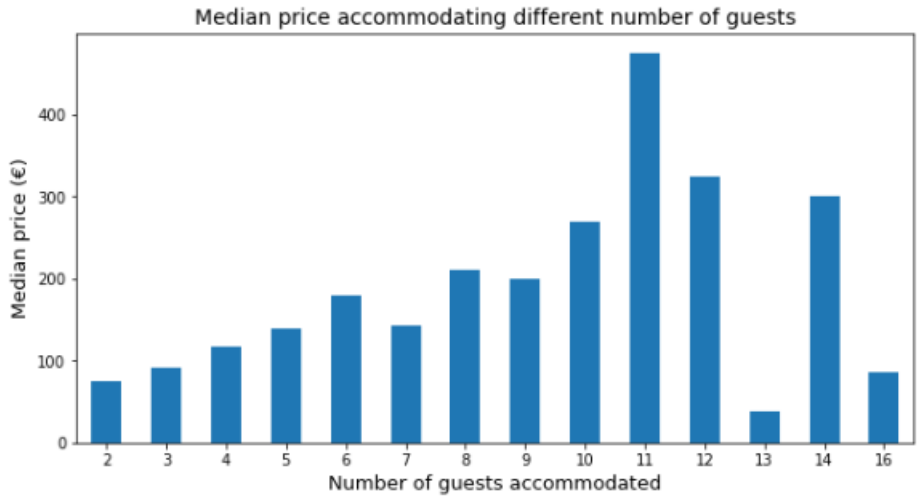
Advertised price distribution:

The median number of listings that the host of each listing has is 1. This means that on average (median) each listing is hosted by a host who only manages that listing. The mean is slightly higher (2) due to some hosts managing larger numbers of listings, as discussed above in the Time Series section. For example, the host with the highest number of listings has 666 listings under its ID. About half of listings are from hosts with one listing, and half are from multi-listing hosts.

Two difficulties in discovering how many listings hosts have on average are:
*this number is only known on the level of the listing, so hosts with more listings are represented more frequently (e.g a host with 10 listings may be represented up to 10 times in the dataset)

*a host's other listings may not be in Munich, so some multi-listing hosts may appear multiple times in the dataset, and others may appear only once.

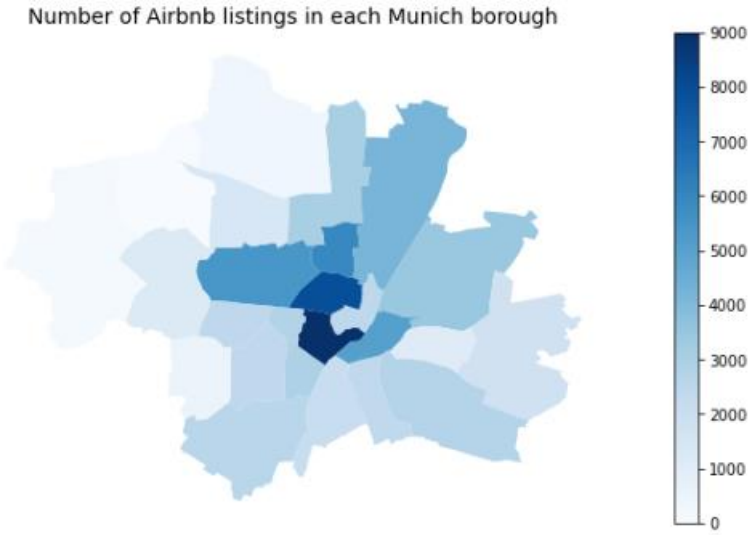
Median Price per number of guest accommodated:



3.2.3 Categorical features

Much of Airbnb listings are centred around Munich's Centre (Altstadt, Maxvorstadt), which is consistent with huge draw for tourists throughout the year.

Airbnb listings per borough:



About 90% of properties are apartments/flats. The remainder are houses or more uncommon property types (e.g. bed and breakfast).

About 55% of listings are entire homes (i.e. you are renting the entire property on your own). Most of the remainder are private rooms (i.e. you are renting a bedroom and possibly also a bathroom, but there will be other people in the property). Fewer than 1% are shared rooms (i.e. you are sharing a room with either the property owner or other guests).

For every review category, the majority of listings that have had a review have received a 10/10 rating for that category (or 95-100/100 overall). Ratings of 8 or below are rare. Guests seem to be most positive about communication, check-ins and accuracy. As noted previously, over a quarter of listings have not yet been reviewed.

The most common time period in which current Airbnb listings had their first review is 2-3 years. This means that a lot of listings on the site have been active for at least a couple of years. However, relatively few have been active for more than four years.

The most common category for the time since a listing received its last review is 1+ years. This means that a lot of listings have not been reviewed recently. The majority of these are probably what are sometimes referred to 'inactive' listings, because although they are technically live on the site, they do not have their calendards open and are not available to book.

3.2.4 Venue Proximity

We were looking to explore proximity to certain venues as a possible price predictor. Walkability and ability to reach places could be a deal-maker or breaker when it comes to choosing an accomodation. Proximity to certain venues, such as touristic attractions, restaurants, cafes and even shops could help us predict price. For this, we have used the Foursquare API to explore the venues per neighbourhood. As discussed before, Maxvorstadt and Altstadt are the areas which concentrates the majority of Airbnb listings.
We have used Foursquare API to explore the venues around the listings, using the latitude and longitude of each neighbourhood. We then found out which venues are the most common and selected the most common venues as points of interest (POIs) for our accessibility analysis.
As we have found out, the most common Venues are *Bakeries*, *Supermarkets* and *Bus Stops* followed by *Cafés* and different *Restaurants* and *Gardens*.
Different restaurant venues were in subcategories which made them less common than if they were aggregated. Thus, for the purpose of accounting for the venues that may have the most impact on price, the venues were limited to the most common categories.

