University of Copenhagen

Airbnb Host Pricing
A Principal-Agent Problem

Examination No. and Sections:
212: 1.2.1, 1.2.2, 3.2.2, 3.2.3, 5.1, 6.2, 7.0, 7.1, 7.2, 7.4
236: 2, 3.3, 4.0, 4.1, 6.2
135: 1.1.1, 1.2, 2, 3, 3.1, 3.2.0, 3.2.1, 4.2, 5.0, 5.3, 5.4, 5.5, 5.6, 6.0, 6.1, 7.3, 7.4, 8
207: 1.1.0, 1.1.1, 1.1.2, 5.2, 6.1, 7.2, 7.4, 8, 9

Course 2200-B5-5F18
Social Data Science
August 2, 2019

Character count: 48.845

KØBENHAVNS
UNIVERSITET

Contents

1 Introduction

This paper sets out to model the pricing mechanism for Airbnb listings in Copenhagen. We will gather data publicly and apply different models to obtain a way to determine the price of a listing based on only data host post on the Airbnb website.

1.1 Introduction to Airbnb

Airbnb was officially founded ten years ago in San Francisco, California. As of 2018, the company was valued at USD 38 Billion [1]. Airbnb has utilized the shift towards the *sharing economy* with clever use of opportunities provided by modern technology. In brief, Airbnb operates a marketplace which enables users to list and rent living spaces across the globe. While the company has expanded their value proposition in the recent years with tourism-related services, the marketplace is the key driver behind the success of Airbnb.

> Today, Airbnb-beds outnumber the amount of hotel rooms in Denmark.
>
> *Horesta, March 2017 [2]*

Airbnb has proven an issue for Danish policy-makers since it is an internationally operating company; its European headquarters is in Ireland, (which has a different tax system) creating the possibility for renters in Copenhagen to tax evade [9]. Moreover, home sharing practices have proven to increase short-term rents in major cities [8] and decrease the demand for hotels and hostels.

Recently, a historical agreement has been made between Airbnb and the Danish government, since it will be the first time that the company will automatically be sharing their data upon homeowners' incomes with a country's tax authority. The agreement is in line with the government's objective of a sharing economy, as long as the corresponding taxes are paid. It still needs to pass from the parliament, but it includes the limitation of the amount of days that an owner can list its property (to 70 days per year) and gives a tax-free allowance up to 40.000 DKK per year [10] .

1.2 Research Question

This paper serves to answer whether or not it is possible to construct an Airbnb pricing model from publicly available data. This is motivated by three reasons. First of all, we believe there might be a principal-agent problem between Airbnb and Airbnb hosts due to a mismatch in market knowledge. Secondly, Airbnb can influence hosts by anchoring the host on a given price, as suggested by the Airbnb pricing algorithm. Thirdly, if a pricing model can be constructed, this serves as a basis for creating an independent pricing model for hosts where the endogeneity from the current data is removed.

1.2.1 The Principal-Agent Problem and Anchoring

As pointed out by Steven D. Levitt and Chad Syverson infamous working paper titled *"The Value of Information in Real Estate Transactions"*[3], the real estate agents (in this case the Airbnb cooperation) receive only a fraction of the incremental profit when the listing is priced higher. This means that Airbnb has an economical incentive to secure more bookings by setting a lower price, since the end customer is more likely to rent a given listing, when the price is relatively low. This means the host will lose a relatively larger share of revenue than Airbnb when the price is lowered.

Airbnb might exploit this opportunity by displaying the before mentioned "suggested price" as a form of 'anchor'. As Amos Tversky and Daniel Kahneman proved in their article *"Judgment under Uncertainty: Heuristics and Biases"*[4] this form of priming – more specifically 'anchoring' – has a substantial effect on the decision making of an individual. When the Airbnb host-to-be is presented a suggested price, it will act as an anchor and will unconsciously affect the decision they will make – and probably not to their own benefit. This is possibly a way for Airbnb to exploit the asymmetric information in the market in which it acts as a monopolist. The anchoring may be extra effective since people do not have a good natural estimation of the listing price of their own home. This is enhanced by the fact that they are not presented with listing prices and availability of similar homes, but have to research it themselves.

Here we have a textbook example of the principal-agent problem affecting the 24,289 active Airbnb hosts in Copenhagen. Our proposal to a solution is to create total transparency in the estimation of a listing price. This exact problem has been discussed in regards to the similar company Uber, which offers peer-to-peer 'ride-sharing'. Despite the fact that the opportunity cost for an Uber driver is larger than the one for an Airbnb host, we will apply the same discussion to the peer-to-peer 'home-sharing' market. We will build our own model from our available data and try to determine whether or not it is possible to build a pricing model which can serve as an alternate estimator for the price of a listing.

1.2.2 Reservations about data and potential biases

In this paper we are using a cross-section data set from Airbnb listings, meaning that the exact algorithm we want to make transparent already will have affected our data and therefore also our model. We cannot completely remove this form of endogeneity from the model at this time, which we will discuss later. We therefore expect to estimate a lower price than the true price of a given listing. Of course we cannot know for sure whether this algorithm has an intentional bias or not, but this still means that this model will ultimately estimate the Airbnb prices as they were in the market per July 29^{th}, 2018.

Even though there might be a negative bias in the Airbnb model, this is not directly implemented on all listings, since the hosts in the end will make the decision, making the bias not-consistent throughout the data set. Our approach to the model will therefore be to estimate the price as precisely as possible from the features we have in the data set and the ones we can create on our own. This means that removal of the endogeneity is beyond the scope of this paper, but we will be investigating the possible bias through out the paper. It is central to this paper that our model could be applied on the same data structure without any bias and will present a non-influenced price.

2 Web-Scraping and Ethics

As many other tech-giants, Airbnb has not been interested in sharing their data publicly. Only when strictly necessary, data has been shared with public institutions in order to advance the agenda of Airbnb. On the rise of the secular digital trend, the value of data has been ever appreciating and correspondingly less and less data is being made public. Few companies, such as Uber release their data to certain scientists under specific contractual agreements, as to not hurt their business case. While this is not the case with Airbnb, some data is available due to the fact that the service is an online marketplace where some information must be shared between buyer and seller in order to facilitate a transaction.

By utilizing fairly recent information gathering techniques such as web-scraping, it is possible to gather a rich data set with a variety of features describing each listing. While Airbnb on purpose makes this quite difficult to do, some individuals have overcome this challenge in order to make the data public.

The increasing amount of public data on specific individuals and items of importance in their daily life may raise privacy concerns. While this information willingly has been supplied by individuals by listing their homes, the data should not be used in unethical contexts. Airbnb slightly protects the host by blurring the listing address and removing surnames making the listings less identifiable, but the data is still sufficient for modeling and estimating Airbnb impact.

