Norwegian School of Economics

Bergen, Spring 2023

Evaluating the Impact of Image Features on Airbnb Price Predictions

A Machine Learning Approach to Hedonic Pricing

Sjur Gobeil Garcia

Supervisor: Mateusz Mysliwski

Master thesis, MSc in Economics and Business Administration,

Major: Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Abstract

This thesis explores the influence of image features on the predictive performance of hedonic price models for Airbnb listings. By integrating machine learning methods, image quality features, colour features, and black-box model interpretation methods, the study demonstrates the value of these components in the field of property price prediction. This thesis utilizes a novel dataset scraped in 2023 from Amsterdam which offers updated insights into the role of image features in Airbnb pricing. After deploying 10 different machine learning models, the XGBoost model yields the best predictive accuracy based on several performance metrics. Although the enhancement in predictive performance of the XGBoost model by inclusion of image features was not statistically significant, these features showed non-negligible influences and interactions in the decision-making process of the model. These findings imply a potential role of image features in refining property price models, providing valuable insights for the stakeholders in the fields of hospitality, real estate, advertising, and machine learning research.

Acknowledgements

I would like to express my gratitude to my supervisor, Mateusz Mysliwski, who has assisted me throughout this process. Mateusz's extensive knowledge and insightful perspectives have been instrumental in shaping this research work. His readiness to answer my queries, provide detailed explanations, and offer constructive criticism has been invaluable to me.

Contents

Contents

ABSTRACT **2**
ACKNOWLEDGEMENTS **2**
CONTENTS **3**
1. INTRODUCTION **5**
2. LITERATURE REVIEW **7**
3. THEORY **10**
3.1 MACHINE LEARNING 10
3.2 TRANSFER LEARNING 11
3.3 PERFORMANCE METRICS 12
3.4 INTERPRETABLE MACHINE LEARNING METHODS 13
3.4.1 Permutation Feature Importance *13*
3.4.2 Accumulated Local Effects *14*
3.4.3 H-statistic *15*
3.4.4 Shapley Values *16*
3.4.5 Local Interpretable Model-Agnostic Explanations *17*
4. DATASET **18**
4.1 CLEANING AND PRE-PROCESSING 18
4.2 IMAGE DATA 19
4.2.1 Blind/Refereneless Image Spatial Quality Evaulator *19*
4.2.2 Image labels *20*
4.2.3 Hue, Saturation & Value *22*
4.2.4 Clarity *23*
4.2.5 Preprocessing of Non-Image data *23*
4.2.6 Near-zero variance filter *29*
4.2.7 Normalization *29*
5. MODELS **30**
5.1 TUNING HYPERPARAMETERS 30
5.2 MODEL RESULTS 32

5.3 INTERPRETING THE EFFECT OF IMAGE FEATURES 35

5.3.1 Permutation Feature Importance 35

5.3.2 ALE 36

5.3.3 H-Statistic 38

5.3.4 Shapley Values 39

5.3.5 LIME 41

6. LIMITATIONS AND FURTHER WORK 43

7. CONCLUSION 45

REFERENCES 46

APPENDICES 51

1. Introduction

In the data-driven era, advancements in computational power and machine learning algorithms have revolutionized decision-making processes across various sectors, including the hospitality industry. One significant player in this realm is Airbnb, a platform that connects individuals offering accommodations (hosts) with those seeking them (guests). An essential aspect of a listing's success on Airbnb is its pricing, which underlines the necessity for accurate price prediction models.

The driving force behind Airbnb price prediction models is to achieve a balance in the rental market. Properties priced too high risk being left vacant, as potential tenants are deterred by the inflated costs. Conversely, underpricing a property can results in significant potential earnings being left on the table. Effective price prediction models can assist in mitigating these risks, helping to maintain a stable and fair rental market.

Existing research on Airbnb price prediction has predominantly focused on conventional predictors, such as location, amenities, reviews and host-related features. With the rise of sophisticated image analysis techniques, there is an intriguing opportunity to investigate whether image-relate features can enhance the accuracy of these models. Images, often being the first point of interaction between potential guests and listings, play a substantial role in shaping perceptions and decisions.

The traditional methods used to reveal important features in estimating property prices have been hedonic pricing models. The hedonic pricing theory suggests that a good or product in itself does not deliver utility. Instead, it is composed of characteristics that each contribute some level of utility. As per this framework, the market price paid by a consumer for a particular product is linked to the utility derived from these various characteristics (Lancaster, 1966). In the context of Airbnb listings, each offering does not provide utility in itself, but rather, its components or features- such as location, amenities, and potentially images- each contribute to its overall utility. Thus, the listing's price can be tied to the utility these features provide. Hedonic models are designed to quantify individual characteristics while maintaining interpretability. As a result, most research related to hedonic pricing employs simple parametric models that work with structured, conventional data. An excellent illustration of

such a model is Ordinary Least Squares (OLS) regression, as it enables the determination of attribute-specific prices simply by examining the regression coefficients. Interpreting these coefficients might lead to deceptive results. Furthermore, because machine learning methods are constructed on complex, nonparametric models, they can assist in identifying new features and exposing potential nonlinear dependencies that typically go unnoticed in traditional hedonic approaches. The application of machine learning models no longer necessitates them being treated as "black boxes". I perceive this study as part of the modernization of hedonic methodology, forming a connection between two disparate research groups.

In this thesis, I critically examine the hypothesis that image features can improve the predictive performance of Airbnb price prediction models. I employ different machine learning models and evaluate the impact of various image features alongside traditional ones. I also leverage model interpretability methods for black-box models to discern the contribution and importance of each feature in the prediction process.

This study embarked on an exploration of how image-related features might influence and potentially improve the accuracy of price prediction models, specifically focusing on the Airbnb platform. The motivation behind this research was driven by the intersection of technological curiosity and a clear business imperative: to uncover valuable insights that could streamline pricing strategies within the hospitality industry.

The findings of this thesis hold relevance for multiple stakeholders within the Airbnb ecosystem. For hosts, it provides an understanding that factors other than image-related attributes are likely to be more influential in determining optimal pricing. For guests, it underscores the fact that image-related features of a listing, while important for visual appeal and information, are not necessarily that indicative of price variations. For Airbnb and similar platforms, it helps in setting realistic expectations about the utility of image analysis in their predictive modelling and pricing algorithms.

This research aims to investigate the impact of image-related features on Airbnb price predictions by employing machine learning models and interpretation methods to explain the extent to which these features contribute to the accuracy and understanding of the prediction models. Motivated by this, I will answer the following problem formulation:

How do image features affect the predictive performance of hedonic price models for Airbnb listings?

The main finding of this study is that, to a modest extent, the incorporation of image features enhanced the predictive performance of the top-performing model (XGBoost) when adding the image features. Nevertheless, the robustness of this finding was challenged by an unsuccessful significance test. On the other hand, certain image features consistently emerged across various interpretative methods used for the black-box model, suggesting their valuable contribution to the model's performance.

The rest of this paper will be structured as follows. First a literature review will be presented containing the most relevant literature for this thesis. Secondly, a theory section will be dedicated to describing the different methods used for the analysis. Subsequently, the data will be presented along with the steps taken to prepare the data for the models. After this, the results of the models will be showed. The models will include several variables that are assumed to be important price predictors. Then new models will be made where the image features are removed to compare results. Next, different interpretation methods will be applied to the best performing model. Finally, A conclusion of my findings and suggestions for further research will be proposed.

2. Literature Review

This thesis intersects multiple strands of literature, providing valuable contributions across diverse fields. Primarily, it engages with the extensive body of research exploring the determinants of Airbnb pricing, offering fresh insights that can stimulate further studies within the hospitality and real estate sectors. Additionally, this thesis enriches the current understanding of the predictive capacity and effectiveness of machine learning models for property pricing. It does this by offering nuanced insights into their practical applications and potential implications. By integrating image recognition to understand the role of image features, this work contributes to the growing body of computer vision. Moreover, its findings on the significance of visual elements in pricing strategies have implications for marketing and advertising fields, thereby expanding its relevance beyond traditional boundaries.

A range of studies have drawn attention to the potential benefits of machine learning methods traditional hedonic pricing models. Moreno-Izquierdo et al. (2018) demonstrated the potential advantages of AI and machine learning methods over traditional hedonic pricing models with Airbnb data about the Valencian Community. They found that the application of neural networks led to more satisfactory price estimations than traditional pricing models. They also

found that listings with more photos tend to have higher prices. In their linear hedonic pricing model, Dogru and Pekin (2017) also find that the number of photos is valued by guests. Although not used in the context of Airbnb, Potrawa and Tetereva (2022) integrate machine learning tools to enhance conventional hedonic pricing models for houses in Rotterdam. They find that explainable AI methods can be used for black-box models and uncover nonlinear relationships between some predictors and the housing prices. This is something that traditional hedonic pricing models cannot do since they fit linear models with Ordinary Least Squares Regression.

Poursaeed et al. (2017) use crowdsourcing to categorize luxury levels of real estate photos and showed that it can improve price estimates with neural networks. Ahmed and Moustafa (2016) use Speeded Up Robust Features extractor to extract visual features from images. They then used Support Vector Machines and Neural Networks to improve the estimation of house prices.

Zhang et al. (2022) delve into the composition of a high-quality image, drawing from photography literature to identify 12-human-interpretable image attributes pertaining to composition, color, and figure-ground relationship. Their analyses establish systematic differences between images that have Airbnb verification symbols and those that do not, and they predict how each attribute correlates with property demand, finding significant correlations in the theorized direction. The authors used difference-indifference (DiD) analysis and deep learning to discover an increase of 8.98% in occupancy rates for properties where pictures were taken by professional photographers, compared to the properties where the pictures were taken by hosts. Inspired by this paper, I will include a four of the same image attributes in my models, but instead of using DiD method, I will use machine learning models. I am limited to only use for of the same attributes because these are the only attributes that are within the computational limits of my resources.

The same authors wrote a different paper where they found that having verified photos led to spillover effects. Namely, listings with verified photos in a neighbourhood led to a higher demand for other verified listings in the same area, while unverified listings in these neighbourhoods experienced a decrease in demand. This suggests that a high proportion of verified listings in a neighbourhood can elevate the neighbourhood's overall image, increasing its attractiveness to potential renters (Zhang et al., 2016).

Nguyen et al. (2018) attempt to comprehend how potential Airbnb guests derive first impressions from listing images by predicting human impressions of ambiance from listing photos. They used crowdsourcing to annotate the images on various physical and ambiance attributes. They found that they could best label their images using GoogLeNet convolutional neural network trained on the Places205 dataset, a large collection scene-centric image. In this thesis, I will employ the ResNet18 model instead. Fagerstrøm et al. (2017) examined the influence of the personal profile images of hosts, specifically their facial expressions, on buyer behavior in Airbnb. They found that hosts that had positive facial expressions in their profile pictures increased approach behavior and rental likelihood. On the other hand, negative facial expressions or the lack of a profile picture altogether reduced demand for their listings. Furthermore, Ert et al. (2016) found that trustworthiness inferred from a host's photo significantly impacts the guest's decision making and that it matters more than review scores. These papers support the idea that images on a host's profile do matter for price setting.

Kalehbasti et al. (2019) made Airbnb prediction models with a variety of features, including rental characteristics, owner information, and customer reviews. Their objective was to create the best models they could based on several performance metrics. They tested several methodologies for creating the prediction model, ranging from linear regression and tree-based models to more complex techniques such as Support Vector Regression and neural networks. They found that having an abundance of features led to high variance and weakened model performance on the validation set compared to the training set. However, applying a Lasso-based feature selection technique reduced this variance. Their best performing model was an Support Vector Regression model when estimating performance based on R^2 and Mean Square Error.

Luo et al. (2019) developed several machine learning models to predict Airbnb listing prices across three cities, New York, Berlin, and Paris. In their case, it was XGBoost and Neural networks that performed the best in terms of R^2 and Mean Square Error. Particularly notable from their research was that training the models on a combined dataset from New York and Paris was able to generalize and predict prices in a different city, Berlin, respectively. In fact, this transfer learning technique yielded better price predictions in Berlin than when they trained the model only using the Berlin dataset. Transfer learning will be incorporated in this paper as well but with the goal of labeling Airbnb images.

While the individual methods and techniques applied in this thesis may not be entirely novel, the uniqueness lies in their integrated application. This research combines the utilization of image quality features and colour features within the context of machine learning framework. Furthermore, it employs advanced black-box interpretation methods to explain the feature contributions to the prediction model. Significantly, the study applies these combined methods to a completely new and contemporary dataset, specifically Airbnb listings scraped in 2023 in Amsterdam.