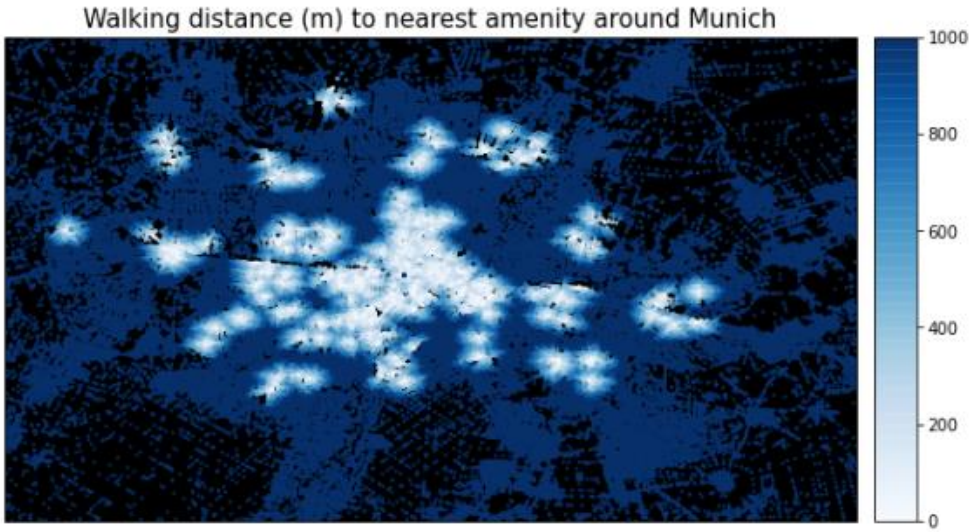
3.2.5 Walkability to nearest Venues

The information on venues was used to select Points of Interest (POIs) for which graphs and accessibility measures were created. For this purpose, *OSMnx* and *Pandana* libraries were used. *Pandana* is a handy graph library that allows for Pandas data frames to be passed through into a network graph that maps graph-level analyses to underlying C operations. In certain situations, such as the performance of accessibility analyses, this makes in-memory performance and iterative development based on this library possible, as opposed to what would be a cumbersome development process with tools that fail to leverage the same degree of C-level operations utilisation.
The street network data with 181,930 nodes was downloaded from the OSM API. Restaurant, Plaza, Café, Beer Garden and Garden are the POIs. Restaurants were aggregated, as they were divided in sub-categories in the POIs dataset. Thus, the POIs dataset consists of 347 restaurant venues, 87 cafes, 34 plazas, 8 beer garden and 3 garden.

3.2.6 Accessibility from each node to amenities

This is where conducting geographical analysis in Python shows its power, as opposed to sitting in front of a *GIS* package. Pandana is built for speed. First, we've passed it a maximum search distance. This allowed a key step that speed up future enquiries: Pandana has build a condensed representation of the network (implemented in C++), allowing rapid calculations within a defined radius of each node. We've then, build a table of distances to the nearest 5 points of interest from a couple of intersections. When this was finished, accessibility analyses for different selected amenities were done in under a second. Two algorithms made this possible: contraction hierarchies and kd-trees.

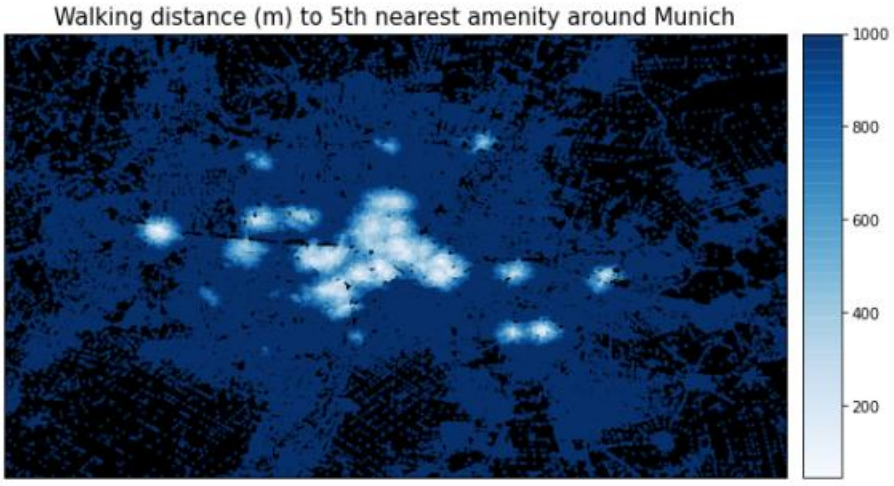
Walking distance to nearest amenity:



As seen above, for most of the zones people have to walk more than 500 meters to reach the nearest amenity, whereas Munich's *Centre* has walking distances of less than 100 meters on average. The map shows the walking distance in meters from each network node to the nearest Café, Plaza, Beer Garden, Restaurant and Garden.

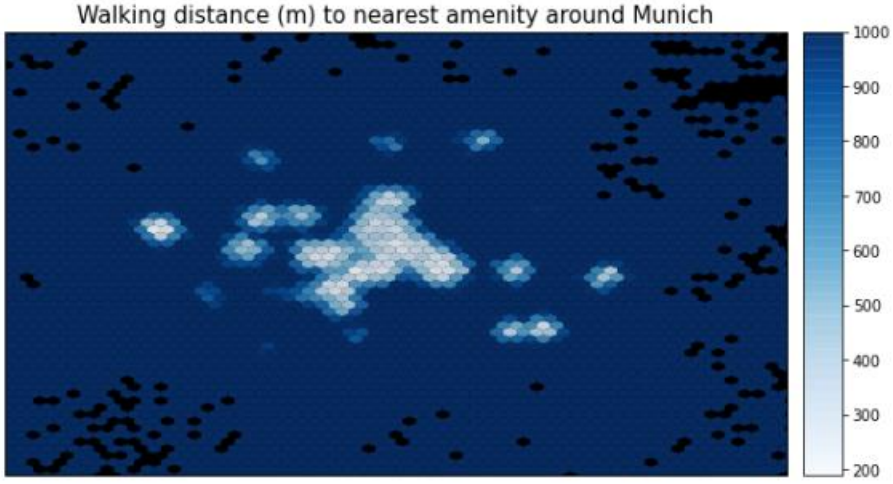
The map shows the walking distance in meters from each network node to the nearest Café, Plaza, Beer Garden, Restaurant and Garden. However, a better indicator of accessibility might be having access to a large number of amenities. So instead of the nearest, accessibility to the fifth-nearest amenity was plotted.