3 Exploratory Data Analysis

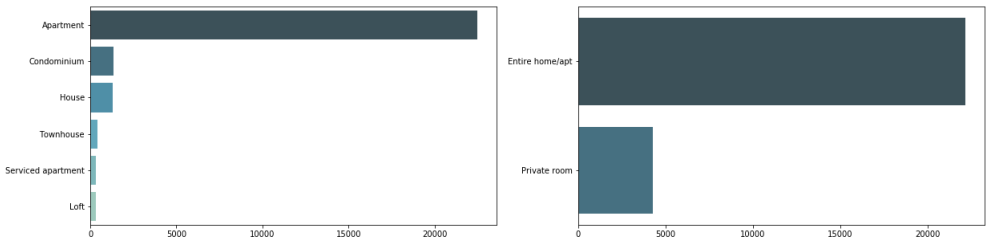
The main data set was compiled July 29^{th}, 2018 by *insideairbnb.com* and consists of 26.560 unique (both active and inactive) listings of living spaces in the Greater Copenhagen region and has 96 descriptive features. They are both numerical as well as categorical and are primarily concerned with host or home attributes. Furthermore, names of Metro and S-Train stations were scraped from *Wikipedia* in order to be used with geographical data as described in section 3.3.

3.1 Categorical Features

In this section, we will take a closer look at some of the categorical features in the data set. They will be manipulated in different ways to make them comprehensible for the prediction models that will be used later on.

Each listing indicates the type of property and if relevant the type of room. From Figure 1, we can, not surprisingly, observe that the clear majority of listings in Copenhagen are apartments.

Figure 1: Property and room type for the living space listings.



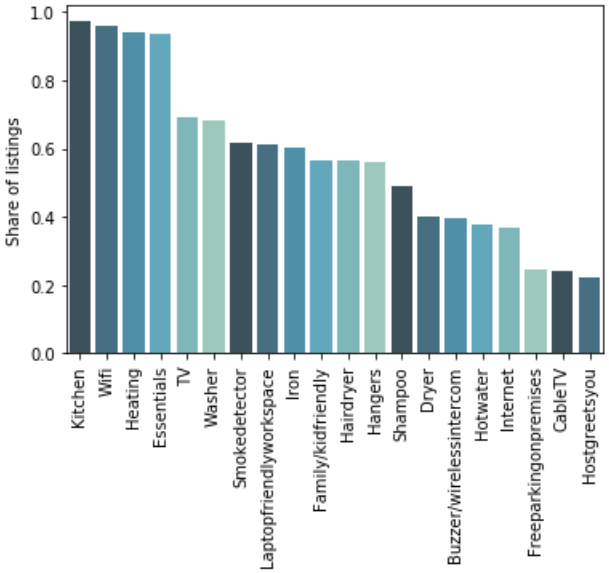
Note: Property types with less than 200 observations have not been included.

We also note that roughly 80 percent of listings are for complete living spaces. One possible reason could be that people who are willing to rent out a single room would prefer a more permanent tenant, e.g. a student, reducing uncertainty of future availability drastically. On the other hand, Airbnb's pricing per night is significantly above the monthly average rent in the open market, creating an incentive for using the Airbnb marketplace and not a permanent tenant.
The property type plays a role in indicating which amenities will be available at that location. Literature has indicated amenities are important for evaluating prices and Airbnb's own pricing model also puts a fixed premium on certain amenities such as WiFi [16].

Some amenities figures are fairly accurate. A report from the Danish Emergency Management Agency from 2009 stated that around 62 percent of households in the Copenhagen area own a smoke detector [6], which is consistent with our data. However the magnitude of some amenities differ in the report and in our data set. An example is WiFi which more than 95 percent claim they have while less than 40 percent claim to have internet[1]. This could prove problematic for the model due to unreported items in the data.

[1]This could also be interpretation issues as *internet* could be understood as cabled access.

Figure 2: Top 20 amenities in Airbnb listings.



3.2 Numerical Features

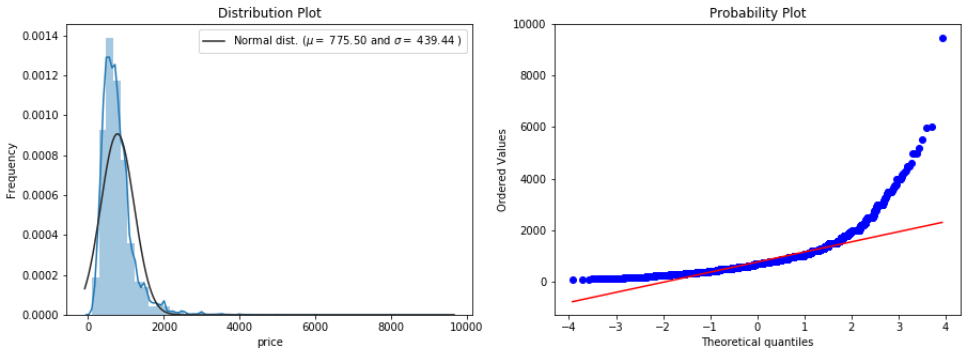
In this section we take a closer look at some of the numerical features in our data set; the most important being the price of the listing, which will be the target variable in our predictive models. This section also includes the distribution of reviews as well as other key features.

3.2.1 Price

The price variable is in DKK[2] and indicates the listing price for a single night. The data set features some outliers, and we have chosen to omit observations where a single night per person costs more than 2.000 DKK, e.g. removal of listings with a price of more than 63.000 DKK with accommodation for four. This leaves us with the distribution shown in Figure 3.

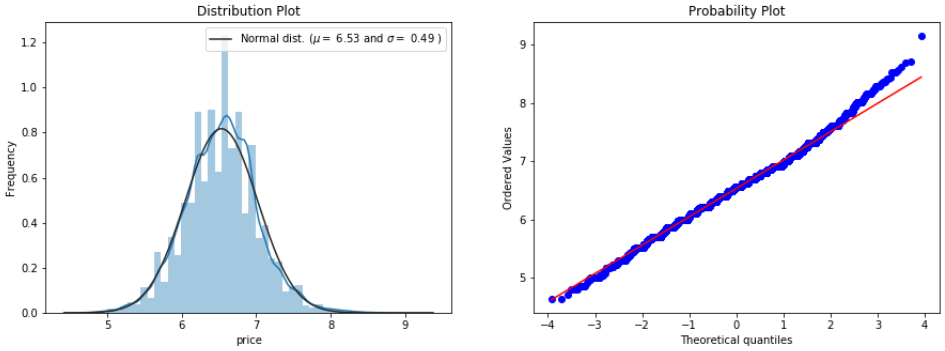
[2]Raw data set indicates $ but is not accurate.

Figure 3: Price per night



In order to optimize the model we have chosen to transform the price variable with the natural logarithm. We see from Figure 3 that the price variable is not normally distributed. By applying the natural logarithm, the variable is transformed and it is clear from Figure 4 that the transformed variable is closer to being normally distributed, yet with some inconsistency as indicated by the probability plot.