3. Theory

3.1 Machine learning

Defined broadly, a machine learning algorithm is a process that improves its ability to perform tasks over time by learning from its experience with data (Goodfellow et al., 2016). This thesis will leverage the capabilities of ten distinct supervised machine learning models to predict Airbnb prices for various listings. Hence, a brief description of the types of models used will be explained.

Linear models, including Linear Regressions, Lasso, Ridge, and Elastic Net, represent a fundamental class of predictive models that assume a linear relationship between the dependent variables and the parameters or coefficients of the independent variables in the model. The simplest of these, Linear Regression, serves as a foundation for understanding statistical relationships and is often a first line approach due to its interpretability. The more sophisticated extensions, Lasso, Ridge, and Elastic Net, introduce regularization terms to the loss function in order to manage overfitting and multicollinearity.

Tree based models, including Decision Trees, Random Forests, and XGBoost, offer an intuitive way of capturing non-linear relationships and interactions between variables. Decision Trees split the data along the features to segregate the target variable into its most distinguishable states. However, they can easily overfit the training data. To mitigate this, Random forests use an ensemble of different decision trees, built on different subsamples and subsets of features, to ensure generalizability and robustness. XGBoost further refines this concept by applying a gradient boosting framework, optimizing for both model performance and computational efficiency.

Models such as K-Nearest Neighbours (KNN) and Support Vector Machines (SVM) leverage the geometric properties of the feature space for their predictions. KNN, as a distance-based method, predicts the outcome for new instances by examining the outcomes of its nearest neighbours in the feature space. The prediction is then made based on the average outcome of these nearest neighbours. In the context of regression, SVM operates by constructing an optimal hyperplane within the feature space that can best predict continuous outcomes. The objective of SVM is to fit the best hyperplane that minimizes the error between the predicted and actual values, often resulting in a robust model with considerable prediction accuracy. In constructing this hyperplane, SVM mainly considers those instances in the data that are hardest to accurately predict, known as "support vectors". By focusing on these challenging instances, SVM aims to maximize generalization performance, thereby yielding a model that is robust to variability in data (Wilimitis, 2021).

Artificial Neural Networks, inspired by the biological neural networks that constitute the human brain, are a powerful tool for modelling complex patterns and high-dimensional data. They consist of interconnected layer of nodes or "neurons" where each connection can transmit a signal from one neuron to another. The receiving neuron processes the signal and signals downstream neurons connected to it. Neural networks' capacity to learn from errors, model non-linear relationships, and handle large-scale data make them particularly effective in various prediction tasks.

3.2 Transfer Learning

This thesis uses transfer learning to label the images, which will then be used as predictors in the models. Consequently, a clarification of what transfer learning entails will follow in this section of the thesis.

Transfer learning is a machine learning technique that involves using a pre-trained model as a starting point for a new task, rather than training a new model from scratch. Transfer learning has become increasingly popular in recent years, particularly in the field of computer vision. In transfer learning, a pre-trained model is typically trained on a large dataset, such as ImageNet, which contains millions of images across thousands of classes. The pre-trained model learns to recognize a wide variety of visual patterns and features that are relevant to many different tasks. To use transfer learning, the pre-trained model is typically modified by replacing the final layer(s) with a new layer(s) that is specific to the new task.

Transfer learning has several advantages over training a new model from scratch. First, transfer learning can significantly reduce the amount of data and computational resources required to train a new model, as the pre-trained model has already learned to recognize a wide variety of visual patterns and features. Secondly, it often results in improved performance on new tasks, as the model has already developed an understanding of many relevant visual features that can be utilized in new contexts. Lastly, transfer learning can help to mitigate the problem of overfitting. This is because pre-trained models, having been trained on large datasets, are already equipped with the ability to generalize to unseen images effectively.

3.3 Performance Metrics

To evaluate how the image features affect the predictive performance of hedonic price models, we need to quantify the model performance. Mainly, this will be achieved by comparing the models based on three performance metrics, which are commonly used in the related literature.

Root Mean Square Error (RMSE) is a common metric used to measure the error of a prediction model of a continuous variable. It is defined as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{i_i} - \hat{y})^2} \tag{1}$$

RMSE is a metric that measures the average magnitude of prediction error in the mode, essentially determining how far off the model's predictions are from the observed values. It does this by calculating the square root of the average squared differences between the predicted and observed values. A lower RMSE indicates a better fit of the model to the data, as the predicted values are closer to the actual ones (Hodson, 2022).

Mean Absolute Error (MAE) is defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2}$$

MAE is calculated as the average of the absolute differences between the predicted and observed values, because it uses absolute value of the difference, the MAE does not heavily penalize large errors, making it more robust to outliers compared to RMSE. A lower MAE indicates a better model performance, implying the predictions are closer to the observed values.

R squared is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (3)$$

R^2 is a statistical measure that represents the proportion of the variance for a dependent variable that is explained by the independent variables in a regression model. R^2 ranges from 0 to 1, with 0 indicating that the model explains none of the variability of the response data around its mean, and 1 indicating that the model explains all the variability of the response data around its mean. In other words, a higher R^2 means our model fits our data better.

3.4 Interpretable Machine Learning Methods

It is possible to expand our understanding of the effect image features have on the machine learning models' precision beyond just the performance metrics. Specifically, we can expand our understanding of the effects image features might have on the predictions if we explore which predictors matter in making the predictions. Consequently, the methods used to interpret the impact of the predictors will be presented in this part of the thesis.

3.4.1 Permutation Feature Importance

The Permutation Feature Importance (PFI) quantifies the significance of a feature by assessing the rise in the predictive error of the model when the feature is randomly shuffled. The underlying principle here is that, if an important feature's values are randomly reordered in the training set, it would disrupt the inherent relationship between the feature and the target variable, thus leading to a deterioration in the model's performance. Essentially, this approach hinges on the comparison between a baseline performance metric (such as RMSE) and the

metric obtained post permutation of a particular feature's values within the training dataset (Boehmke & Greenwell, 2019). Fisher et al. (2019) provides a model-agnostic approach to computing the feature importance and serves as an expansion of the method proposed by Breiman (2001).

The inputs of the model are the trained model $\hat{f}$, the feature matrix x, the target vector y, and an error measure $L(y,\hat{f}(X))$.The steps to retrieving the PFI scores are as follows:

1. Compute the original error of the model $e_{orig} = L(y,\hat{f}(X))$
2. For each feature j in the matrix:
   - A new feature matrix X_{perm} is created by randomly shuffling the values of the current feature j in the data X, effectively disrupting the relationship between this feature and the actual outcome y.
   - The error e_{perm} is estimated based on the predictions on this permuted data so $e_{perm}= L(Y,\hat{f}(X_{perm}))$.
   - The permutation Feature Importance is calculated as either the quotient $FI_J = e_{perm}/e_{orig}$ or the difference between the permuted error and the original error $FI_J = e_{perm}-e_{orig}$.
3. Finally, features are ranked in descending order of their Permutation Feature Importance. The order signifies the relative importance of each feature in the trained model.

3.4.2 Accumulated Local Effects

Accumulated Local Effects (ALE) is a technique used to explain the predictions of any machine learning model. It measures the main effects of the features, meaning the effect of each feature after accounting for the average effects for all other features (Kim, 2022). Molnar (2023) proposes this equation for computing ALE:

$$\hat{\tilde{f}}_{j,ALE}(x) = \sum_{k=1}^{k_j(x)} \frac{1}{n_j(k)} \sum_{i:x_j^{(i)} \in N_j(k)} \left[\hat{f}(z_{k,j}, x_{-j}^{(i)} - \hat{f}(z_{k-1,j}, x_{-j}^{(i)}\right] \quad (4)$$

1. First, the range of the feature is partitioned into many intervals. These intervals are usually determined by the quantiles of the feature's distribution.
2. Next, for each interval z, the differences in the predictions are calculated, as represented in the brackets of the equation. Specifically, the feature of interest j in the

data instance i is replaced by the upper bound $z_{k,j}$ and lower bound $z_{k-1,j}$ of the interval, leaving other features x_{-j} constant. The difference in predictions gives us the local effect of that feature for the instances in the interval.

3. Thirdly, the average of the local effects across all instances in each interval are computed. This means adding up all the effects and dividing the number of instances in the interval. $N_j(k)$ represents the set of instances for which the *j*-th feature falls within the *k*-th interval.
4. Furthermore, the averaged effects across all intervals up to each point in the feature's range are accumulated. This results in the uncentered ALE at that point. If a feature value lies in the third interval, for example, its uncentered ALE would be the sum of the effects of the first, second, and third intervals.

The resulting ALE of a feature at a certain value can be interpreted as the main effect of the feature at that value compared to the average prediction of the data. For example, an ALE estimate of -1 at x_j =2 means that when the j-th feature has a value of 2, the prediction is lowered by 1 compared to the average prediction.

3.4.3 H-statistic

The H-statistic measures interaction effects of the features. It can be used to measure the interaction effects of one feature with all other features or it can measure pairwise interaction effects. The former method will be used in this thesis, which is originally introduced by Friedman and Popescu (2008) and later adapted by (Molnar, 2023). The equation for the H-statistic is:

$$H_{jk}^2 = \frac{\sum_{i=1}^{n}\left[\hat{f}\left(x^{(i)}\right) - PD_j\left(x_j^{(i)}\right) - PD_{-j}\left(x_{-j}^{(i)}\right)\right]^2}{\sum_{i=1}^{n}\hat{f}^2\left(x^{(i)}\right)} \quad (5)$$

In this equation, the prediction of the model for the *i*-th instance in the data is denoted by $\hat{f}\left(x^{(i)}\right)$. The second term represents the partial dependence of the *j*-th feature. Moreover, the third term represents the partial dependence for the other features (denoted -*j*) for the *i*-th instance. The numerator calculates the squared difference between the prediction of the *i*-th

instance and the combined effect of the partial dependences of the *j*-th feature and all other features. This essentially measures the deviation of the model's prediction from what would be predicted by considering only the partial dependences, for each instance. The denominator of the equation is the sum of the squares of the model's predictions for all instance in the data. This acts as a kind of normalization term, ensuring that the measure of dependence that is calculated is scaled appropriately.

When the value of the H-statistic is 1, it signifies that the feature's influence on the prediction solely arises through interactions, meaning it has no direct effect. Conversely, a value of 0 implies there are no interactions, suggesting that the influence stems exclusively from the main effect (O'Sullivan, 2021).

3.4.4 Shapley Values

The Shapley Value is a concept that was initially developed in the field of cooperative game theory. It provides a way to fairly distribute the gain among all players in a cooperative game based on their individual contributions. It is particularly useful in situations where the contributions of players are interdependent, and hence the total gain cannot simply be divided equally or based on individual contributions. In the context of machine learning and particularly model interpretability, Shapley Values have been adapted to measure the importance of features in a predictive model. The prediction for a particular instance can be considered as the "total gain", and the "players" are the features used in the model. The aim is to distribute the "gain" (i.e., the prediction for the instance) among the features based on their individual contributions.

However, as with the original application in game theory, calculating Shapley Values can be computationally expensive, especially when the number of features is large. This has led to the development of various approximation methods, such as the model-agnostic approach by Štrumbelj and Kononenko (2014):

$$\widehat{\phi}_j = \frac{1}{M}\sum_{m=1}^{M}((\hat{f}(x_{+j}^m) - \hat{f}(x_{-j}^m)) \tag{6}$$

1. For each iteration (m=1,…,M), a random instance (z) is drawn from the data matrix X.
2. Next, a random permutation (o) of the feature values is selected.
3. The instances x and z are then reordered according to this permutation, creating x_o and z_o.
4. Two new instances are created:
    - One with the j-th feature: x_{+j}. Here, Feature values up to the j-th feature are taken from x_o and the remaining feature taken from z_o.
    - One without the j-th feature: x_{-j}. Here feature values up to and including the (j-1)-th feature are taken from x_o, but the j-th feature is taken from z_o and the remaining features are taken from z_o as well.
5. The marginal contribution is computed as $\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$
6. The average Shapley Value is computed: $\phi_j(x) = \frac{1}{M}\sum_{m=1}^{M}\phi_j^m$ (Molnar, 2023)

3.4.5 Local Interpretable Model-Agnostic Explanations

Local Interpretable Model-Agnostic Explanations (LIME) is a local interpretation method used to explain individual predictions. Introduced by Ribeiro et al. (2016), LIME is a particular implementation of what are known as local surrogate models. Local surrogate models are simple models utilized to interpret individual predictions of more complex, black-box machine learning models. These surrogate models are trained to mimic the predictions of the original black-box model.

$$explanation(x) = \underset{g \in G}{arg\ min}\ L(f, g, \pi_x) + \Omega(g) \tag{7}$$

Here, g denotes the explanation model for a specific instance *x* and is typically a simple model such as linear regression. The model is designed to minimize the loss L, represented by a metric like RMSE, which measures how close the explanation is to the original model's prediction, denoted by *f*. At the same time, the model's complexity, $\Omega(g)$, should be kept to a minimum, which often translates to using fewer features.