Walking distance to the 5^{th} nearest amenity:



This time, it’s even more noticeable that the city centre is more accessible.
Accessibility scores can quickly be constructed to answer a given question: whether it's access to essential services, or walkable neighborhoods that appeal to young workers. For the purpose of this analysis, access to restaurants, gardens, plazas, beer gardens are taken as essential for Airbnb users. Hence, all amenities were weighted equally and distance to the fifth nearest amenity was used as a compound measure of accessibility. This gives a clearer picture of which neighborhoods are most walkable, compared with plotting just the distance to the single nearest venue/amenity.

Walking distance to 5th nearest amenity in hexabins heatmap:



4. Modeling

4.1 Preparing the data

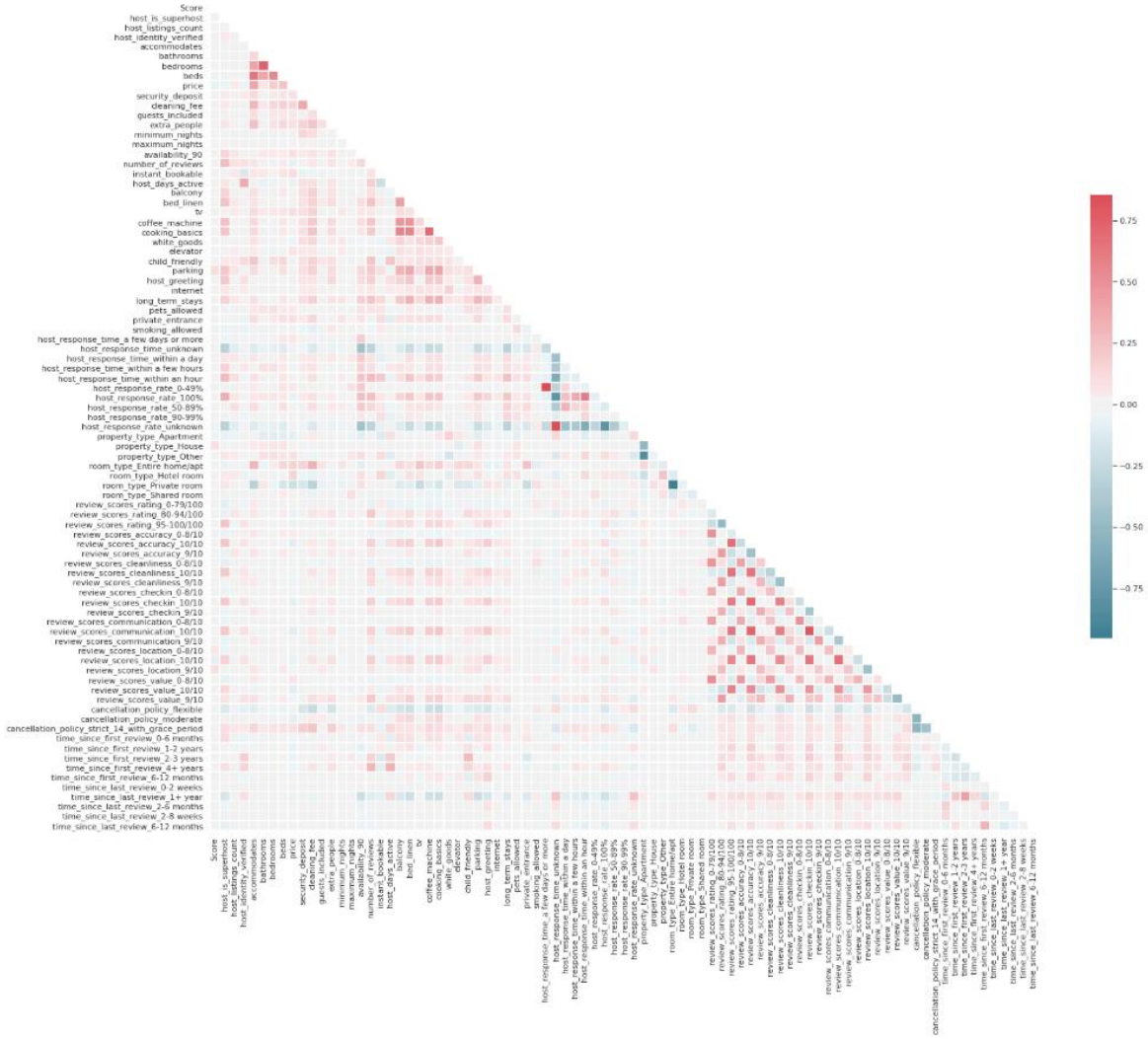
To assess multicollinearity, categorical variables were hot-encoded. The accessibility score (distance to the 5th nearest venue) was assigned to each listing, based on which neighbourhood they belong to. Thus, the geographical data and venue data columns were dropped, as they were no longer needed.

First multicollinearity plot:

:

On the above first analysis, it did not look like there were any significant collinear relationships with neighbourhood variables, so these were temporarily dropped to produce a clearer heatmap for the remaining features.

Second multicollinearity plot:

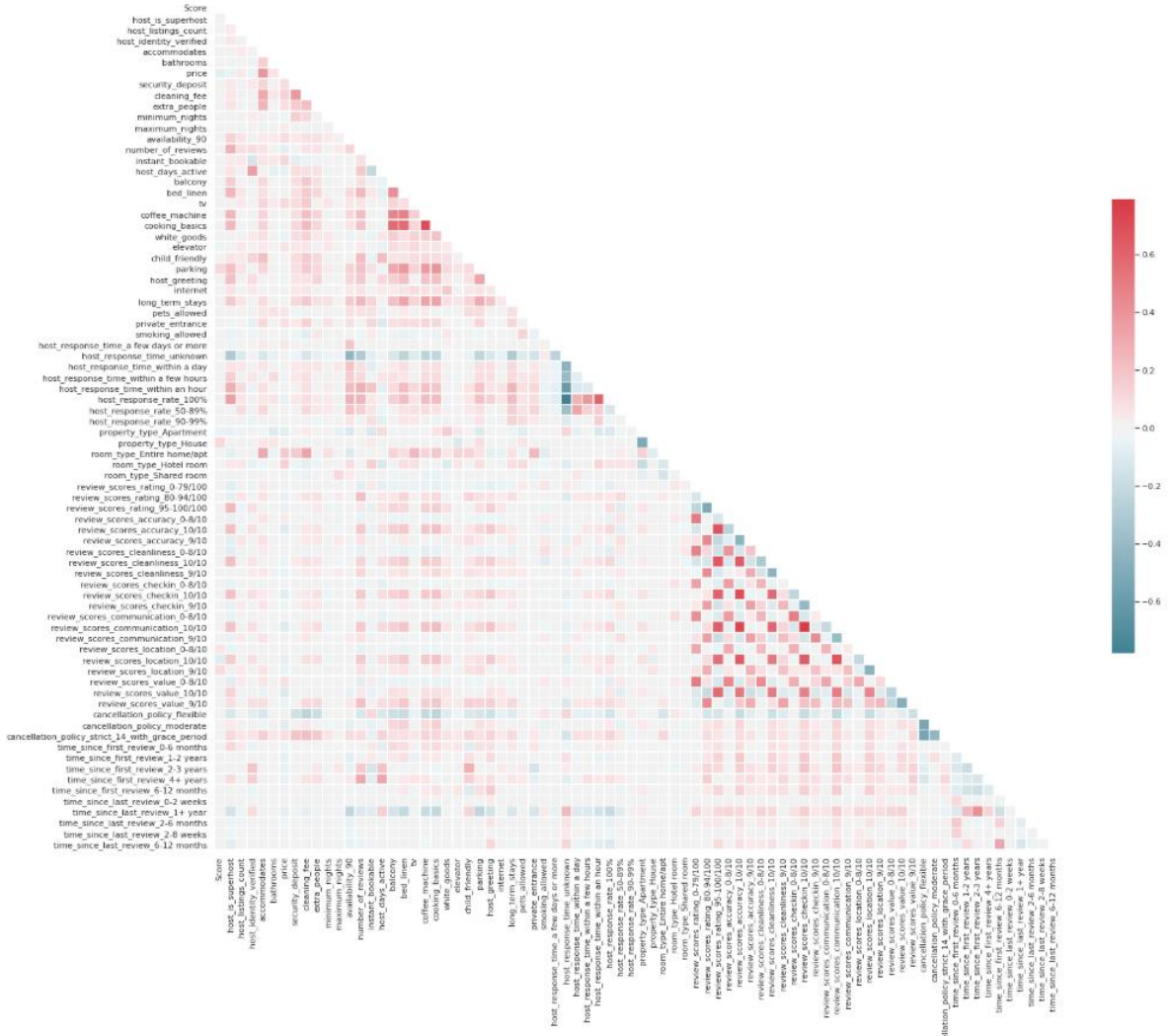


From the second analysis, certain areas of multi-collinearity were observed:

- *Beds*, *bedrooms*, *guests_included* and the *number of people* that a property accommodates are correlated. The number of people accommodated has traditionally been a higher priority search parameter on Airbnb, as it is more relevant for private and shared rooms than the number of bedrooms (and is still the second highest priority parameter when searching on the site, after dates.
- Unsurprisingly, there are perfect correlations between *NaN* reviews (i.e. listings that are not reviewed yet) for different review categories, and first and last review times. *NaN* categories can therefore be dropped.
- The same is true for *host_response_rate_unknown* and *host_response_time_unknown*. One of these rates will be dropped.
- There is a correlation between *host_response_rate 0-49%* and *host_response_time_a few days or more*. One of these will be dropped.
- There are strong negative correlations between *property_type_Other* and *property_type_Apartment*, and between *room_type_Private room* and *room_type_Entire_home_apt* (as these were the main two categories of their features before they were one-hot encoded). Although these are important categories, one of each will be dropped in order to reduce multi-collinearity (apartments and private rooms, as these are the second most common categories).

After dropping problematic features, the following was the final multicollinearity assessment:

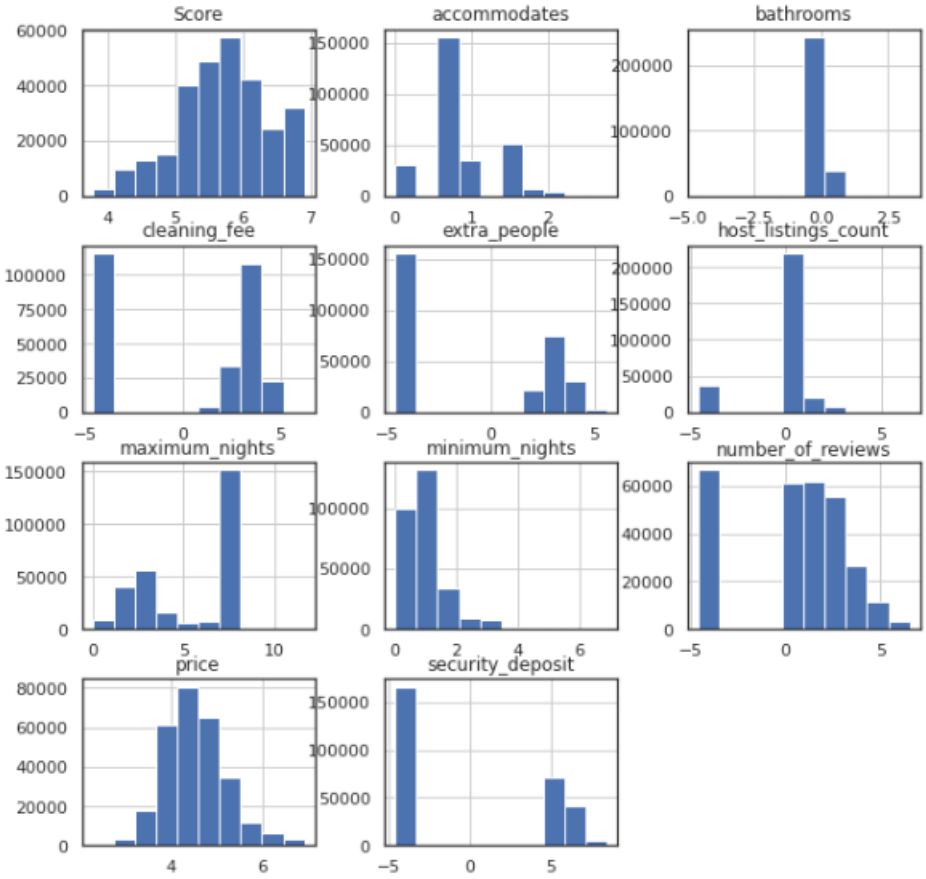
Final Multicollinearity Plot:



There were still some fairly strong correlations between highly rated properties of different reviews categories - i.e. if a property gets a 10/10 for one category, it is likely to get a 10/10 for other category. However, these will be left in for now and can be experimented with later, to see if removing them improves the model.

Other than *availability_90* and *host_days_active*, the remaining numerical features were all positively skewed, and thus log transformation was performed. The transformation helped some of the distributions, although some features contained a large number of 0s, hence these features are not normally distributed. Most importantly, however, the target variable price appeared to conform more closely to the normal distribution.

Log transformed features:



4.2 Models

A Spatial Hedonic Price Model (OLS Regression), with the LinearRegression from Scikit-Learn library and the Gradient Boosting method, with the XGBRegressor from the XGBoost library were used and compared. The evaluation metrics used were mean squared error (for loss) and r-squared (for accuracy).

4.2.1 Model 1: Spatial Hedonic Price Model (SHPM)

The hedonic model involves regressing observed asking-prices for the listing against those attributes of a property hypothesized to be determinants of the asking-price. It comes from *hedonic price theory* which assumes that a commodity, such as a house can be viewed as an aggregation of individual components or attributes (Griliches, 1971) Consumers are assumed to purchase goods embodying bundles of attributes that maximize their underlying utility functions (Rosen, 1974).

In addition to the characteristics of the Airbnb listings, we've added location features as they have been shown to be important factors in influencing the price (see here and here for examples). Ideally, Lagrange multiplier tests should be conducted to verify if there is spatial lag in the dependent variable and therefore a spatial lag model (see this post for spatial regression using Pysal) is preferred for estimating a spatial HPM. However, for the purposes of this post, we are only using a conventional OLS model for hedonic price estimation that includes spatial and locational features, but not a spatial lag that accounts for spatial dependence.

So, the first explanatory variables are the listings characteristics (*acommodates*, *bathrooms*, etc) and our second group of explanatory variables based on spatial and locational features are *Score*, which is the network distance to 5th nearest venue we computed with Pandana;
and *Neighbourhood belonging*, 1 if the listing belongs to the specified neighbourhood, 0 otherwise.

Ridge Regularization was also experimented with, to decrease the influence of less important features. Ridge Regularization is a process which shrinks the regression coefficients of less important features.
The Ridge Regularization model takes a parameter, alpha , which controls the strength of the regularization.

For this purpose, a few different values of alpha were tested to see how this changes results.

4.2.2 Model 2: Gradient boosted decision trees

Boosting is an ensemble technique where new models are added to correct the errors made by existing models. Models are added sequentially until no further improvements can be made. A popular example is the AdaBoost algorithm that weights data points that are hard to predict.

Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

XGBoost (eXtreme Gradient Boosting) is an implementation of gradient boosted decision trees designed for speed and performance. Is a very popular algorithm that has recently been dominating applied machine learning for structured or tabular data.

This approach supports both regression and classification predictive modeling problems.

4.3 Improving models

In the “Preparing the data” section above, it was noted that a lot of the review columns are reasonably correlated with each other. They were left in to see whether they would be useful after all. However, the feature importances graph produced by the XGBoost model suggested that they were of relatively low importance. Thus, all review columns other than the overall review rating we dropped, and the same Hedonic regression and XGBoost structure were used to see whether this produces a better models.

5. Results

5.1 Spatial Hedonic Price Regression Model

For the Spatial Hedonic Price Regression Model, the features explain approximately 40% of the variance in the target variable.

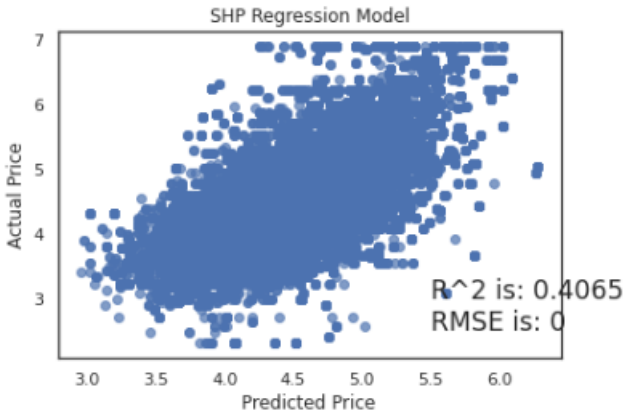
Model 1: Spatial Hedonic Table

Training RMSE: 0.2799
Validation RMSE: 0.2784

Training r2: 0.4028
Validation r2: 0.4065

Interpreting the Mean Squared Error value (RMSE) is somewhat more intuitive than the r-squared value. The RMSE measures the distance between predicted values and actual values. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance, and has the useful property of being in the same units as the response variable. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response. This relationship can be observed graphically with a scatter plot:

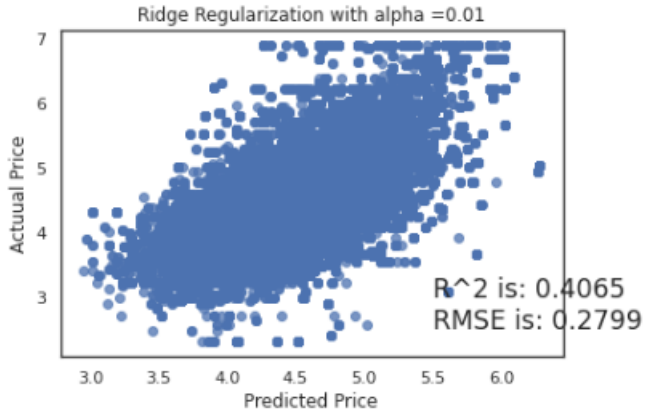
Actual vs predicted price for Hedonic Regression:



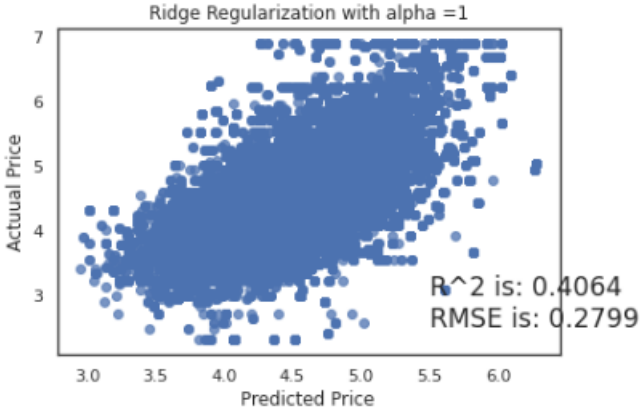
If the predicted values were identical to the actual values, this graph would be a straight line y = x because each predicted value x would be equal to each actual value y.

For the Ridge Regression, the RMSE values are are almost identical as the lambda value increases, which means the model prediction does not improve substantially with the ridge regression model.

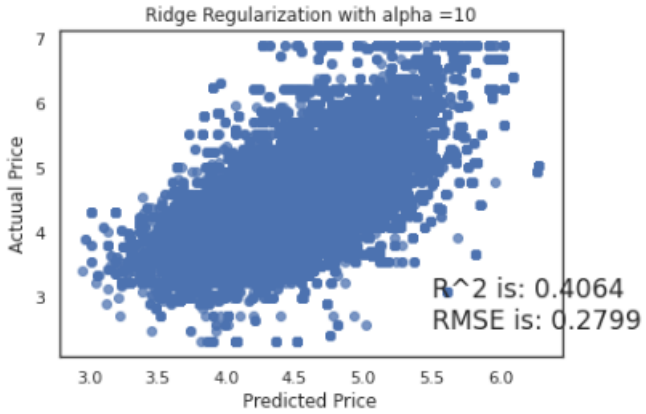
Actual vs predicted price for Hedonic Regression with alpha=0,01:



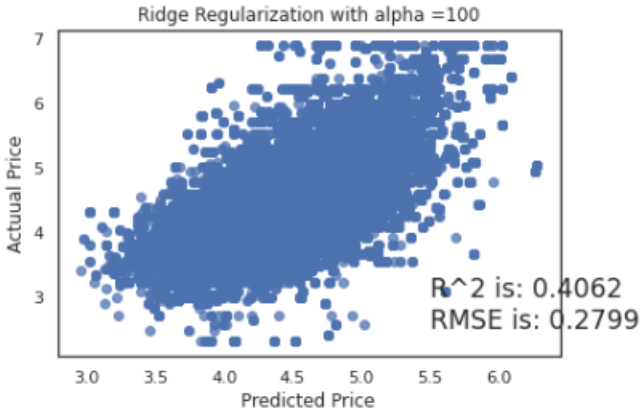
Actual vs predicted price for Hedonic Regression with alpha=1:



Actual vs predicted price for Hedonic Regression with alpha=10:



Actual vs predicted price for Hedonic Regression with alpha=100:



5.2 Gradient Boosted Decision Trees

For the XGBoost model, the features explain approximately 84% of the variance in the target variable, with a lower RMSE compared to the Spatial Hedonic Model.

Model 2: XGBoost Table

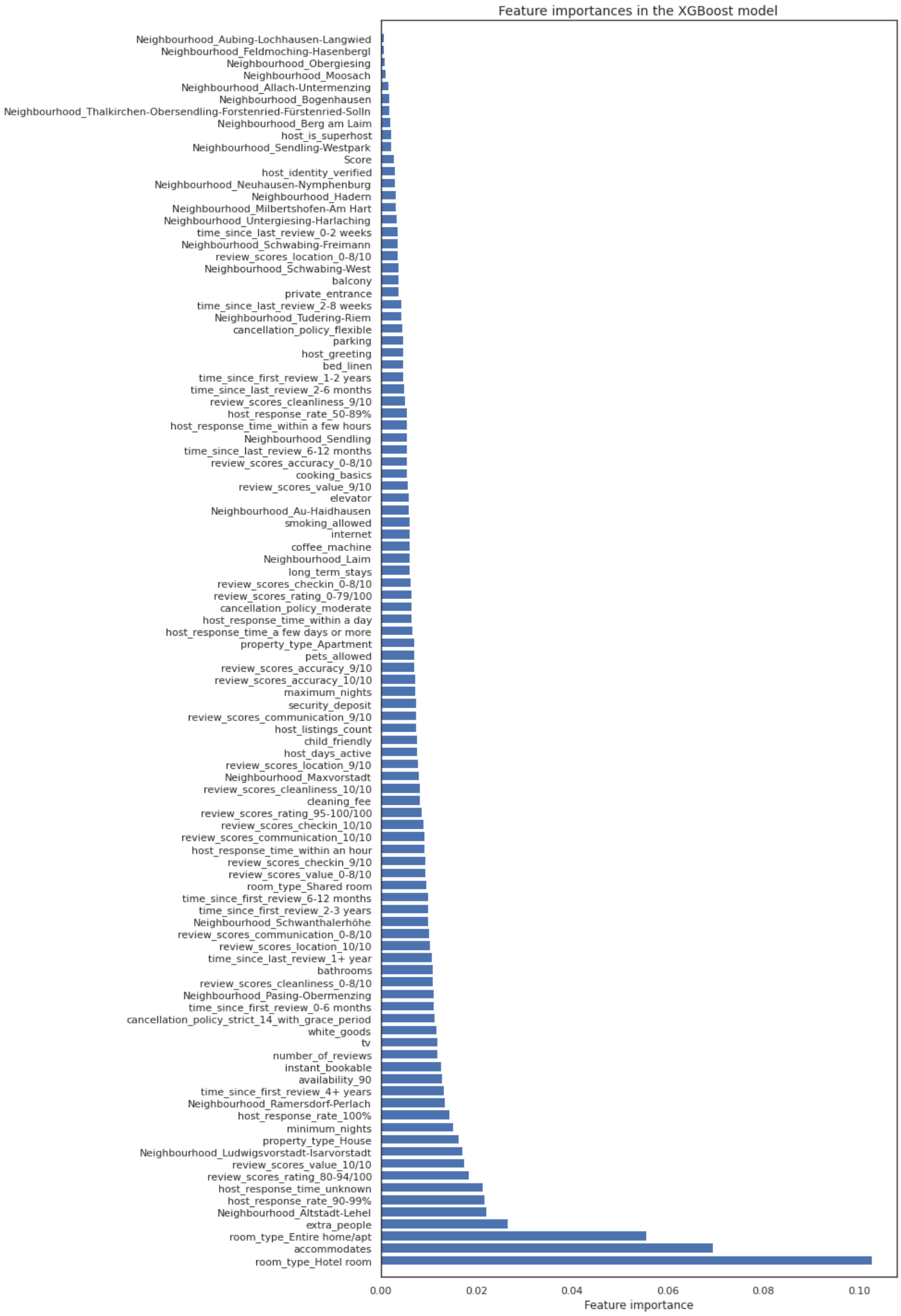
Training MSE: 0.0739
Validation MSE: 0.0761