Figure 4: log(Price) per night

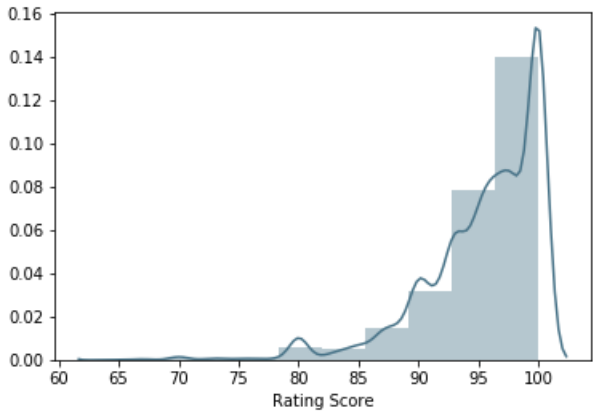


3.2.2 Reviews

Across 15,795 listings[3] the average rating of an Airbnb listing is 95.27 on a scale from 20 to 100 (1 to 5-star rating system). The variable functions as an average total rating across different topics like cleanliness, responsiveness, value for money etc. We see that 90 percent of listings have a rating of 90 or more; 27 percent of listings have a rating of 100. The distribution can be seen in Figure 5. These are remarkably high ratings, and they do not vary a lot - the standard deviation is 5.4 meaning there are pretty much only very positive reviews.

[3]The drop in observations from 26,560 to 15,795 will be explained in section 4

Figure 5: Distribution of review scores



Furthermore there is a variable representing the number of reviews. The average listing has 18.25 ratings, but this varies more across the data set than the average review, as the standard deviation is 26.87. This is primarily due to relatively few large observations, as the max observation is 473, and the 99^{th} percentile is 134.

The data set also includes reviews given per month. The median listing receives one review every second month. The distribution of this variable (as seen in Table 1) tells the same story as the distribution of number of reviews. There is a lot observations around the mean, but relatively few outliers, giving us yet another distribution skewed towards the right.

Table 1: Descriptive statistics

	Rev. score	No of Rev.	Rev./month	Accom.	Listings/host
mean	95.27	18.25	1.05	3.31	1.37
std	5.40	26.87	1.20	1.62	2.29
min	20.0	2.0	0.02	1.0	1.0
25%	93.0	5.0	0.32	2.0	1.0
50%	97.0	10.0	0.64	3.0	1.0
75%	100.0	21.0	1.29	4.0	1.0
max	100.0	473.0	17.0	16.0	60.0

3.2.3 Listings and accommodation

Accommodation is the best available measure for size in the data set.[4] Even though this is not a perfect medium, there is a clear tendency if more people can sleep in a given listing, the residence is to some extent larger. Furthermore extra space is not of great importance in an Airbnb listing

[4]Size in square feet is a feature, but the data only exists for approximately 600 listings in total.

since it is not for permanent residence, but mostly a substitute for a hotel room. The average listing can accommodate 3.3 persons, ranging from 1 to 16. A relatively small standard deviation of 1.6 implies that most listings are apartments for a couple or small family.

Meanwhile, the average host has 1.37 listings; 81 percent of the hosts has only one listing, while only five percent has more than five. This implies that the majority of the Airbnb market in Copenhagen are for people occasionally renting out their own apartment. But there is still a small share of people renting out many apartments with the max being 60 listings per host, probably as a business.

3.3 Geographical Variables

This section serves to describe some of the geographical features contained in the data set, as well as some further features are constructed in order to improve the performance of the model.

3.3.1 Relative Distances

In the original data set, listings include a feature *transit*. As *transit* is a description in words regarding transportation possibilities, it does not have any specific format, and therefore it is hard to parse the relevant information. Luckily the data set also contains both latitude and longitude, which enables the option of calculating distances from apartments to specific locations, such as nearby stations. The location of a listing is not accurate, but within 150 meters of the exact location.

We can include the estimated distance to transit as a feature and potentially improve the pricing model. For each apartment three distances is calculated: The distance to Nørreport, the closest S-Train station and the closest Metro station. The distance to Nørreport station is included as it represents the distance to central Copenhagen. The first step to calculate the distances is obtaining the names of all Metro and S-train stations. These names are scraped from *Wikipedia*. Regular expressions as well as other string manipulations are applied on the raw HTML in order to identify the list of stations.

To calculate the distance, the longitude and latitude of the stations are needed. *gps-coordinates.org* is a website which provides latitude and longitude, given a location as a string. The outgoing network activity from the website, revealed the query which is used to get the latitude and longitude. The query even contains an access token, making it possible to use the request in the code. However the token is dynamic, so it needs to be updated daily. A sleeper, assigned to one second, is active between each query call, to reduce the traffic on the server. To calculate the actual distance d between two destinations, the Haversine formula is used. *Note that* φ *is*

latitude, λ is longitude, R is the radius of the earth, and all angles are in radians.

$$a = sin^2(\frac{\Delta\varphi}{2}) + cos\varphi_1 \cdot cos\varphi_2 \cdot sin^2(\frac{\Delta\lambda}{2}) \quad \text{(Haversine)}$$
$$c = 2 \cdot atan2(\sqrt{a}, \sqrt{(1-a)})$$
$$d = R \cdot c$$

From this process, we now have the distance to nearest S-train and Metro stations, as well as the distance to Nørreport, and all of them are presented in Table 2. The average distance to the Metro station is a bit shorter than the one to the S-train, meaning that the Airbnb listings are placed very much along the Metro line of Copenhagen. The distribution of the distance between the listings and the public transportation is right skewed, while the one to Nørreport tends to symmetric (indicating Nørreport is a good medium as city centre by assuming the geographical placement of listing can be described by a normal distribution). The right skewness of the distance to public transportation stations means that the listings are generally close to stations, while a few are far away.

Table 2: Distance from listings to interest points (in km)