G denotes the set of potential explanations, such as all possible linear regression models. The proximity measure, π_x, outlines the size of the neighbourhood around instance x that is considered for the explanation. In practice, LIME is primarily tasked with optimizing the loss part, leaving the user to determine the complexity, for instance by selecting the maximum number of features that the linear regression can utilize (Molnar, 2023). Here are the steps for training local surrogate models:

1. Choose the instance you are interested in and for which you require an explanation of its black box prediction.
2. Introduce variations into the dataset and acquire the black box predictions for these new data points.
3. Allocate weights to the new samples based on their proximity to the instance of interest.
4. Construct a weighted, interpretable model on the dataset with the introduced variations.
5. Interpret the local model to explain the prediction.

4. Dataset

The data utilized in this thesis is sourced from Inside Airbnb, a project dedicated to providing free, publicly available data about Airbnb listings globally. The specific dataset used was scraped by Inside Airbnb on March 9, 2023 and contains all of the listings in Amsterdam that were on the Airbnb websites at that time. Furthermore, the dataset only contains information that is publicly displayed, and does not contain any personal information like private host responses or exact addresses. The dataset contains 6998 observations before cleaning.

4.1 Cleaning and Pre-processing

Data cleaning and pre-processing involves identifying and handling missing values, dealing with outliers, converting data types, and creating new variables as needed. I will also explore the data to identify patterns or trends and use visualization techniques to gain insights into the data. Most of the data was retrieved as a CSV format while the geometry data used for mapping purposes was contained in geojson files.

4.2 Image Data

This thesis acquires thumbnail images of the listings by accessing the URL to the thumbnails that are present in the dataset from Inside Airbnb. Different image recognition methods are used to extract features from these images that will be used in the prediction models.

4.2.1 Blind/Refereneless Image Spatial Quality Evaulator

In this thesis, I apply the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) model, a popular no-reference image quality assessment metric, to evaluate the perceived quality of digital images. The goal of Brisque is to predict the quality of an image without comparing it to a reference image or relying on any specific knowledge of the image content or context. Brisque is based on a machine learning approach that uses a trained support vector regression (SVR) model to estimate the quality of an image. The SVR model is trained using a set of natural images and their corresponding subjective quality scores obtained through subjective judgements. The model is designed to capture statistical regularities of the image and has been shown to be effective in predicting perceived image quality (Mittal et al., 2012).

To calculate the Brisque score for an image, the image is first preprocessed by down sampling and divided into non-overlapping blocks. Then, the statistical features are extracted from each block and combined into a feature vector that represents the entire image. Finally, the feature vector is passed through the trained SVR model to estimate the perceived quality score. The output of Brisque is a quality score between 0 and 100, where higher scores indicate higher perceived quality. Brisque has been shown to be highly correlated with human perception of image quality and has been used in a wide range of applications, including image and video processing, compression, and transmission (Mittal et al., 2012).

BRISQUE scores were computed in Python, during which the images were converted into grayscale (2-dimensional) and their pixel intensity values were normalized to floating point numbers within the range [0,1]. A timer was inserted into the code to see how long it would take to complete the computations. When running the computations on the CPU the estimated time was 25 hours. Therefore, an attempt was made to run it in Google Colab's cloud services since they have publicly available tensor GPU's that can process image computations exceptionally fast when attaching the CUDA library. The code was able to successfully run until it was forcefully stopped due to my free compute units being used up after scoring 8% of

the images. However, the estimated time was 1 hour and 45 minutes for all the images when running in the cloud. The remaining brisque scores for the images were computed locally with my own GPU which did not have the same computational speed as the GPU that Google Colab had, but it was faster than running it on the CPU as the time it took for the remaining 92% of the images was approximately 13 hours.

Given that the BRISQUE model was not trained on the specific images used in this thesis, some calculated scores fell outside the typical 0-100 range. It should be noted that these scores, while not within the expected range, still provide relative measures of perceived image quality. The highest score was 117 and the lowest was -28. The images in Figure 1 and Figure 2 display the pictures with the highest and the lowest BRISQUE score, respectively.

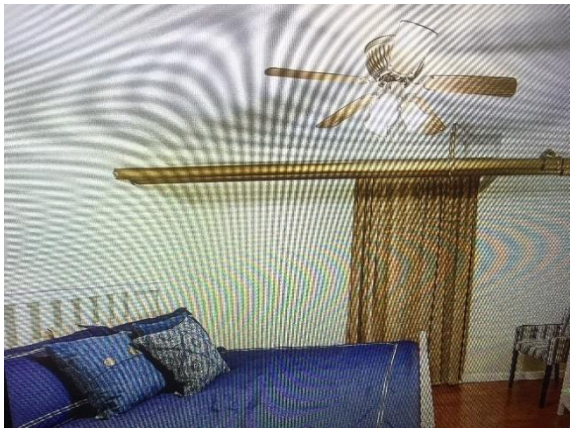

Figure 1- Highest brisque score, 117 (lowest quality)

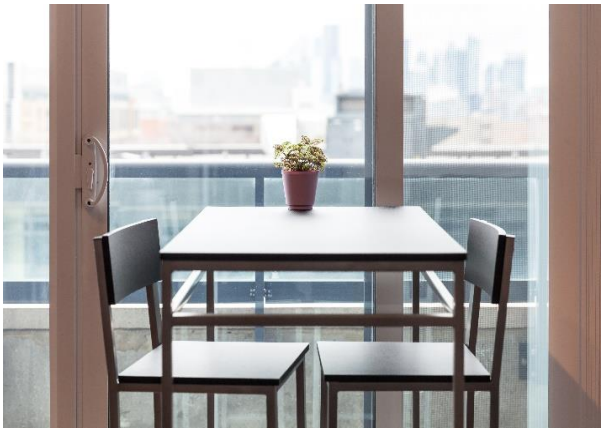

Figure 2- Lowest brisque score, -28 (highest quality)

4.2.2 Image labels

In order to label the images with scene classification, a pretrained model called ResNet18 was used. The weights for the model as well as the labels were downloaded from the Places website (Zhou et al., 2018a). The class names for the 365 scene categories were read from a downloaded text file and the images were loaded from a folder using the Python imaging Library (PIL). Then, they were converted to the expected format that was needed as input for the model. Most of the images were originally in RGB format which is the correct input format for the model, but a few were also in BGR and RGBA format and needed to be converted. The process of preparing the images also involved converting them to 256*256 pixels first, then

cropping them to 224*224. Afterwards, the images were converted to a tensor format which is needed for the neural network. Furthermore, the color channels were normalized with the means equal to 0.485, 0.456 and 0.406, and standard deviations equal to 0.229, 0.224 and 0.225 for the colors red, green and blue, respectively. These transformations are commonly used when working with the ResNet models and can improve the accuracy and performance of the model (Zhou et al., 2018b). Table 1 showcases the ten most recurrent labels, as generated by the model.

Top 10 Classes	
Class	Frequency
living_room	1052
dining_room	707
bow_window	524
bedroom	519
artists_loft	375
television_room	329
waiting_room	299
kitchen	245
hotel_room	218
patio	153

Table 1-Top image labels from the ResNet18 model

Given the similarity among some of the labels, the price prediction models might perform better if similar labels are consolidated into a single category. On the other hand, it can also lead to loss of information which is detrimental to the validity of the results. However, it can be beneficial to recode similar labels to the same label to increase the occurrences, thereby making it simpler for the models to use these as predictors. Therefore, images with the label home_theater were recoded to television_room, and hotel_room, bed_chamber and dorm_room were recoded to bedroom. Zhou et al. (2018a) supplement their study with an additional file outlining a scene hierarchy, which distinguishes whether the images are indoor or outdoor. I incorporated this dataset with the labels to examine whether this feature would contribute to better price predictions.

Moreover, the method of transfer learning employed here carries a limitation, given that I did not personally train any part of the model. An alternate approach could have involved utilizing human input to label a subset of the images, thereby allowing for the optimization of the ResNet model parameters specifically tailored to my image set.

4.2.3 Hue, Saturation & Value

Hue, Saturation and Value (HSV) is a colour model that is extensively used in computer graphics and image analysis. It offers an alternative to the more prevalent RGB and RGBA color models commonly used in digital imagery. The HSV model is uniquely designed to align more closely with human colour perception. It characterizes colours based on three distinct aspects, each with a range that can vary depending on the specific software being utilized. The definitions that I will provide are based on the OpenCV library, a popular tool for image processing in Python (Bradski, 2000).

Hue distinguishes one color from another, and it is defined within a range of [0,179]. Figure 3 visually represents the spectrum of colours as defined by their respective hues in the HSV model. “Warm hue” refers to a certain range within the colour spectrum often associated with evoking emotional responses. Warm hues, which include colours such as red and yellow, are commonly linked with increased levels of excitement. On the contrary, cool hues like blue and green tend to evoke feelings of relaxation (Valdez & Mehrabian, 1994).

Saturation describes the intensity or purity of the color. A colour at full saturation contains no elements of white or black, whereas a desaturated colour has components of white or black added to it. Essentially, a colour with higher saturation appears more vivid, while a colour with lower saturation seems grayer. Saturation in the HSV model spans a range of [0,255]. Value is also recognized as brightness and indicates the lightness or darkness of a colour. A colour with a high value is bright, whle a colour with a low vaue is darker. The brightness/value range in the HSV model is [0,255] (Faisal,2023). Figure 4 visually illustrates the interplay of the three components of the HSV model, demonstrating how hue, saturation, and value collaboratively determine a colour’s representation.

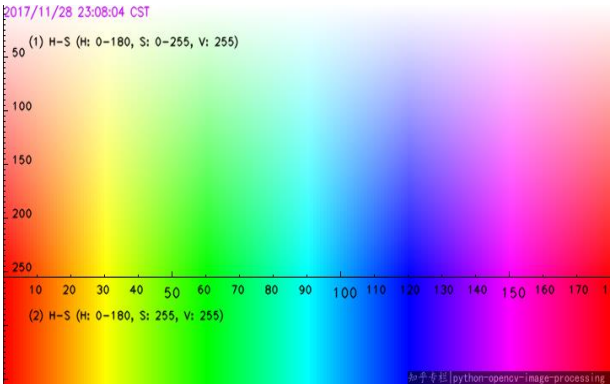

Figure 3- Hue range,

sourced from: (Choosing the Correct Upper and Lower HSV Boundaries for Color Detection With`Cv::inRange` (OpenCV), n.d.).

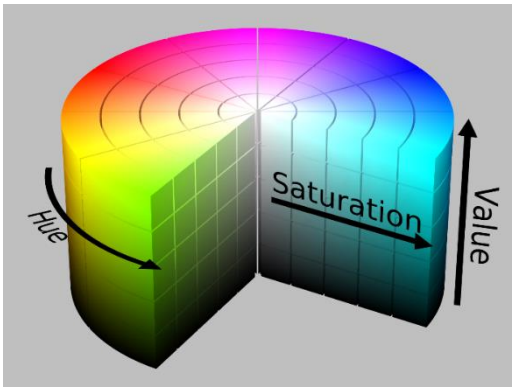

Figure 4- Colour wheel,

sourced from: (File:HSV Color Solid cylinder.png - Wikimedia Commons, n.d.).

4.2.4 Clarity

Assessing the clarity, or the sharpness, of the images in our dataset can be an interesting predictor for our study, as clearer images are generally perceived as being of higher quality and could potentially influence the price of an Airbnb listing. In order to assess the clarity of the images, I used the variance of the Laplacian operator as a metric. This method has been shown to be a reliable indicator of image clarity in previous studies (Pech-Pacheco et al., 2000).