Training r2: 0.8423
Validation r2: 0.8378

Apart from its superior performance, a benefit of using ensembles of decision tree methods like gradient boosting is that they can automatically provide estimates of feature importance from a trained predictive model.

Generally, importance provides a score that indicates how useful or valuable each feature was in the construction of the boosted decision trees within the model. The more an attribute is used to make key decisions with decision trees, the higher its relative importance.
This importance is calculated explicitly for each attribute in the dataset, allowing attributes to be ranked and compared to each other.

Importance is calculated for a single decision tree by the amount that each attribute split point improves the performance measure, weighted by the number of observations the node is responsible for. The performance measure may be the purity (Gini index) used to select the split points or another more specific error function.
The feature importances are then averaged across all of the the decision trees within the model.


Feature importances in the XGBoost model
Neighbourhood_Aubing-Lochhausen-Langwied
Neighbourhood_Feldmoching-Hasenbergl
Neighbourhood_Obergiesing
Neighbourhood_Moosach
Neighbourhood_Allach-Untermenzing
Neighbourhood_Bogenhausen
Neighbourhood_Thalkirchen-Obersendling-Forstenried-Fürstenried-Solln
Neighbourhood_Berg am Laim
host_is_superhost
Neighbourhood_Sendling-Westpark
Score
host_identity_verified
Neighbourhood_Neuhausen-Nymphenburg
Neighbourhood_Hadern
Neighbourhood_Milbertshofen-Am Hart
Neighbourhood_Untergiesing-Harlaching
time_since_last_review_0-2 weeks
Neighbourhood_Schwabing-Freimann
review_scores_location_0-8/10
Neighbourhood_Schwabing-West
balcony
private_entrance
time_since_last_review_2-8 weeks
Neighbourhood_Tudering-Riem
cancellation_policy_flexible
parking
host_greeting
bed_linen
time_since_first_review_1-2 years
time_since_last_review_2-6 months
review_scores_cleanliness_9/10
host_response_rate_50-89%
host_response_time_within a few hours
Neighbourhood_Sendling
time_since_last_review_6-12 months
review_scores_accuracy_0-8/10
cooking_basics
review_scores_value_9/10
elevator
Neighbourhood_Au-Haidhausen
smoking_allowed
internet
coffee_machine
Neighbourhood_Laim
long_term_stays
review_scores_checkin_0-8/10
review_scores_rating_0-79/100
cancellation_policy_moderate
host_response_time_within a day
host_response_time_a few days or more
property_type_Apartment
pets_allowed
review_scores_accuracy_9/10
review_scores_accuracy_10/10
maximum_nights
security_deposit
review_scores_communication_9/10
host_listings_count
child_friendly
host_days_active
review_scores_location_9/10
Neighbourhood_Maxvorstadt
review_scores_cleanliness_10/10
cleaning_fee
review_scores_rating_95-100/100
review_scores_checkin_10/10
review_scores_communication_10/10
host_response_time_within an hour
review_scores_checkin_9/10
review_scores_value_0-8/10
room_type_Shared room
time_since_first_review_6-12 months
time_since_first_review_2-3 years
Neighbourhood_Schwanthalerhöhe
review_scores_communication_0-8/10
review_scores_location_10/10
time_since_last_review_1+ year
bathrooms
review_scores_cleanliness_0-8/10
Neighbourhood_Pasing-Obermenzing
time_since_first_review_0-6 months
cancellation_policy_strict_14_with_grace_period
white_goods
tv
number_of_reviews
instant_bookable
availability_90
time_since_first_review_4+ years
Neighbourhood_Ramersdorf-Perlach
host_response_rate_100%
minimum_nights
property_type_House
Neighbourhood_Ludwigsvorstadt-Isarvorstadt
review_scores_value_10/10
review_scores_rating_80-94/100
host_response_time_unknown
host_response_rate_90-99%
Neighbourhood_Altstadt-Lehel
extra_people
room_type_Entire home/apt
accommodates
room_type_Hotel room
0.00
0.02
0.04
0.06
0.08
0.10
Feature importance

The top 10 most important features are:

1. If the rental is a hotel room type (*room_type-Hotel room*)
2. How many people the property accommodates (*accommodates*)
3. The type of property (*property_type_home/apt*)
4. Is it possible to include additional guests? (*extra_people*)
5. The rental is in the neighbourhood Altstadt-Lehel (*Neighbourhood Altstadt-Lehel*)
6. The rate of hosts responses to messages (*host_response_rate_90-99%*)
7. If the score of review-ratings is over 80% (*review_scores_rating_80-94/100*)
8. The review scores rating is 10/10 (*review_scores_value_10/10*)
9. The rental is in the neighbourhood Ludwigvorstadt-Isarvorstadt (*Neighbourhood_Ludwigvorstadt-Isarvorstadt*)
10. The property type should be a House (*property_type_House*)

The most important feature is the rental being a hotel room. Which makes sense. Guests are giving high marks to the hotels they book on Airbnb. Guests give their stays in boutique hotels, bed and breakfasts, and other hospitality venues booked on Airbnb an average rating of 4.7 out of 5 stars. The expanded hotel offerings are making it easier for consumers to use Airbnb to find last-minute accommodations when home hosts are often already booked.