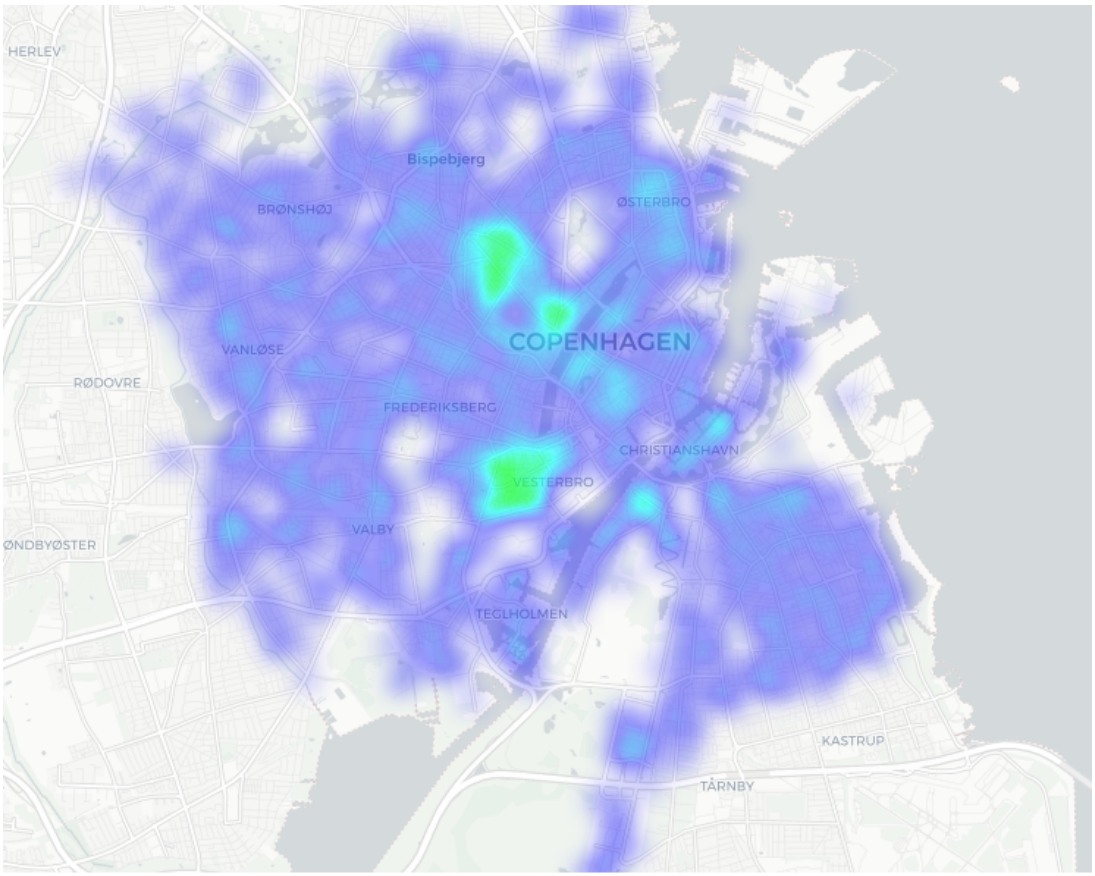
	S-train	**Metro**	**Nørreport**
mean	1.15	0.92	3.33
std	0.98	0.54	1.54
min	0.01	0.01	0.02
25%	0.55	0.52	2.28
50%	0.84	0.84	3.17
75%	1.27	1.21	4.27
max	5.76	4.45	9.89

The correlation between the distance to Metro stops and Nørreport is 0.4, meaning the closer to city centre the listings is placed, the closer it is to a Metro stop. The correlation between the distance to S-train stops and to Nørreport is -0.4, meaning that the further away the listing is from Nørreport, the closer it generally is to a S-train stop. Therefore – as it should be – the suburban train covers mainly the suburbs, and the Metro covers mainly the city centre.

3.3.2 Pricing Heat-map

The map clearly shows that areas such as Vesterbro and Nørreport as well as inner and central Nørrebro are popular places to list apartments. The tendency is still that center proximity is correlated with amount of listings, which is in alignment with a tendency from tourists wanting to live centrally with access to the rest of the city by public transport. It is noteworthy that even though it visually seems like Nørreport has a great amount of listings, it only contains 1.54 percent of all listings based on a range of 0.5 km.

Figure 6: Heat-map of Airbnb listings in Copenhagen



Note: The heat-map is created using latitude and longitude with no weight.

4 Data Processing

The data set we are using contains 96 features and 26,560 listings. Not all of these features are relevant, and before feeding our prediction model with data, we will remove or transform those with little or no significance for our model. This section contains the documentation of the most important transformations.

4.1 Removal of observations

To feed our model with relevant information, many listings is removed from the original data. Some of them have missing data, which cannot be generated with imputation. For features such as *host, bedroom, bathroom, bed neighbourhood* and *review data* some listings do not have data and are therefore removed. The focus of the pricing model is short term rentals, which is why all the listings with rental periods greater than 31 days are discarded. Inactive users i.e. people

who have not updated their calender for a year is removed as well, resulting in the loss of 3031 listings. Moreover outliers such as apartments with considerable high prices (2k + per person per night) as well as prices below 100 DKK is removed. Finally, listings with less than two reviews are removed for two reasons. The first reason being these listings tend to be cheaper than typical similar listings due to a missing signaling factor of a good review. Secondly, since there are no reviews, this data is missing in different features. By making sure listings have reviews, we are certain that they have been 'traded' on the marketplace and therefore there is a good chance that the price is attractive to a given customer[5]. Listings with less than two reviews is by far the biggest reduction to our data set as it subtracts 7,528 listings.

4.2 Imputation of values

The original data is not formatted very well, e.g. areas containing multiple zip codes. To simplify this, all *zip codes* are grouped in to 18 areas, where one area is categorized as *other*. Not all listings contain both *zip codes* and *neighbourhood*. If the *neighbourhood* feature is missing while the *zip code* is listed, the neighbourhood value can be generated from it. If both *neighbourhood* and *zip code* is missing then the listing is dropped.

The *neighbourhood* feature is reduced into six categories from 18. Columns containing irrelevant data are dropped out, which discards 59 features. Throughout the formatting, imputation of missing features is done by either using the mean from all other listings or imputing 0. Categorical variables are onehotencoded[6] unless a categorical ranking is clear. The *amenities* feature is also onehotencoded, resulting in an additional 124 features. More processing is done on the data set, while all of the above mentioned are some of the noteworthy transformations.

5 Machine Learning

This section seeks to explain the different techniques that are used in order to build the prediction model, from the construction of new features with sentiment scoring to ensembling regression and random forest based models.

5.1 Sentiment Analysis

The data set contains a feature *summary* which is a string, written by the host as a short description of their listing. This string consists of 55 words on average. The summary primarily regards the general "theme" of the listing. We have applied sentiment analysis on this feature to examine if the host's wording has any effect on the pricing of the listing. Our hypothesis is

[5]The price can of course vary over time, but we assume that no drastic price changes are made after receiving reviews.

[6]Dummy creation in sklearn

that the more experienced the Airbnb host is, the better description he will be able to write, resulting in more bookings and higher prices.

5.1.1 Constructing text features

We have transformed every summary from a string to an integer, rating the sentiment of the summary. To do this we have used the AFINN lexicon[5]. The AFINN word list contains $2,477$ words, and a manually assigned value for each word. The range is from -5 to $+5$ where $+5$ is a very positive sentiment. The function looks for every given word in the *summary* string, summarizes the points and thereby assigns a score to the *summary*. We did not compute a score based on number of words, since we have many words that are not identified, favouring the short summaries. Originally, AFINN was built to categorize tweets, which may have an impact on the precision of our model as the type of language used on Airbnb might not be similar to tweets.

In order to identify the language of the strings, LangDetect is used. LangDetect is based on Google's Language Detector Model through the Cloud Translation API implemented in e.g. the Google Translate platform. The model has a 99 percent precision across 53 languages; including Danish and English. When applied to our data set only 0.4 percent of the values within the *summary* were not identified as either Danish (12.6 percent) or English (87 percent). Summaries with languages that were neither Danish or English were imputed with the score 0.