The Laplacian operator is a second-order derivative measure, which, when applied to an image, accentuates areas of rapid intensity change, such as edges. The variance of these highlighted areas provides a scalar measure of clarity: a high variance suggests a well-focused image with clear edges, while a low variance indicates a blurry image (Sagar, 2021). This method was implemented in Python using the OpenCV library. OpenCV's efficient computation allowed for quick calculation the Laplacian and its variance for each image in the dataset. In this case, the Laplacian operator was applied after gray-scaling the images.

4.2.5 Preprocessing of Non-Image data

To provide a comprehensive comparison between models that incorporate image features and those that do not, it is essential to also consider non-image related features. These traditional features offer a basis of comparison, allowing us to understand the value added, if any, by the

inclusion of image data. In this section, the pre-processing steps taken for these features are detailed.

The ”amenities” feature contained a long string of comma-separate values for each Airbnb listing. In order to make amenities more useful for analysis, I first created a set of all possible amenities by splitting the string into individual values. Considering that there are several hundred amenities in total, only the most important ones are kept. The decision of which to keep are inspired by the word cloud of amenities in figure 5, the network graph in figure 6, and information from Airbnb Resource center (*The Amenities Guests Want*, 2020). All opening curly brackets were removed while the ending curly brackets were replaced with commas which are then used as the delimiter. Furthermore, all quotations were removed in addition to all leading whitespaces. After this remained a list of all the different amenities listed in the dataset. However, a lot of the amenities are very similar and have been converted to share the same name. For instance, “wifi” and “internet” will be interpreted as the same amenity and will be treated simply as “wifi”. The same applies for all the different descriptions of coffee machines like “coffee maker”, “espresso machine” and “espresso maker”. A few examples of amenities that were removed are “hangers”, “toilet” and “crib”. I then used the set of possible amenities to create binary variables for each amenity, indicating whether or not each listing had that amenity. For example, if an Airbnb listing had a TV, the TV variable would be assigned a value of 1 and 0 otherwise. The final amenities chosen are illustrated in table 2.

Figure 5- Wordcloud of amenities.

Amenities
wifi
air_conditioning
non_basic_electronincs
bbq
balcony
view
bed_linen
breakfast
tv
coffee
cooking_basics
white_goods
elevator
gym
child_friendly
parking
host_greeting
pool
long_stay
pets
private_entrance
safety
self_check_in
smoking_allowed
smoke_alarm

Table 2- List of chosen amenities.

In addition to the provided features in the dataset, I have also made new predictor variables derived from the 'description' field for each listing. This text field contains a detailed explanation of the property, provided by the host, and can include aspects about the property's amenities, unique selling points, nearby attractions, house rules, and more. After removing stop words like 'the', 'and', and 'in', and all non-english words, a network graph of the most used words in the descriptions was made In essence, figure 6 is a network graph of the words in the Airbnb descriptions. Each node in the graph represents a word, and each edge represents a pair of words that often occur together. The size of the nodes represent how often the words occur and the size of the link indicate how frequently the two words occur together. Moreover, it can be used to understand the main themes and topics that are mentioned in the descriptions.

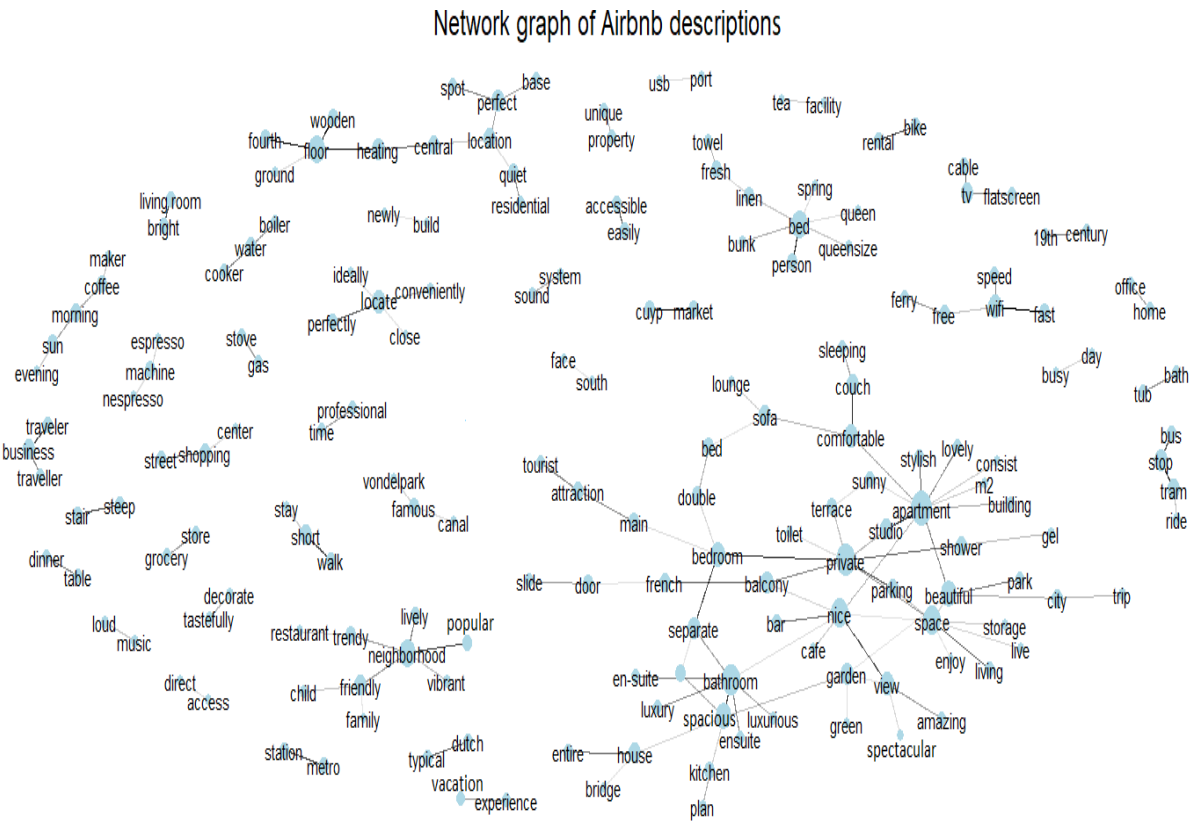

Figure 6- Network graph of popular words in the descriptions

Moreover, an analysis of the top 100 words from the description feature and how the frequencies of these words varied across different price ranges was done.To do this, I first computed the proportion of each word in each price range. To quantify the association of each word with the price range, I fitted a logistic regression model for each word, using the count of the word and the total count in each price range as a binomial response, and the price range as the predictor.

After fitting these models, I extracted the estimated coefficients for price range and their associated p-values. For the lower price ranges, I selected the words that not only had a statistically significant association with the price range (p-value <0.05), but also had the largest negative coefficient estimates. These words are more commonly used in descriptions of listings with lower price ranges. For higher price ranges, I followed a similar approach, but instead selected the words that had the largest positive coefficient estimates. These words are more commonly used in descriptions of listings with higher price ranges. The “low price words” and “high price words” are depicted in figure 7 and 8, respectively.

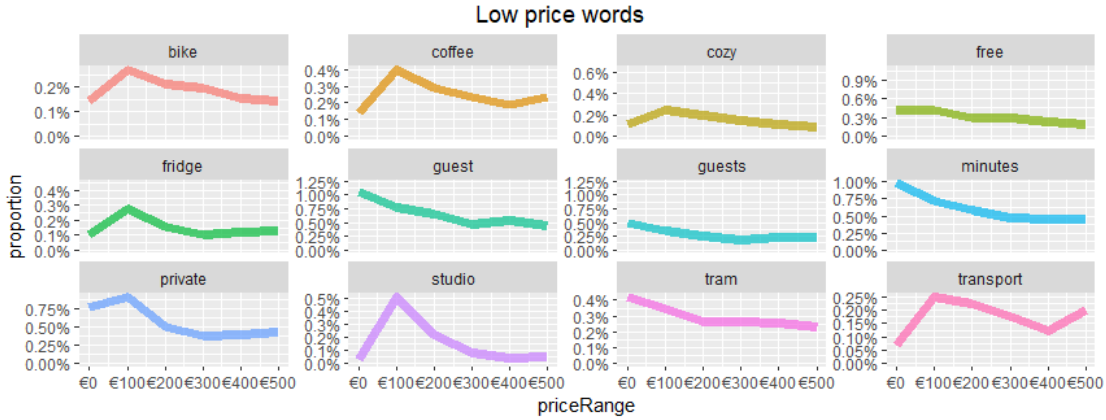

Figure 7- Most common words for lower priced listings.

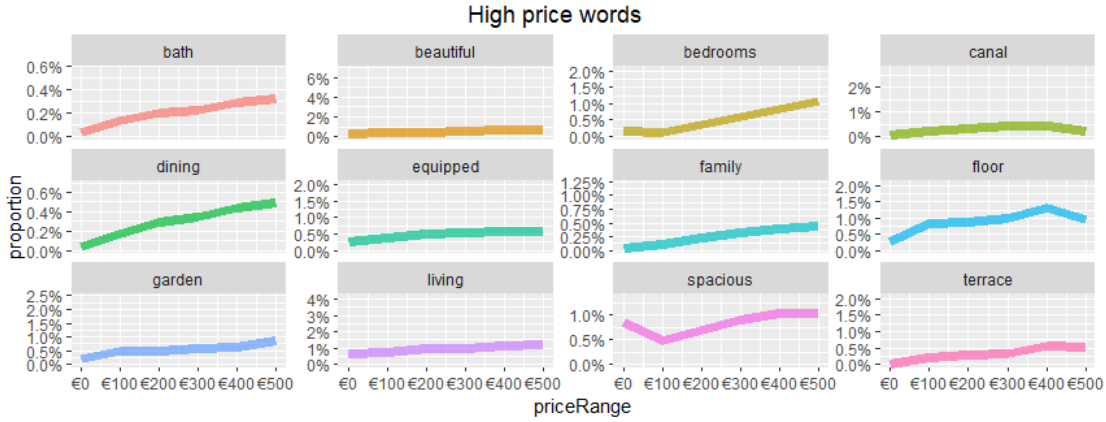

Figure 8- Most common words for higher priced listings.

Sentiment scores for each listing have been included based on the reviews and were made using a lexical approach. This process started with removing all html tags (like '
', '', and '') and new line characters ('\n'), which might interfere with the text analysis. Then, a language detector called 'textcat' was used to only keep English reviews. Furthermore, each comment was broken down into sentences. The final step was to compute sentiment scores on each sentence of the review comments. Sentiment scores were calculated using the sentiment function in 'sentimentr' package, which estimates the sentiment of a sentence by considering the impact of each word on the overall sentiment, as well as considering amplifiers, negators, and adversative conjunctions. The mean sentiment score for each listing was then calculated and used as a predictor in the prediction models.

Moreover, location-based features were also included. The geographical distribution of Airbnb listing prices in Amsterdam can be visualized in figure 9. The map was created using a

shapefile of Amsterdam neighbourhoods, which was merged with the Airbnb dataset to compute the median price per neighbourhood. This shows us that location matters for listing price and that the neighbourhood with highest median price is De Pijp-Rivierenbuurt. Longtitude and latitude will therefore be included as predictors. It would also be possible to one-hot encode the neighbourhoods instead of using latitude and longtitude, but I decided not to do this to avoid increasing the dimensionality of the dataset.

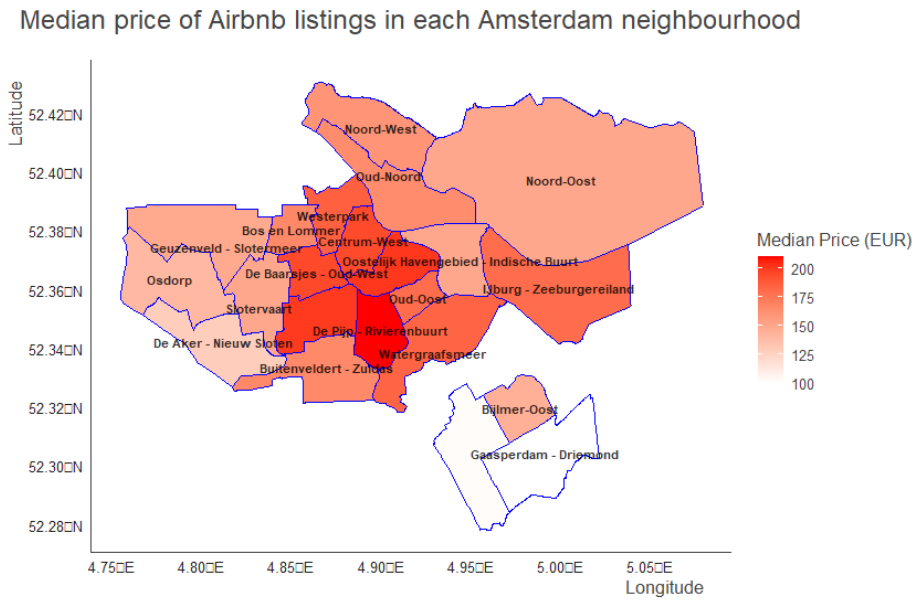

Figure 9- Map over Amsterdam displaying median price based on neighbourhood.