It is not surprising that the second feature is how many people the property accommodates, since that's one of the main things you would use to search for properties in the first place.

We can observe that belonging to a certain neighbourhood increases price more than others and it shows some importance, but it's rather of moderate importance compared to the top 3 features. It is also important to be able to include additional guests, and the price is usually higher than having the included guest from the beginning. Review Scores Value is higher on the importance list (number 8). This is, it is likely renters put more weight in the value of the location instead of judging the location based on neighbourhood and venues around the property. This could also be because Munich is a small and walkable city with good transportation services. Thus, location is not a major problem to reaching main touristic attractions and amenities.

5.3 Model improvement

After dropping the review columns, both Spatial Hedonic Price Regression and XGBoost perform almost exactly the same without them. Hence, because they are able to achieve the same performance with 18 fewer columns, the second model is the preferred model as it requires less data and is less computationally expensive.

Comparison between Spatial Hedonic and XGBoost models with dropped columns:

Spatial Hedonic:	***XGBoost:***
Training RMSE: 0.2799	Training MSE: 0.0739
Validation RMSE: 0.2784	Validation MSE: 0.0761
Training r2: 0.4028	Training r2: 0.8423
Validation r2: 0.4065	Validation r2: 0.8378

5.4 Final Model Selection

Overall, the XGBoost model with dropped review columns is the preferred model, which performs better than both Spatial Hedonic Regression Models and just as good as the first model but is less computationally expensive. It could possibly be improved further with hyper-parameter tuning.

If the predicted values were identical to the actual values, this graph would be a straight line y = x because each predicted value x would be equal to each actual value y.

6. Conclusion

The best performing model was able to predict 84.23% of the variation in price with an RMSE of 0.07. Which means we still have a remaining 15.77% unexplained. This could be due to other features that are not part of our dataset or the lack of need to analyse our features more closely.
For example, given the importance of customer reviews of the listing in determining price, perhaps a better understanding of the reviews could improve the prediction. Using Sentiment Analysis, a score between -1 (very negative sentiment) and 1 (very positive sentiment) can be assigned to each review per listing property. The scores are then averaged across all the reviews associated with that listing and the final scores can be included as a new feature in the model (see here for an example).
Another suggestion is the inclusion of image quality as a feature. Using Difference-in-Difference deep learning and supervised learning analyses on a Airbnb panel dataset, researchers found that units with verified photos (taken by Airbnb's photographers) generate additional revenue per year on average (see here).

It was noticeable that reviews about listing location, rather than the location features themselves, were higher in the feature importance list. Thus, this finding could perhaps be used by Airbnb hosts when writing their listing's description. Highlighting accessibility and location benefits of staying with them could perhaps benefit them and how much they can ask for their listing.

7. References

Marco Peixeiro. 2019 - The Complete Guide to Time Series Analysis and Forecasting. Available at: https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775

Kuan Butts. 2017 – Point of Interest Accessibility. Available at: http://kuanbutts.com/2017/08/08/how-to-pdna/

Geoff Boeing. 2016 - How to Visualize Urban Accessibility and Walkability. Available at: https://geoffboeing.com/2016/07/visualize-urban-accessibility-walkability/

Kalehbasti, Pouya & Nikolenko, Liubov & Rezaei, Hoormazd. (2019) - Airbnb Price Prediction Using Machine Learning and Sentiment Analysis. Available at: https://arxiv.org/pdf/1907.12665.pdf

J. Elith, J. R. Leathwick, T. Hastie. (2008) - A working guide to boosted regression trees. Journal of Animal Ecology. Available at: https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/j.1365-2656.2008.01390.x

Zhang, Shunyuan and Lee, Dokyun (DK) and Singh, Param Vir and Srinivasan, Kannan. (2017) - How Much Is an Image Worth? Airbnb Property Demand Estimation Leveraging Large Scale Image Analytics. Available at SSRN: https://ssrn.com/abstract=2976021 or http://dx.doi.org/10.2139/ssrn.2976021

Chen, T., Guestrin, C., (2016) - XGBoost: A Scalable Tree Boosting System, Proceeding KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 785-794. Available at: https://dl.acm.org/citation.cfm?id=2939785

Rosen, Sherwin, (1974) - Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition, Journal of Political Economy, 82, issue 1, p. 34-55, https://EconPapers.repec.org/RePEc:ucp:jpolec:82:y:1974:i:1:p:34-55.

Liv Osland (2010) - An Application of Spatial Econometrics in Relation to Hedonic House Price Modeling. Journal of Real Estate Research: 2010, Vol. 32, No. 3, pp. 289-320. Available at: https://www.aresjournals.org/doi/abs/10.5555/rees.32.3.d4713v80614728x1

Luxen, D. and Vetter, C. (2011) - Real-time routing with OpenStreetMap data. Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, New York, NY, USA,Available at: http://doi.acm.org/10.1145/2093973.2094062

Limsombunchai, V., (2004) - House Price Prediction: Hedonic Price Model vs. Artificial Neural Network, Paper presented at the 2004 NZARES Conference Blenheim Country Hotel, Blenheim, New Zealand. Available at: https://core.ac.uk/download/pdf/35467021.pdf

Fletcher, F, and Waddell, P. (2012) - A Generalized Computational Framework for Accessibility: From the Pedestrian to the Metropolitan Scale. Transportation Research Board Annual Conference Available at: http://onlinepubs.trb.org/onlinepubs/conferences/2012/4thITM/Papers-A/0117-000062.pdf