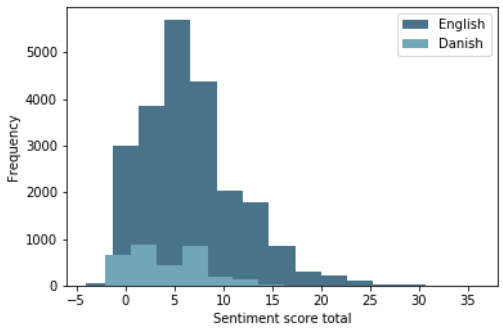
5.1.2 Sentiment of summaries

With the *summary* and *language* as input we can now score each listing. From Figure 7a it is clear that we find a tendency towards a positive sentiment in the summaries and that not all are equally positive. This confirms the hypothesis that some hosts will write more flattering descriptions about their listings than others. This might prove useful in aiding the prediction model. The data does not have a large variance, meaning the summaries are much alike. An average sentiment score of 9 is not impressive when the average word count of the summary is 55, but neither remarkably low.

Further investigation from a TF-IDF analysis shows that many of the top words are very practical and therefore do not have a clear sentiment as seen in Figure 7b. A more customized list would therefore outperform the standard AFINN list. These conclusions are also found when the title of the Airbnb listing is analyzed. Even though the titles are shorter (about 6 words on average), they have even fewer popular words like "apartment", "central" and "close" dominating the TF-IDF analysis. This gave almost an identical shape of the distribution, but with lower variance and therefore we chose the summary data. The top 100 words has a total sentiment score of 25, meaning each word on average is just barely positive. In fact 10 of the top 100 words are positive, while there are no negative words in the top 100.

Figure 7: Sentiment analysis result

(a) Distribution of sentiment score



(b) Word cloud of top 100 words from IDF-IF analysis

5.2 Data Scaling

Regression computes the Euclidean distance between two data points, which could affect the predictive performance of our model since our data set has different magnitudes. We can overcome the issue by scaling our data to the same magnitude [11].

One of the most common scaling methods is StandardScaler; it assumes that the data is normally distributed and scales it with the mean equal to zero and the standard deviation to one, i.e. to a standard normal distribution, using the following formula:

$$\frac{x_i - mean(x_i)}{std(x_i)} \qquad \text{(StandardScaler)}$$

Our data set has two characteristics which makes this scaling technique less optimal; firstly, not all of our data is normally distributed. Secondly, if scaled to a standard normal distribution, in order to compute the empirical mean and standard deviation, the outliers will shrink the range of the feature values. To overcome these two problems we apply another scaling technique, RobustScaler, which centers and scales the features on percentiles, using the following formula:

$$\frac{x_i - Q_1(x)}{Q_3(x) - Q_1(x)} \qquad \text{(RobustScaler)}$$

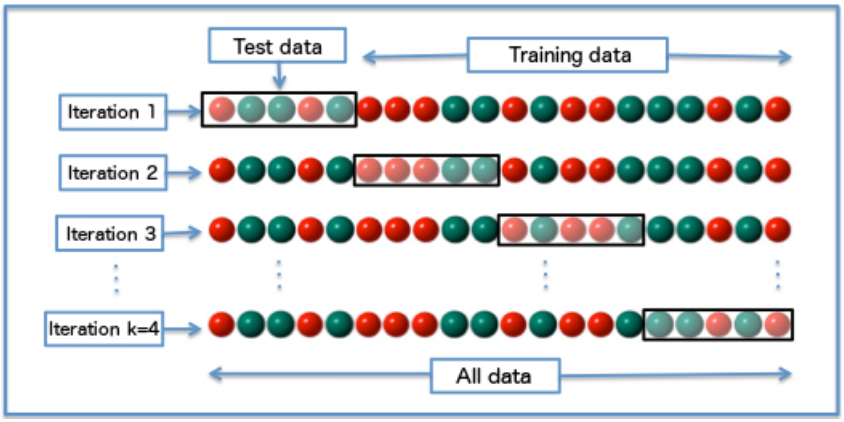
This scaling technique will cause our model to be less vulnerable towards outliers [12].

5.3 Cross-Validation

In order to ensure optimal performance from the pricing model a range of hyperparameters have to be optimized. Ensuring a hermetic seal between training and test data is of utmost importance

since critical errors will occur if the model learns from test data.
By using the form of cross-validation known as k-fold-cross-validation, it is possible to train the supervised model on a specific training data set, while a separate test data set is created for final evaluation purposes. The methodology is as follows. The training set is split randomly into k-folds (in our case, $k = 10$). From the ten samples, nine are selected to train the model upon, where the last one is used to validate the model predictions. This process is repeated ten times, one for each fold, of which the mean of the validation performance will be the final score.

Figure 8: Illustration of the k-fold-cross-validation with $k = 4$ subsamples



The methodology is advantageous compared to a train-validation-test split in which the validation data will not be used for the initial training of the model and only a subset of training listings is used for validation.

5.4 Prediction Techniques

In the following section, the various techniques used for price prediction is discussed, as well as certain challenges that is faced when building a prediction model.

There exists a trade-off between bias and variance in prediction models. Fortunately, by utilizing certain techniques it is possible to manipulate the trade-off, e.g. penalizing over-fitting, in order to create the best possible model. Generally, as model complexity tends to increase, the variance will also increase and the squared bias will decrease. The opposite will occur as model complexity decreases.

The goal of the model is to reach the optimal trade-off between variance and bias in order to minimize test error. An obvious indicator of test error could be the training error, but this is not the case. By increasing model's complexity the training error will decrease, as the model

learns the training data too closely, leading to over-fitting in training data, i.e. poor model generalization. [7]

5.4.1 Lasso

The Lasso regression model is a linear regression model with L1 regularization. L1 regularization consists of penalizing certain weights, resulting in sparse feature vectors where feature weights can be zero [7]. This is useful in our data set as we have a large amount of features, of which not all may be relevant for predicting the price. The model is penalized with the absolute value of the model weight, as seen here:

$$J(w)_{Lasso} = \sum_{i=1}^{n} \left(y_i - x_i'\beta\right)^2 + \lambda \cdot \sum_{j=1}^{p} ||\beta_j|| \qquad \text{(LASSO)}$$

It is possible to optimize the hyperparameter λ[7] of the Lasso model in order to achieve the best possible model fit. By implication, this also means that for $\lambda = 0$, no penalty will occur and therefore the Lasso model will be similar to OLS regression. Lasso does not perform very well on data sets with a high number of features and few observations. Utilizing Lasso on such data sets result in saturation with Lasso selecting at most the number of observations as weights.