Median imputations of missing values were done for the “bathrooms”, “bedrooms”, and “beds” variables. The median was computed based on the training data and then the imputation was applied to the whole dataset. Some of the variables in the raw dataset are likely to only contribute noise to the models and have been removed. Examples of these are “id”, “host_id”, “host_url” and “listing_url”. For variables “host_response_time” and “host_response_rate”, their conversion into categorized formats was deemed necessary due to their limited variability in a continuous numeric format. In this process, one category labelled as “unknown” was introduced. It was considered more beneficial to retain these observations under this category, rather than completely excluding them from the analysis.

Deciding what to do with outliers can be challenging. Retaining them could potentially hamper the predictive performance of the models, while eliminating or transforming them might result

in information loss. In this thesis, I will primarily interpret models that only transform outliers likely to be erroneous, while natural outliers will be preserved. After completing the cleaning steps, the dataset is reduced to 6053 observations. This is mostly because some of the URL's to the thumbnail images did not lead to the images. Descriptive statistics for all the input variables can be seen in figure A1 in the appendix.

4.2.6 Near-zero variance filter

In the analysis, the variables that had little to no variance were identified and removed. Variables that showed minimal variation are unlikely to contribute much useful information to the predictive model, since they are almost constant for all observations. The variables that were removed from this filter were the variables "Wi-Fi", "pool", "smoking allowed", and "breakfast". Wi-Fi is something that most hosts offer. Too few listings include pool and breakfast, and it is also very rare that smoking is indeed allowed.

4.2.7 Normalization

Next, the variables were normalized in the dataset. Normalization is the process that standardizes the range of the independent variables so that they have a range from 0 to 1. The normalization parameters are based on the training set and applied to avoid data leakage. Normalization can be useful when the variables have different scales. By ensuring that all independent variables are on a similar scale, we can help to ensure that the models treat all predictors equally and is not unduly influenced by variables simply because they have larger scales or wider ranges.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{8}$$

Here, x_i represents the original value of a variable, *min(x)* and *max(x)* are the minimum and maximum values of that variable in the training set, and z_i is the normalized value.

5. Models

The models have been trained using the Tidymodels package in R (Kuhn & Wickham, 2020). To ensure unbiased evaluation and model performance, the dataset was divided into training and testing sets. The dataset was randomly sampled into a training set consisting of 75 % of the data and 25 % as test set. To further enhance the reliability of the model, K-fold cross-validation was applied with 10 folds. The training data was used to create stratified folds based on the target variable, price. When performing stratified sampling or cross-validation, the goal is to ensure that each subgroup or stratum is represented proportionally in the training and test sets. In other words, the distribution of the target variable within each subset closely reflects the distribution of the entire dataset. By stratifying the data based on the price variable, we ensure that each subset of the evaluation contains a representative mix of different price ranges. This enhances robustness and reliability of the models. Furthermore, the models will be optimized based on all the performance metrics mentioned in the theory section.

5.1 Tuning Hyperparameters

In the process of building the machine learning models, hyperparameters were tuned for each model to optimize their performance. This involved employing a grid search which is a method for hyperparameter tuning where a predefined range of hyperparameter values is methodically searched. This method can be computationally expensive but is often used due to its simplicity and because it can be highly effective (Ismiguzel, 2023).

There are different strategies for defining hyperparameter space in grid search. Two common approaches were used in this project, namely Regular Grid Search and Latin Hypercube Sampling. In a Regular Grid Search, the grid is constructed in a regular pattern, meaning it covers the hyperparameter space evenly. It generates all combinations of the specified hyperparameter values. This is a brute-force exhaustive searching paradigm where each combination of hyperparameters is evaluated. Although it is it is guaranteed to find the optimal hyperparameters within the specified range, it can be computationally intensive, especially when dealing with many hyperparameters or when hyperparameters can take many values. Unlike Regular Grid Search, Latin Hypercube Sampling does not test all combinations of

values. Instead, it samples the hyperparameter space in a way that ensures a balanced and representative selection of hyperparameter values. The space is divided into a grid, and exactly one value is chosen from each row and column. This strategy reduces the computational cost compared to the Regular Grid Search and can provide a more efficient exploration of the hyperparameter space, especially in cases where it is high-dimensional (Urban & Fricker, 2010). A Regular Grid Search was applied to all the tuned models expect for the XGBoost model and the neural network model. Instead, a Latin Hypercube Sampling approach was used because they consisted of many hyperparameters.

For linear regression, a vanilla model was used, without any hyperparameter tuning. In contrast, for Lasso and Ridge regression, a hyperparameter tuning process was applied. The key hyperparameter in these cases is the penalty term. In both Lasso (L1 regularization) and Ridge (L2 regularization) regression, this penalty term controls the degree of regularization applied to the model (James et al., 2021). Regularization is a technique used to prevent overfitting by discouraging overly complex models, thereby promoting generalizability to unseen data. For Lasso regression, the penalty term is applied to the absolute values of the model coefficients. This can lead to some coefficients being set to zero, effectively eliminating those features from the model. Ridge regression, on the other hand, applies the penalty to the squared values of the coefficients. Unlike Lasso, Ridge cannot set the coefficients to zero but can shrink them close to zero. The Elastic Net model had its mixture parameter tuned to find the optimal balance between Lasso and Ridge regression. The tuning results for the Lasso, Ridge, and Elastic Net Regression can be seen in figure B1, B2, and B3 the appendix.

The XGBoost model had several hyperparameters tuned, including depth, minimum number of observations in a node, reduction on loss required to make further partition, fraction of the samples used to fit the individual base learners, number of variables randomly sampled as candidates at each split, and learning rate. The decision tree model had its complexity cost and tree depth parameters tuned, while the random forest model had the number of variables randomly sampled at each split, and the number of observations in a node parameters tuned. The tuning results for the XGBoost, Decision Trees, and Random Forest model can be seen in figure B4, B5, and B6 in the appendix.

The KNN model had the number of neighbours and the weight function parameters tuned. The SVM model had a radial basis function (RBF) kernel, where the cost and RBF sigma parameters were tuned. Lastly, for the multi-layer perceptron neural network model, several

parameters were tuned, including the number of hidden units, penalty parameter, number of epochs, and the activation function. The tuning results for the KNN and SVM models can be seen in figure B7, B8, and B9.

When certain selections of hyperparameters yielded similar or identical results on the training set, I chose the hyperparameters that yielded the most regularization as this increases the simplicity of the model.

5.2 Model Results

When evaluating the results of the price prediction models with the image features, the test set and validation set results exhibited remarkable consistency, with most models demonstrating similar performance on the validation set and the test set. This observation attests to the robustness of the developed models and their ability to generalize well to unseen data. The results for the models with the image features are in Table 3 and the ones without the image features are in Table 4. Interestingly, in several cases the test set results marginally surpass the validation set results, this somewhat counterintuitive outcome may be attributed to the intrinsic variance in the data sets. It is also plausible that the randomly sampled test set was less complex or had patterns more closely aligned with the models' learned parameters, leading to marginally better performance. The best performing model was XGBoost with an RMSE of 91.81, R^2 of 55.91% and an MAE of 58.56. This RMSE suggests that the predictions do not predict the actual price with a high degree of accuracy. Nevertheless, the main emphasis of this thesis is to see how the image features affect the hedonic price models which requires a comparison with the models that do not use these features. Removing the image features from the models resulted in a slightly worse performance on the XGBoost model in terms of RMSE and MAE. On the other hand, there are no signs of big impacts on the predictive performance for any of the models when excluding the image features. This suggests that while the image features can have some value, they do not fundamentally alter the performance of the hedonic price models. The primary drivers of the predictive performance are likely to be other, more traditional features in the models.

Model Performance Metrics						
	Test Set			Validation Set		
Model	RMSE	R^2(%)	MAE	RMSE	R^2(%)	MAE
Linear Regression	101.35	46.91	67.35	99.41	46.00	67.87
Lasso	101.47	47.02	66.95	98.83	45.37	65.95
Ridge	101.46	46.96	66.92	98.86	45.32	66.06
Elastic Net	101.33	47.07	67.12	98.82	45.38	65.96
XGBoost	91.81	55.91	58.56	91.62	52.81	59.17
Decision Tree	111.78	34.32	74.83	108.23	36.04	70.48
Random Forest	95.79	53.14	62.05	93.63	51.21	60.45
KNN	109.96	42.03	69.71	107.76	37.07	69.34
SVM	94.86	53.22	59.87	95.66	49.99	59.78
Neural Network	99.43	48.90	65.98	100.43	48.93	66.43

Table 3- Model performance with images

Model Performance Metrics						
	Test Set			Validation Set		
Model	RMSE	R^2(%)	MAE	RMSE	R^2(%)	MAE
Linear Regression	101.35	46.91	67.35	99.87	45.50	68.05
Lasso	101.47	47.02	66.94	98.60	45.65	65.73
Ridge	101.53	46.87	66.99	98.74	45.47	65.89
Elastic Net	101.40	47.04	66.87	98.59	45.65	65.74
XGBoost	93.10	55.98	59.09	92.80	51.70	59.64
Decision Tree	111.32	37.08	72.43	108.61	35.75	70.58
Random Forest	96.60	52.54	62.89	94.50	50.37	61.07
KNN	113.20	38.93	71.74	110.23	34.32	70.40
SVM	97.83	52.51	61.57	96.44	48.98	60.68
Neural Network	98.44	49.90	64.88	100.98	48.93	67.43

Table 4- Model performance without images

Having assessed the models' performance based on various metrics, it is beneficial to also examine the nature of the prediction errors using density plots and scatter plots. The mean residual of both density plots is slightly above 0, indicating that on average, the models overpredict. This is because the frequency of overpredictions is higher. Despite this, the presence of a few instances with significantly large underpredictions counteracts this trend, thereby nudging the average residual closer to zero. Moreover, the model that excludes the image features tends to overprice slightly more than its counterpart. Nevertheless, the substantial overlap between the two density plots suggest that the inclusion of image features does not radically alter the predictive accuracy of the pricing models.

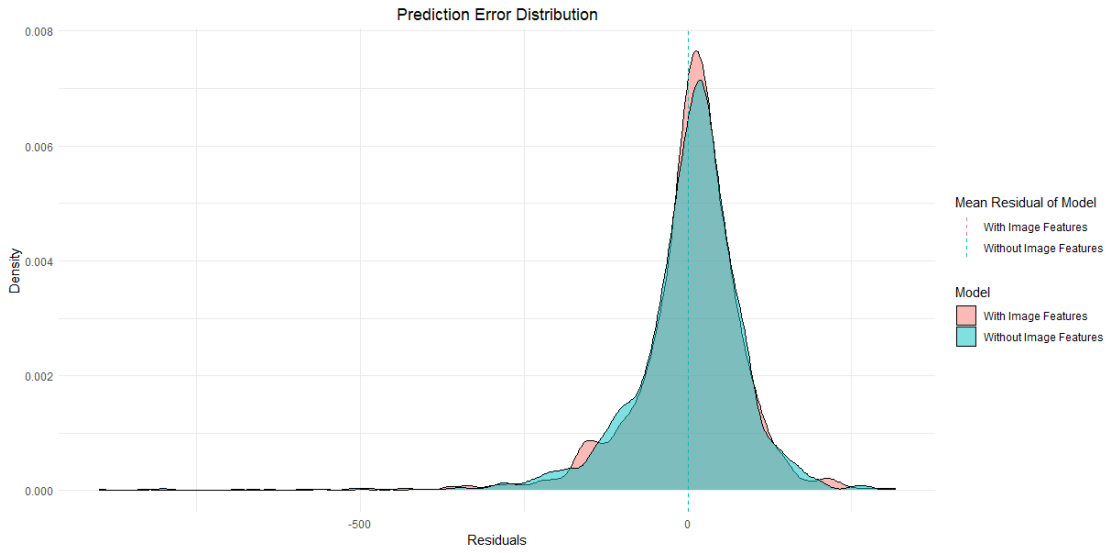

Figure 10- Density plot of residuals.