5.4.2 ElasticNet

ElasticNet is a model which integrates both L1 and L2 regularization. L2 regularization consists of adding a penalty of the squared sum of the weights, as follows:

$$\lambda||\beta||_2^2 = \lambda \sum_{j=1}^{m} \beta_j^2 \qquad \text{(L2 Penalty)}$$

This leads to a model where feature vectors will not be set to zero, but some feature weights will be diminished significantly leading to little to no impact. This leads us to the structure of the ElasticNet cost function[7]:

$$J(w)_{ElasticNet} = \sum_{i=1}^{n} \left(y_i - x_i'\beta\right)^2 + \lambda_1 \cdot \sum_{j=1}^{p} ||\beta_j|| + \lambda_2 \sum_{j=1}^{m} \beta_j^2 \qquad \text{(ElasticNet)}$$

In the sklearn implementation of ElasticNet, it is possible to adjust two hyperparameters. The first being the weights between L1 and L2 regularization, the second being the λ described in 5.4.1 indicating how much weights should be penalized[13].

5.4.3 Gradient Boosting Regressor

The Gradient Boosting Regressor is based on very simple classifiers, known as weak learners. An example of such a weak learner could be a decision tree stump [7]. In simple terms, the model

[7] This hyperparameter is indicated as α in the sklearn package.

fits a given data set, fits the residuals of that model and then creates a new model from the two first fits, i.e. as formulated here:

$$F_{i+1}(x) = F_i(x) + h_i(x)$$

where F_m indicates the function fitting a given y and h_i is defined as the fit residuals $h_i = y - F_i(x)$. By repeating this process, it is possible to create a model that learns from previous mistakes. The number of iterations, the model has to go through, is best determined with cross-validation.

Residuals are minimized with Gradient Descent, enabling minimization of more complicated cost-functions, with the requirement of differentiability. The gradient is calculated at a given point as an average between leaf nodes. Then a "step" is taken in the direction with the largest descent. These steps are to be repeated with a small enough step size, as well as a large amount of iterations in order to determine the minimum of the given cost-function[14][15].

For this analysis, the regression technique used is not *least squares* but *huber*, a combination of OLS[8] and *least absolute deviation*,[9] as a more robust technique to prevent outlier influence.

5.4.4 XGBoost

eXtreme Gradient Boosting is, as the name suggests, built upon the same idea as Gradient Boosting. It improves upon the computational speed and model performance of Gradient Boosting by utilizing some of the same techniques, yet with key differences. XGB introduces model regularization and tree-pruning in order to combat over-fitting. The optimization technique for each iteration is not quite similar to Gradient Boosting as XGB approximates a second degree Taylor-polynomial and optimizes over this. Therefore, the model is also more restricted as second degree differentiability is required for the cost function. [22]

5.5 Hyperparameter Optimization

The prediction model is optimized with three different techniques, depending on the amount of hyperparameters and the trade-off between computational time and model accuracy. The first optimizer is GridSearch. It creates a model on each combination of hyperparameters, a computationally expensive operation. Scikit-learn implements this by iterating over every combination using cross-validation. The different combinations of hyperparameters are each validated where the model with the lowest error is selected.

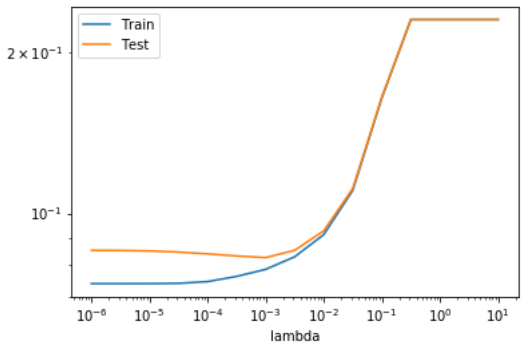
RandomSearchCV is similar to GridSearch in the way that it can optimize models with multiple hyperparameters. Instead of searching through each possible hyperparameter structure,

[8] Initial model is given by the mean of the target values
[9] Initial model is given by the median of the target values

it randomly selects from distributions in order to sample the performance of the model with the given hyperparameters. While RandomSearchCV is not as computationally expensive as GridSearch, the performance is worse as it does not cover all possible options but only a random subset.

A more simple implementation of GridSearch is the Validation Curve. When only one hyperparameter has to be optimized, it is not as computationally expensive to determine the optimal value. An example of a validation curve can be seen for the Lasso model used in our prediction model in Figure 9. In the figure, we also see that the model converges. This is the case for all

Figure 9: Convergence of the Lasso model



of our models used to predict the listing price.

5.6 Ensembling

While a single model can produce fairly good results when optimized, the combination of multiple models can further improve performance. By ensembling the previous mentioned models with a simple weighted average we seek to reduce over-fitting. While some models will tend to price slightly higher than optimum, others will price slightly lower due to different feature weights in each of the models. Therefore by ensembling the models we seek to average out any prediction errors made by any single model. In the weighted average, the weights also serve as hyperparameters which need optimization in order to create the best possible model.

6 Results

Our model goes through four iterations[10] in order to optimize performance. First of all, the model was run without hyperparameter optimization in order to determine a baseline. Secondly, the model was run with optimized hyperparameters on the same data set in order to determine the impact of hyperparameter tuning. The third run was used to optimize ensemble weighting on a validation set. Finally, with optimal hyperparameters for all parts of the model, it was trained on 4/5 of the data and used to predict the last 1/5, where the final model performance was evaluated. Root Mean Square Error (RMSE) is used as a measure of performance of the model.

$$RMSE = \sqrt{\frac{\sum_i^n = 1(\hat{y}_i - y_i)^2}{n}} \qquad \text{(RMSError)}$$

RMSE is an expression of the standard deviation of the residuals. This will give a measure of how far the estimates are from the actual observations in total.

In order to illustrate the impact of hyperparameter optimization, the data set is split into three categories. A training data set (70 percent of data), a validation data set (10 percent of data) and a test data set (20 percent of data). This split is used for the first three rows in Table 3, whereas the final evaluation consists of the training data set (80 percent of data) and a test data set (20 percent of data).

6.1 Model performance

We see that model performance for Lasso and ElasticNet is fairly poor before hyperparameter optimization, yet slightly better than the standard deviation of 439.45, indicating the model is at least more useful than simply guessing the mean every time. Model performance is then evaluated on the same data set, the only difference being that hyperparameters have been tuned. The second row clearly demonstrates the impact of the tuning and clearly increases model performance. Optimization especially impacts Lasso and ElasticNet, but GBR and XGB also show improvement.

In order to squeeze slightly more performance out of the model, the ensembling consisting of a weighted average of the different models is optimized on a validation set. Here the weights are set to minimize RMSE on that data set. The final weights were determined to be 10 percent Lasso, 10 percent ElasticNet, 40 percent GBR and 40 percent XGBoost.

We see that it is possible to further increase model performance slightly with ensembling,

[10] Implementation of PolynominalFeatures was attempted, but due to dimensionality issues such an implementation is limited given current computational constraints. Furthermore, Principal Component Analysis was implemented and the number of components varied as a hyperparameter, but had a negative impact on model performance for all values.