Figure 10 provides additional insight into the instances where the models falter in delivering accurate predictions. It is noteworthy, that the XGBoost models rarely predict a price exceeding 800 euros. Discounting the outliers, the errors of the models appear to be randomly distributed, as evidenced by their scattering around the diagonal line. Pertinently, no significant discrepancy is discernible between the model incorporating image features and the one excluding them. This observation aligns with the prior analysis, further reinforcing the conclusion that image features do not markedly influence the performance of the pricing models.

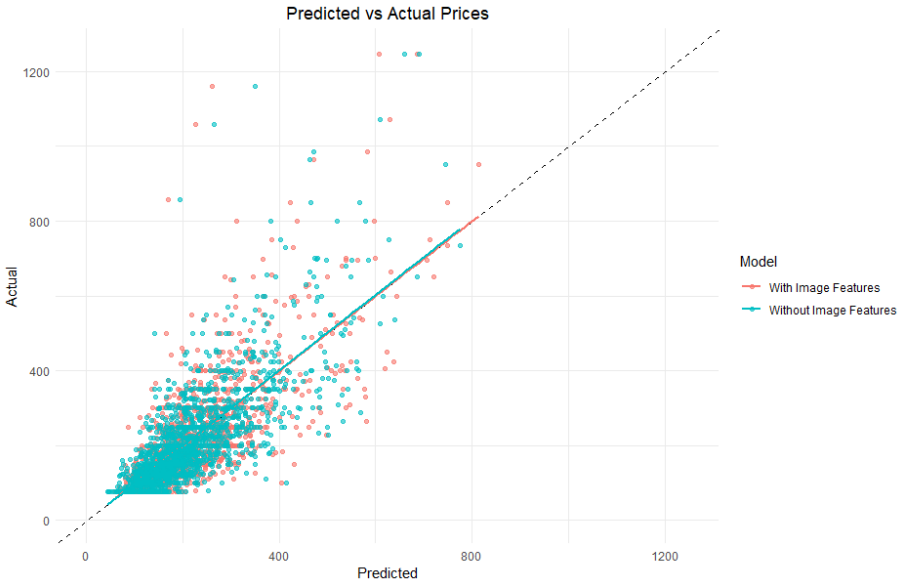

Figure 11- Scatter plot of residuals.

Though there is a minor improvement observed in the optimal model when image features are incorporated, it is useful to evaluate the statistical significance of this improvement. A paired t-test, a commonly applied method to compare means of paired samples, was initially considered. However, it demands an assumption of normal distribution for the differences between paired samples. To evaluate this, a histogram, a qq-plot, and a Cullen and Frey graph were employed to inspect the residuals of the predictions. These plots are fond in the appendix in figure C1, C2, and C3. Furthermore, observations from these analyses suggested deviations from normality. Therefore, a Wilcoxon signed-rank test, a non-parametric alternative to the paired t-test was deemed more appropriate as it requires fewer assumptions regarding the residuals' distribution.

The Wilcoxon test examines the null hypothesis that the median difference between pairs of observations is zero, implying no significant difference in performance between the two

models. A significant result would suggest one model consistently outperforms the other. However, the Wilcoxon test yielded a p-value of 0.445, not significant at conventional alpha level of 0.05. This indicates no evidence of one model consistently outperforming the other.

While the Wilcoxon test is robust against deviations from normality, it requires symmetry in the distribution of differences and independence between pairs. The cross-validation may introduce some dependency between predictions, but this mild violation is generally acceptable and does not significantly impact the test's validity when comparing models.

5.3 Interpreting the Effect of Image Features

What follows is an interpretation of the relevance of the image features in the XGBoost model. Most of the interpretation methods, except ALE, can be affected by correlated features. Figure D1 in the appendix shows that there is mostly no multicollinearity in the model.

5.3.1 Permutation Feature Importance

In 2, the features of the model are presented, ordered by their importance. The plot was made using the "iml" package (Molnar, 2018). All the image features, except those associated with image labels, are present in the plot, suggesting they affect the predictive performance of the models positively. In addition to using the model-agnostic approach to determine feature importance, other methods that are specific to tree-based models were also employed by using the "vip" package in R (Greenwell & Boehmke, 2020). Three distinct approaches were adopted. Firstly, "gain" the default method, determines the fractional contribution of each feature by quantifying the total gain that arises from the splits of the corresponding feature. This essentially quantifies the improvement in accuracy attributed to a feature's splits. Secondly, the "cover" method gauges the number of observations that are associated with each feature. This allows us to grasp the feature's representation within the dataset. Lastly, the "frequency" method computes the relative number of times each feature is used across all trees in the ensemble. This metric essentially highlights the frequency of a feature's utility in shaping the decision trees (Greenwell & Boehmke, 2020). The noteworthy point is that, irrespective of the approach used, the image features consistently surfaced as important. This consistency reinforces the claim that the image features can improve the performance of the pricing models.

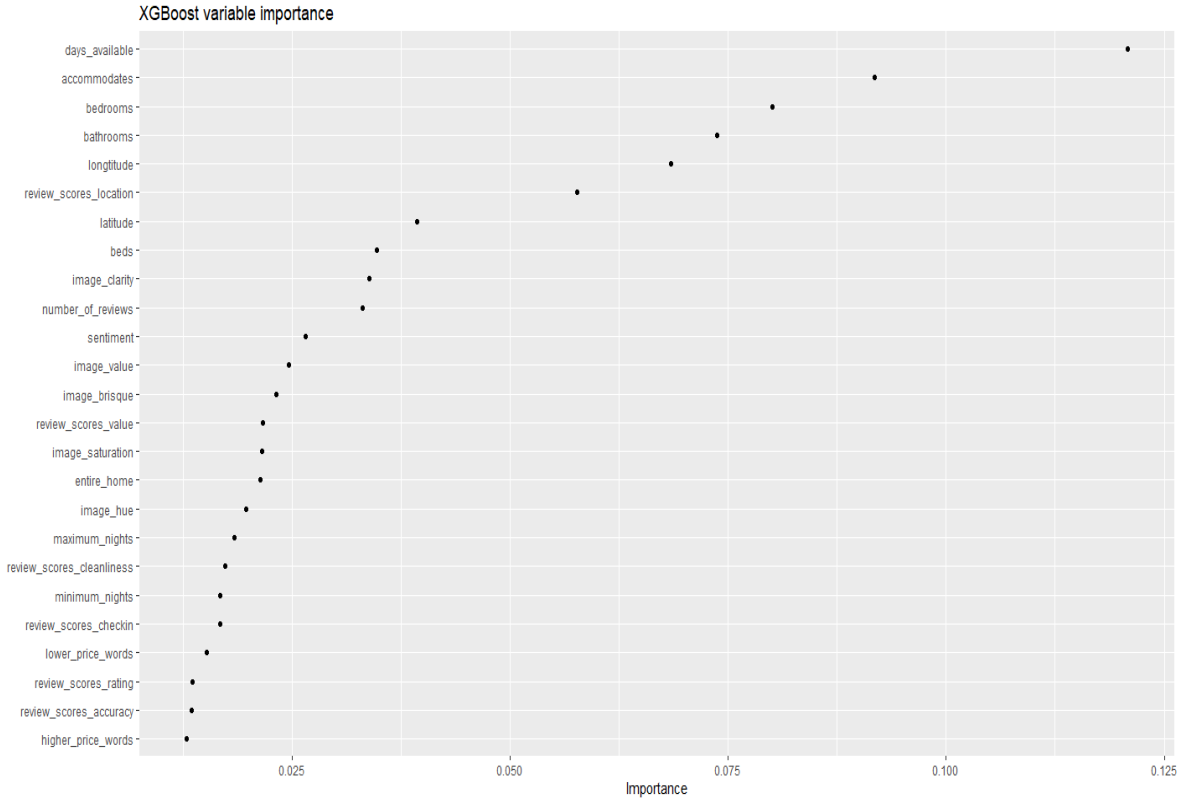

Figure 12- Variable importance plot.

5.3.2 ALE

This section foucses on the ALE plot analysis. ALE plots offer a straightforward interpretation. The relative effect of a change in the feature value, given the value of that feature, can be discerned from the ALE plot. The plots are centered around zero, which facilitates the interpreatiotn since the value at each point in the ALE curve signifies the deviation from the average prediction. The rug mapping on the x axes in figure 13 indicates the observations. For instance, most of the observations of image clarity are from 0 to 3000, and the number of observations with values exceeding that are less frequently occurring in the sample.

There is a rising effect on the price as image clarity increases, especially beyond a clarity value of 2400. This suggests that clearer images tend to yield higher listing prices. From the brisque plot, there is an indication that the highest quality images are associated with an average increase in price of €20 compared to baseline. The brighter images on average increase the price by €5. Conversely, darker images can lower the price on average by the same amount. Regarding the image hue, there is an average price increase at low and high hue levels,

indicating that the average colour for the image is red. As for the saturation level and the image level, the effects are either nonexistent or very small.

As mentioned in the theory section, quantiles of the feature's distribution are used to define the intervals. This approach ensures equal data instances in each interval, contributing to the fair representation of the feature's effect on the model predictions across its distributions. However, this method does come with a downside. Due to the nature of the quantiles, the interval lengths can vary significantly, particularly when the feature's distribution is heavily skewed with a majority of either low or high values. This can result in unusual-looking ALE plots with potentially misleading interpretations. In this context, the distributions are slightly skewed for some of the features even after winsorizing the data.

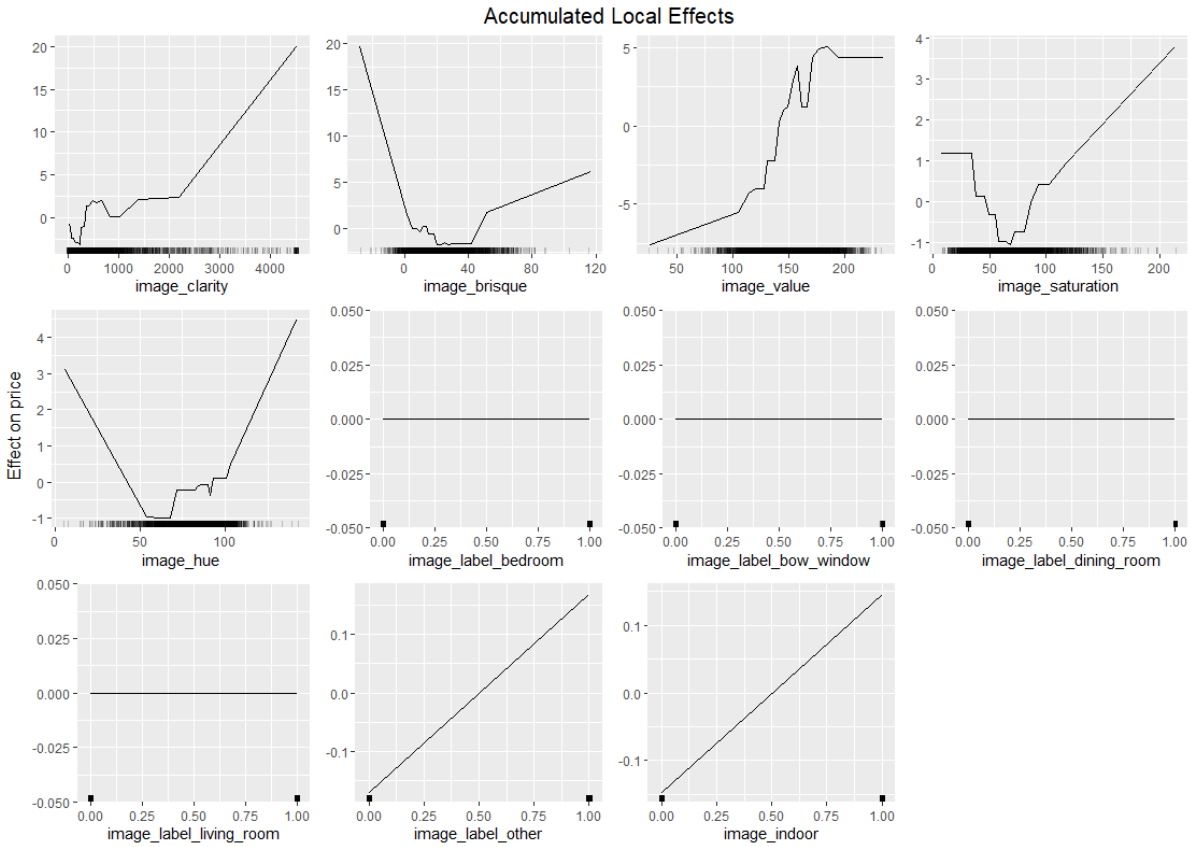

Figure 13- ALE plots.

5.3.3 H-Statistic

The H-statistic ranks features based on their interaction with other features in the model, with the feature having the highest H-statistic placed at the top. This feature is the one that interacts the most with all the other features in the model. The length of the bar represents the H-statistic for a particular feature. A longer bar implies a higher degree of interaction. If a feature scores high in this plot, it means that the effect of this feature on the price variable depends heavily on the values of the other features. If a feature scores low on the plot, its effect on price is largely independent of the other features. Furthermore, the features with high H-statistic can be considered important because of their combined effect with other features. The H-statistics can be seen from figure 14.

Some of the image features show up in the H statistic, namely, “image_brisque”, “image_clarity”, “image_saturation”, and “image_value”. However, the highest interaction effect among the image features is the brisque score with an H statistic of 0.06. This implies that the image features have a small degree of interaction with the other predictors. Remember, the H-statistic measures the interaction effects, not the main effect of the features on the target variable. Therefore, a feature can have a high importance due to its direct effect on the target variable (high feature importance) but have a low H-statistic if it does not interact with other features. Conversely, a feature can have a low feature importance but a high H-statistic if it mainly affects the target variable through its interactions with other features. To conclude,

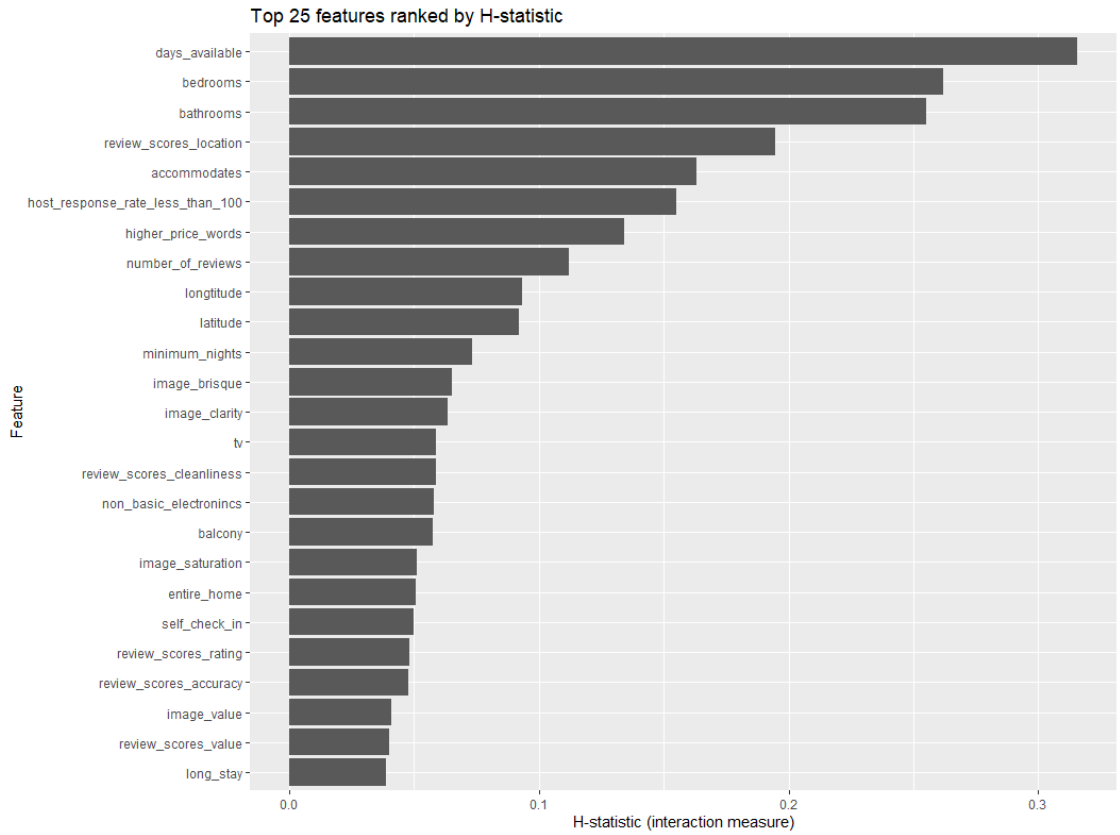

Figure 14- H-statistics (Interaction measure)

5.3.4 Shapley Values

Figure D2 in the appendix displays a SHAP (Shapley Additive explanation) visualization plot specifically made for XGBoost and is made with the "SHAPforxgboost" package. Features are ordered from top to bottom by the sum of SHAP value magnitude. The x-axis location of each dot shows the impact of that feature on the model's prediction. The SHAP values on the left side are linked with lower prices and the SHAP values on the right side are linked with higher prices. When the dots are yellow it signifies that the feature value is low, whereas the purples values indicate a high feature value. Blue lines are inserted into the plot to emphasize image features.

We can see that when the image clarity is lower (blurrier) the price tends to be lower. When the images are clearer, however, the price tends to be higher.

The brightness of the image color, represented by the feature "image_value", showed a complex relationship with the predicted price. While brighter images were generally slightly more likely to contribute to higher prices, there were notable exceptions where brighter colors were associated with lower prices. This finding suggests that the effect of image brightness on

price might depend on other features, as seen from the H-statistic in figure x. Another possible explanation is that the relationship is simply not linear, or that the feature causes noise.

A complex relationship is also the case for the brisque feature. We observe a mixture for both yellow (high image quality) and purple (low image quality) dots across both sides of the plot. This pattern suggests that the impact of image quality is not linear either.

As for image saturation, the SHAP values suggest that this image feature plays an interesting role in the prediction of price. As we see from the plot, images with higher saturation(vivid colors) are more frequently associated with higher predicted prices, as indicated by the prevalence of purple dots on the right side. Conversely, images with lower saturation (more muted colors or grayscale) are often associated with lower predicted prices, suggested by the presence of yellow dots on the left side. This could be interpreted as an indication that more vibrant, colorful images tend to be associated with higher-prices listings, while more muted images tend to be associated with lower-priced listings. However, as with any other feature, the impact of image saturation on price prediction does not exist in isolation and can be influenced by the values of other features in the model.

The SHAP values for image hue reveal that images with a higher value of hue- corresponding to specific colors or shades- are associated with both low and high predicted prices.

As for “image_hue” lower values for this feature are associated with higher prices and that slightly higher hues bring the price down. To further examine this we can examine a partial dependence plot of the SHAP values in figure x. It seems like the warm colours (red and yellow) are associated with higher prices whereas cooler colours (blue and green) are associated with lower prices.

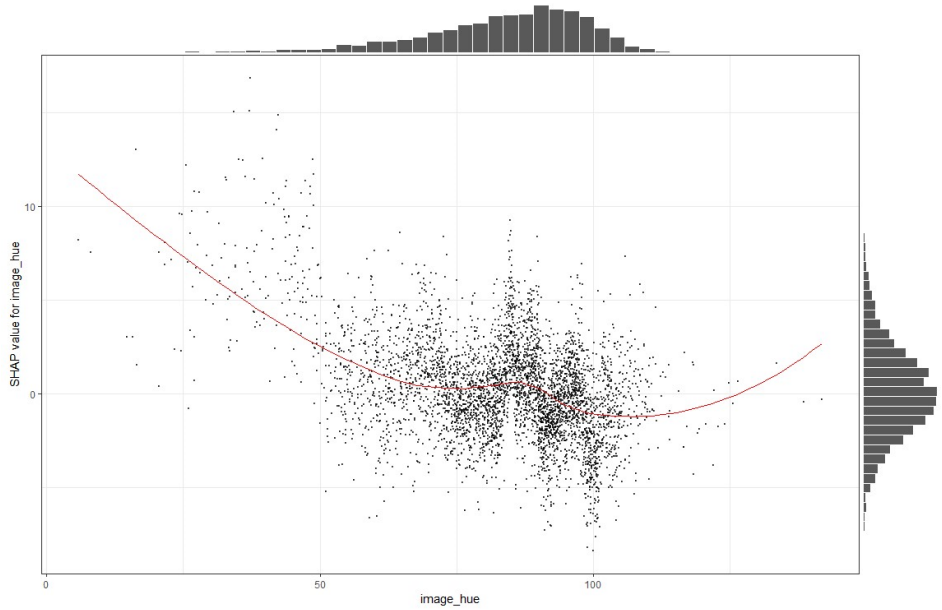

Figure 15- Partial dependence plot for image hue

5.3.5 LIME

When it comes to the features that affect the prediction of the most expensive listing in the test data, the image features seem to not play a big role. Figure 16 shows that a high quality image according to brisque score as a very small positive impact on the price of this listing. The same applied for the saturation of the image. Nonetheless, it is the fact that the image has air conditioning, has more than one bathroom, and is available for more than 98 days that seem to have the highest impact in increasing the price for this prediction. However, on a scale of 0 to 1, the explanation fit (R^2) has a value of 0.52 and the correct prediction would have been a price of 1246. We should therefore not put too much faith in these explanations.

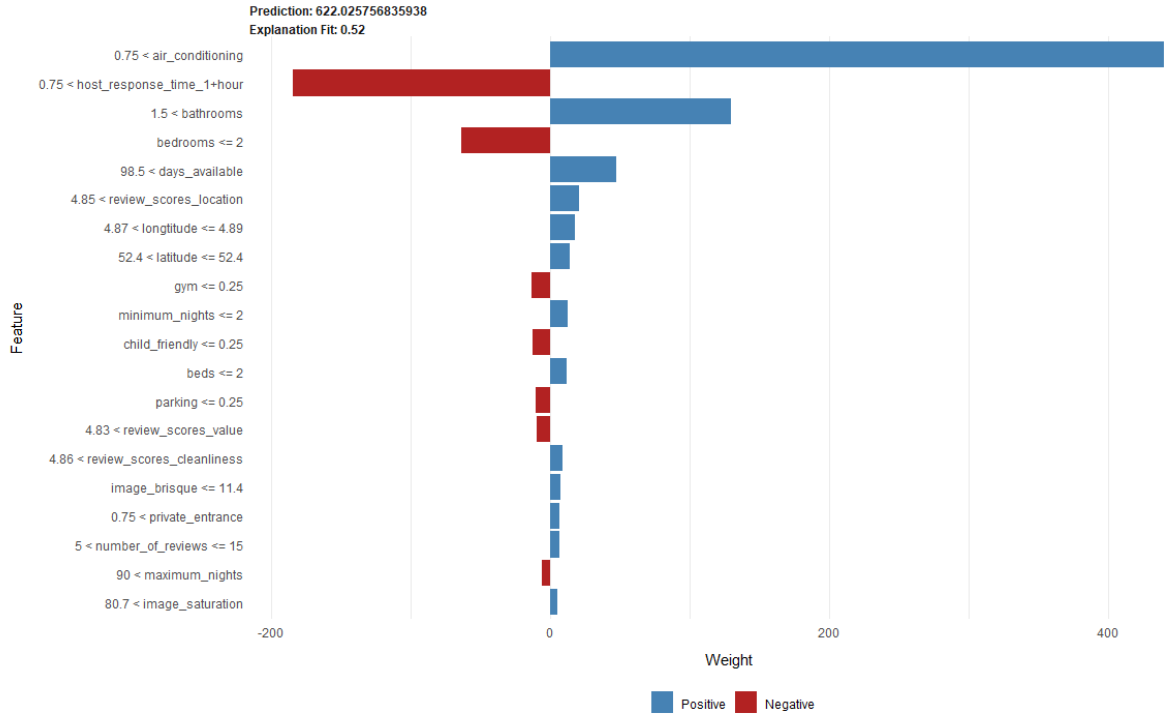

Figure 16- LIME plot for the most expensive listing.