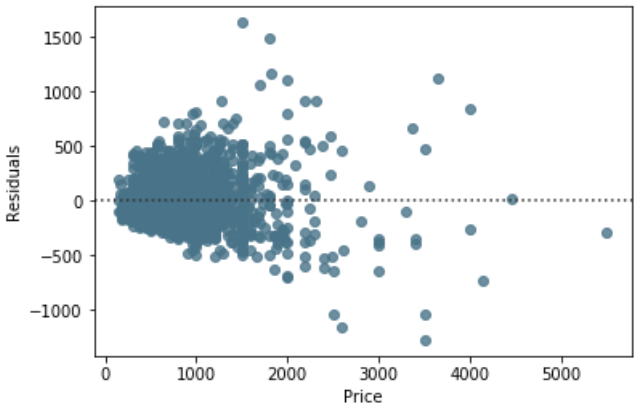
even though the change is not as drastic as with hyperparameter optimization, i.e. the best single-model RMSE is at 237.45 and the ensembled model scores 235.78.

Table 3: Prediction models: RMSE

Numbers as RMSE	Lasso	ElasticNet	GBR	XGB	Ensemble
Pre-Optimization (Only Test)	407.60	407.60	243.49	244.48	N/A
Post-Optimization (Only Test)	245.74	245.73	240.88	240.77	N/A
Post-Optimization (Only Val)	268.47	267.54	243.26	244.83	242.49
Final Model (All Test)	245.42	245.68	237.45	239.88	235.78

Graphically, the residuals are plotted in Figure 10. In a scatter plot, it is more clear to see the variation that remains in the residuals.

Figure 10: Residual plot



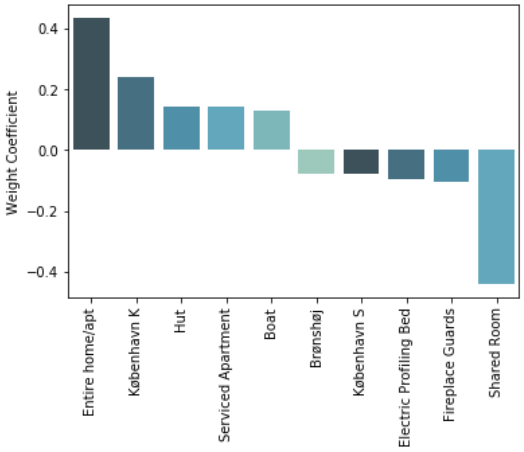
We see that the model is fairly accurate for listings in the price range of 200 to 1500 DKK, but suffers quite extensively beyond that range. The listings priced above 1500 DKK make up approximately six percent of the total. With regards to the features that were constructed during the data analysis, we found that distance to transport and sentiment scoring of the *summary* had very little impact on the model results. For the Lasso model, we see the features that have the largest importance from Figure 11a.

It is expected that the dummy indicating that the listing consists of an *entire home/apartment* has a very positive weight, as well as the *zip code* for the centre of Copenhagen. We also find that a *shared room* has a very negative impact on price, unsurprisingly. While the *fireplace guards* has a negative weight, only 19 listings have this feature, making it less important in the grand scheme of things.

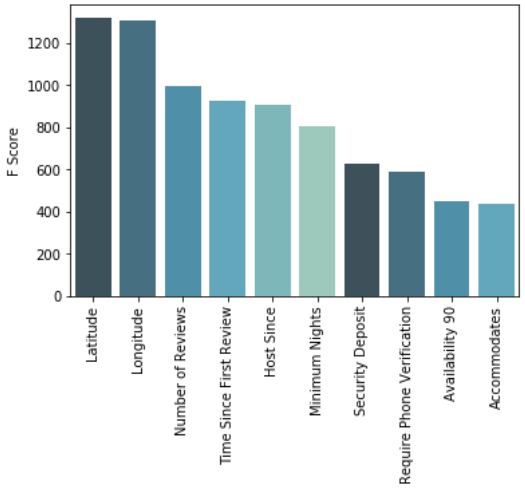
Due to the decision tree nature of the model, it is possible to segment latitude and longitude

Figure 11: Model features

(a) Most important Lasso weights



(b) Splits made by XGBoost



into smaller areas than what is given by *zip code* and other distance metrics, enabling price premiums to be weighed more accurately. The model has also placed a clear significance on host and review parameters, as well as *accommodates* as a proxy for size.

As the models do not place the same importance on the features, it perhaps explains the ensembling result as their strengths can be complimentary[11]. XGBoost the number of splits made for that given feature as seen in Figure 11b.

6.2 Critique of the model

Our main point of critique has been the endogeneity of the data, but this does not apply to the model since it simply predicts from the data given. The model could be improved by even more insightful data.

We also see that there are features missing which could improve the model further. An example could be a human measurement of *character* as a variable stating the feel of the apartment. An example could be, that an apartment in a building facing the lakes. The apartment facing the lakes will have a higher price, than the apartment not facing the lakes all other features equal. We are not able to catch differences like these, giving us a less precise model. The same argument can be given about the rest of the available data that we have not been able to process, like pictures of the apartment and/or hosts, geographical price-clustering etc. Furthermore, the way

[11] As Lasso and ElasticNet are quite similar as well as GBR and XGBoost, we have chosen only to include parameters from one of each.

we structured data, filters out hosts with less than two reviews, meaning the price estimation is based on hosts with some experience.

Our model also suffers from the fact that it is being fed cross-sectional data. As earlier mentioned this means, that the price is measured for a single day. This day is in the high season, and this gives an upward bias to the price. It could also be that other things will have an effect on many or even all of the listings. The same goes for special events that will give a demand boost; these can drive prices up, but will not be caught in our model.

7 Discussion

In the following section we seek to discuss the context of a pricing model, the impact of changing market dynamics and further work. The model results, as earlier stated, is a fair estimation of price with the given inputs. It takes many obvious features into account, but also relatively easy accessible computed features which can have an effect on listing's price. It has a clear scope of how the data is used, and does give an estimation of the true price. We now have a transparent model to give an estimated price for every Airbnb host, even though it suffers from endogeneity.

The relevance of transparency seems high when you look at recent highlights from Airbnb's past years. Airbnb has already been sanctioned by EU over their unclear price showcasing from the customer side [17]. They also plan an initial public offering in 2019, and therefore might look to boost revenues [18] in order to increase their valuation further. Furthermore, a new price suggestion algorithm has been implemented throughout the last years [19]. All these cases point towards a need for a transparent price suggestion.

7.1 Supply-side Dynamics

This model could offer (with the optimal data input) clear information for the suppliers in the peer-to-peer home sharing market. This would cause the issue with asymmetric information to be resolved. This way, the market power would not any longer be with the third party that is Airbnb, but a more optimal equilibrium between the supply and demand.

If the model is offered to all hosts (suppliers) and used by many we are likely to see a new price evaluation from the supply side of the market. Under full information we expect to find a higher price for many hosts. Basic economical theory implies that this will cause the hosts to set the price higher – which will decrease the demand, but attract more hosts and thereby converge to a new equilibrium. Even though this is basic theory, the actual effect on the supply is determined by rigidity of prices and the elasticity of the supply.

One essential part of the equation is by how much the supply will react to the new and higher observed price. The suppliers price elasticity therefore determines how many new suppliers will enter the market, and thereby also determine the new prices. This will also vary across host types, since professional hosts will be more inelastic than people renting out their own homes, due to the opportunity cost affiliated with each type. Furthermore the rigidity of price will also determine the impact. The case where few first movers set the price higher, the demand will fall dramatically for these few, and they are therefore likely to not increase the price permanently. There the price will also be determined be the price evaluation of each host. Another hypothesis is that the model can break down entry barriers by informing people about the opportunity cost by not renting out their apartment on Airbnb. The model will make entry to the market more easy and convenient. This may cause additional suppliers to enter the market, lowering the price even further.

7.2 Demand-side Dynamics

Demand is not directly affected by adjusting the asymmetric information relationship, but will react to a potential change in supply. To interpret the changes to the demand it is essential to discuss the elasticity, which depends on the type of agents in the market. If tourists are looking at Airbnb listings, they will have many possible substitutions e.g. hotels or even a new destination. The substitution possibilities will therefore determine a relatively elastic demand, meaning it is likely that the demand will fall due to a increase in price.

7.3 Externalities

Airbnb has become the face of an increase in tourism and the externalities that follow. While some are positive, such as giving citizens the opportunity increase their income by renting out their home. It is also positive that tourists stay longer in Airbnb homes than in hotels. Yet a great deal of criticism has also come along as well. Negative externalities such as increased noise, decreasing neighbourhood cohesion, gentrification and a decrease in hotel worker wages[21] have been observed.

As pointed out by Gravari-Barbas and Guinand Sandra in their paper *"Tourism & gentrification in contemporary metropolises, international perspectives"* [20] fluctuations in prices for short term renting could cause fluctuations in prices of the real estate market, since the supply for long term renting is different from the short term market. Whether this is positive or negative depends on the eye of the beholder, but it can be problematic when residents become Airbnb hosts in order to pay their rent, further increasing property values, participating in a vicious circle.

By influencing the market dynamics of the home-sharing market externalities, positive as well as negative may be scaled according to market activity. Some externalities such as tax evasion

can be mitigated, e.g. with the Danish tax deal, while others do not have a simple solution and require a political assessment before further action can be taken.

7.4 Further work

In order to further determine whether or not Airbnb is utilizing the asymmetric market information, more data gathering and analysis is required. As the Airbnb model varies price over a relatively short time in order to adjust for e.g. weekend vs. weekday demand while non-algorithm based pricing is stable, it would be possible to separate the two, therefore removing the before mentioned endogeneity. Furthermore the pricing model still has significant optimization potential, e.g. implementing scoring of images as well as further feature engineering.

With the rise of the sharing economy alongside the introduction of more and more complex pricing models, we believe this market dynamic may repeat itself in multiple markets where a third party facilitates trades between two individuals. This could be relevant to delve into in further detail.

8 Conclusion

The paper set out to answer whether or not it was possible to construct an Airbnb pricing model from publicly available data. We found that it indeed is possible for the Greater Copenhagen area and with a RMSE of 235.78 the model is significantly better than simply guessing the mean of 493.45. We managed to improve model performance quite significantly with the use of hyperparameter optimization as well as ensembling. The model was not entirely accurate and could have included more data sources such as imagery and seasonality in order to be useful for prolonged periods of time instead of only this cross-section. We also saw that our own constructed features did not play a central role in the model, meaning the price decision process is not easily predicted.

We discussed whether or not the Airbnb-market suffered from a principal-agent problem due to asymmetric information between Airbnb and the host. Furthermore we touched upon if Airbnb was incentivized to anchor hosts on lower prices in order for Airbnb to further their profit, especially considering their current time-line in regards to their initial public offering. This serves as a stepping stone for further research as the current endogeneity in the data due to influence from Airbnb's own prediction model prevents further investigation. The externalities of Airbnb were also briefly discussed as this is a key subject for potential policy intervention.

9 Bibliography

References

[1] Trefis Team: *As A Rare Profitable Unicorn, Airbnb Appears To Be Worth At Least $38 Billion.* Forbes, May 2018.

[2] Michael Olsen & Rasmus Straka: *Horesta: Nu er der flere Airbnb-senge end hotelværelser i Danmark.* Politiken, March 2017.

[3] Steven D. Levitt & Chad Syverson: *Market Distortions when Agents are Better Informed* National Bureau of Economic Research, January 2005

[4] Amos Tversky & Daniel Kahneman: *Judgment under Uncertainty: Heuristics and Biases* Hebrew University, Israel 1974

[5] Finn Årup Nielsen: *Evaluation of a word list for sentiment analysis in microblogs.* Danish Technical University, March 2011

[6] Beredskabsstyrelsen: *Undersøgelse af forekomsten og effekten af røgalarmer 2009.*

[7] Sebastian Raschka & Vahid Mirjalili: *Python Machine Learning.* Packt Publishing.

[8] Horn, K., & Merante, M. (2017): *Is home sharing driving up rents? Evidence from Airbnb in Boston.* Journal of Housing Economics, 38, 14-24.

[9] Tim Worstall: *What A Surprise, AirBnB Chooses Dublin As European Headquarters, Here Comes The 2% Tax Rate.* Forbes, Sep 2013.

[10] Emil Gjerding Nielson: *Airbnb to report homeowners' income to Danish tax authorities.* Reuters, May 17th 2018.

[11] Christian Szegedy Sergey: *"Batch Normalization".* Cornell University, 2015.

[12] Pedregosa et al.: *Scikit-learn: Machine Learning in Python.* JMLR 12, pp. 2825-2830, 2011

[13] sklearn: *Documentation, linear_model.ElasticNet.* Accessed 28/08/18.

[14] sklearn: *Documentation, ensemble.GradientBoostingRegressor.* Accessed 28/08/18.

[15] Hastie et al: *The Elements of Statistical Learning.* 2nd edition, Springer, 2009.

[16] Tarik Dogru & Osman Pekin: *What do guests value most in Airbnb accommodations? An application of the hedonic pricing approach* Boston Hospitality Review, 01 June 2017.

[17] EU Commission Press Release: *EU consumer rules: The European Commission and EU consumer authorities push Airbnb to comply* EU, July 2018

[18] Ingrid Lunden & Romain Dillet: *Airbnb aims to be 'ready' to go public from June 30, 2019* Techcrunch, June 2018

[19] Dan Hill (previous Airbnb employee): *The Secret of Airbnb's Pricing Algorithm* August 2015

[20] Gravari-Barbas M. & Guinand Sandra: *Tourism & gentrification in contemporary metropolises, international perspectives* Abingdon, Oxon New York, NY: Routledge is an imprint of the Taylor & Francis Group, an Informa Business, 2018.

[21] Shirley Nieuwland and Rianne van Melik: *Regulating Airbnb: how cities deal with perceived negative externalities of short-term rentals* Current Issues in Tourism, Routledge, July 2018

[22] Tianqi Chen, Carlos Guestrin: *XGBoost: A Scalable Tree Boosting System* KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016