The cheapest listing in the test set costs 77 euros but is predicted to cost 119 euros. In this case the image hue has a positive effect on the price whereas the image clarity has a negative one. It is rather the fact that the listing has a lack of many bedrooms, bed linen, bathrooms and higher priced words in the description that seem to drag the prediction of the price down, to name a few. In this case the explanation fit is even worse than for the highest prediction which means that we should not trust these local interpretations too much.

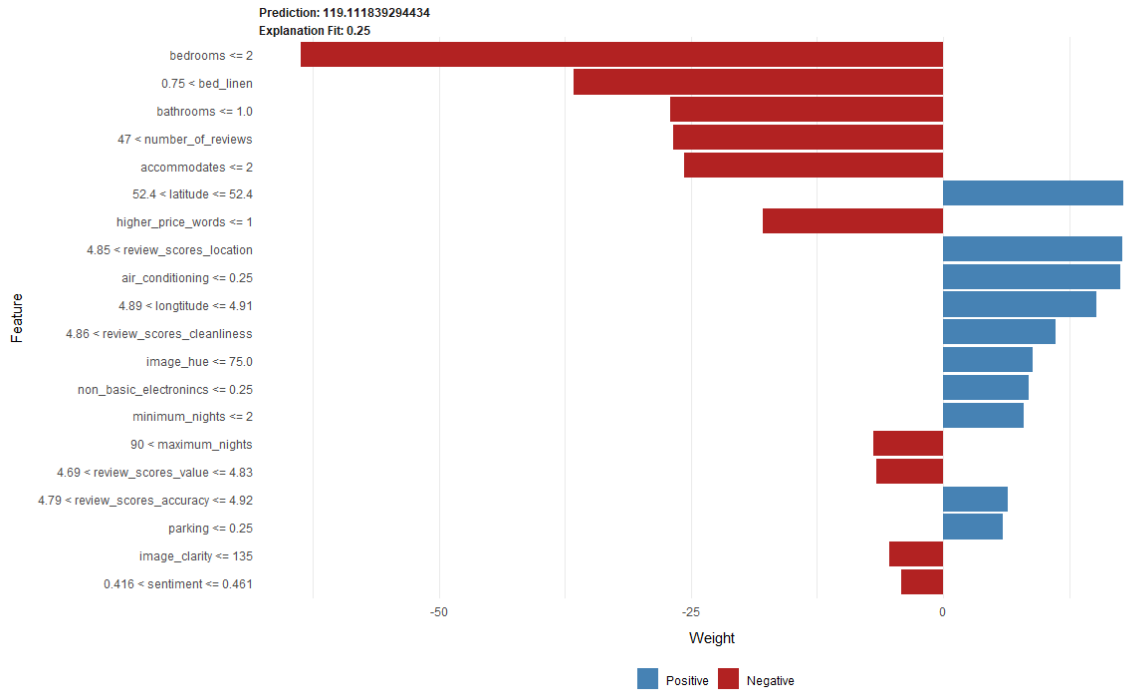

Figure 17- LIME plot for the cheapest listing.

6. Limitations and Further work

The data utilized in this study, while comprehensive, comes with inherent challenges and limitations. Primarily, the dataset only includes advertised prices, often referred to as "sticker" prices. The sticker price is the nightly price that hosts present to potential guests. However, it may not reflect the actual average amount paid per night by previous guests. The nature of sticker pricing is such that hosts, particularly those less familiar with Airbnb, can set these prices to any arbitrary amount, often leading to extremely low or exceedingly high values (Lewis, 2021). It is important to consider the role of hosts' individual preferences and capabilities. Some hosts may have a more precise understanding of the market and thus might be more adept at setting optimal prices for their listings. On the other hand, others might have less experience or information, leading to potential mispricing. Additionally, budgetary constraints or the personal objectives of the host could influence the listed price, creating further source of unexplained variation in prices. Nonetheless, it should be noted that this limitation also suggests an area for future research: understanding the decision-making processes of Airbnb hosts when it comes to setting prices. By doing so, we could better model and predict Airbnb prices and improve the accuracy of models like the ones used in this thesis.

Moreover, some additional costs that the data does not capture are the additional fees like guest fee, cleaning fee, pet fee and local taxes depending on location. These constraints in the dataset imply that while it serves as a suitable proof of concept, the study could potentially be enhanced with more enhanced data. Access to data detailing actual average nightly rates paid by guests, available through platforms such as AirDNA, would offer a more accurate representation of Airbnb pricing trends and their prediction.

While using a pretrained model like ResNet18 speeds up the process and simplifies the workload, it may not be perfectly suited to my specific dataset. ResNet18 was trained on a general dataset and was not fine-tuned to the specific images used in this thesis. This might limit its capacity to accurately label the scenes within this specific context. Future work could involve manually labeling some of the images, for example with the use of crowdsourcing to improve the label classifications.

Despite the inclusion of image features such as image labels, hue, saturation, value/brightness, and clarity, this thesis did not consider other potential influential image features that could provide additional predictive power or insights. For instance, it could be interesting to include features related to the spaciousness or the figure-ground relationship of the image. Clear figure-ground separations can possibly draw more attention and increase property demand (Zhang et al., 2016). Moreover, detecting amenities though images could also potentially affect the predictive performance for the listings.

Another consideration relates to the role of the thumbnail images. It is plausible that a high-quality or visually appealing thumbnail image could attract more views or clicks on a listing, but this might not directly translate is also possible that a good thumbnail image causes more traction and clicks on the listing but that it does not result in higher prices. Instead, the impact of image features might be more evident in metrics related to user engagement, such as click-though rates, booking conversion rates, or even guest satisfaction ratings. Additionally, a potential limitation of this thesis lies in the use of only the thumbnail images provided for each Airbnb listing. As each property listing usually includes multiple images showcasing different parts of the property, the thumbnail image alone might not be fully representative of the listing's overall aesthetic and functional appeal. In essence, the use of a single thumbnail image alone might not capture the full variety of rooms, amenities, and spaces offered by the property. Analysing multiple images would allow for a more comprehensive understanding of the property's visual presentation, which could further enhance the model's predictive

capability. However, this approach would significantly increase the complexity and computational demands of the image analysis, which should be considered in the planning stages of future studies.

7. Conclusion

The goal of this study was to answer the following problem formulation:
How do image features affect the predictive performance of hedonic price models for Airbnb listings?

To answer the problem formulation, the inclusion of image features in the hedonic price models for Airbnb listings provided a slight enhancement to the predictive performance of the XGBoost model. Specifically, the RMSE of the XGBoost model improved from 93.10 to 91.81 when adding the image features. While the observed improvement was not statistically significant, it underscores the potential utility of image features in refining the predictive process. Notably. The impact of these features varied across different models. Several models exhibited no performance improvements, or even registered a decline, when the image features were incorporated.

Despite the marginal on overall predictive performance, it is important to highlight that image features demonstrated to be important in various model interpretation techniques. The image features showed consistent relevance in according to the Permutation Feature Importance plots, ALE plots, H-statistics, and Shapley values. This suggests that these features play a role in shaping the model's decision-making process. However, image features pertaining to image labels did not reveal notable importance across the interpretation methods used in this study.

These findings imply that while image features may not drastically boost overall model performance, they engage in significant interactions with other features, contributing to model complexity and decision-making nuances. This raises further questions about the intricate role of image features in hedonic pricing models, warranting continued exploration in this area.

References

Ahmed, E., & Moustafa, M. (2016). House price estimation from visual and textual features. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1609.08399

Boehmke, B. C., & Greenwell, B. (2019). Hands-On Machine Learning with R. In *Chapman and Hall/CRC eBooks*. https://doi.org/10.1201/9780367816377

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

Choosing the correct upper and lower HSV boundaries for color detection with`cv::inRange` (OpenCV). (n.d.). Stack Overflow. https://stackoverflow.com/questions/10948589/choosing-the-correct-upper-and-lower-hsv-boundaries-for-color-detection-withcv/48367205#48367205

Dogru, T., & Pekin, O. (2017). What do guests value most in Airbnb accommodations? An application of the hedonic pricing approach. *Boston Hospitality Review*, *5*(2). https://vtechworks.lib.vt.edu/handle/10919/79602

Ert, E., Fleischer, A., & Magen, N. (2016). Trust and reputation in the sharing economy: The role of personal photos in Airbnb. *Tourism Management*, *55*, 62–73. https://doi.org/10.1016/j.tourman.2016.01.013

Fagerstrøm, A., Pawar, S., Sigurdsson, V., Foxall, G. R., & Yani-De-Soriano, M. (2017). That personal profile image might jeopardize your rental opportunity! On the relative impact of the seller's facial expressions upon buying behavior on Airbnb™. *Computers in Human Behavior*, *72*, 123–131. https://doi.org/10.1016/j.chb.2017.02.029

Faisal. (2023). OpenCV HSV range. *EDUCBA*. https://www.educba.com/opencv-hsv-range/

File:HSV color solid cylinder.png - Wikimedia Commons. (n.d.). https://commons.wikimedia.org/wiki/File:HSV_color_solid_cylinder.png

Fisher, A., Rudin, C., & Dominici, F. (2019). All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, *20(177)*, 1–81. https://arxiv.org/abs/1801.01489

Friedman, J. H., & Popescu, B. E. (2008). Predictive Learning via Rule Ensembles. *The Annals of Applied Statistics*, *2*(3), 916–954. https://www.jstor.org/stable/30245114

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Greenwell, B., & Boehmke, B. C. (2020). Variable Importance Plots—An Introduction to the vip Package. *R Journal*, *12*(1), 343. https://doi.org/10.32614/rj-2020-013

Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development*, *15*(14), 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022

Ismiguzel, I. (2023, March 29). Hyperparameter Tuning with Grid Search and Random Search. *Medium*. https://towardsdatascience.com/hyperparameter-tuning-with-grid-search-and-random-search-6e1b5e175144

James, G. M., Witten, D., Hastie, T., & Tibshirani, R. (2021). Introduction. In *Springer eBooks* (pp. 1–14). https://doi.org/10.1007/978-1-0716-1418-1_1

Kalehbasti, P., Nikolenko, L., & Rezaei, H. (2019). *Airbnb Price Prediction Using Machine Learning and Sentiment Analysis* (1st ed., Vol. 12844). Springer Cham. https://doi.org/10.48550/arXiv.1907.12665

Kim, S. (2022, July 15). Explainable AI (XAI) Methods Part 3 — Accumulated Local Effects (ALE). *Medium*. https://towardsdatascience.com/explainable-ai-xai-methods-part-3-accumulated-local-effects-ale-cf6ba3387fde

Kuhn, M., & Wickham, H. (2020). *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles*. Tidymodels. Retrieved March 6, 2020, from https://www.tidymodels.org/

Lancaster, K. (1966). A New Approach to Consumer Theory. *Journal of Political Economy*, *74*(2), 132–157. https://doi.org/10.1086/259131

Lewis, L. (2021, December 10). Exploring Airbnb prices in London: which factors influence price? *Medium*. https://towardsdatascience.com/predicting-airbnb-prices-with-deep-learning-part-2-how-to-improve-your-nightly-price-50ea8bc2bd29

Luo, Y., Zhou, X., & Zhou, Y. (2019, December 14). *Predicting Airbnb Listing Price Across Different Cities*. https://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26647491.pdf

Mittal, A., Moorthy, A. K., & Bovik, A. C. (2012). No-Reference Image Quality Assessment in the Spatial Domain. *IEEE Transactions on Image Processing*, *21*(12), 4695–4708. https://doi.org/10.1109/tip.2012.2214050

Molnar, C. (2018). iml: An R package for Interpretable Machine Learning. *Journal of Open Source Software*, *3*(26), 786. https://doi.org/10.21105/joss.00786

Molnar, C. (2023). *Interpretable Machine Learning* (2nd ed.). https://christophm.github.io/interpretable-ml-book/

Moreno-Izquierdo, L., Egorova, G., Peretó-Rovira, A., & Más-Ferrando, A. (2018). Exploring the use of artificial intelligence in price maximisation in the tourism sector: its application in the case of Airbnb in the Valencian Community. *Journal of Regional Research*, *42*. http://hdl.handle.net/10045/86772

Muralidhar, K. (2021, December 30). Outlier detection methods in Machine Learning - Towards Data Science. *Medium*. https://towardsdatascience.com/outlier-detection-methods-in-machine-learning-1c8b7cca6cb8

Nguyen, L., Ruiz-Correa, S., Mast, M. S., & Gatica-Perez, D. (2018). Check Out This Place: Inferring Ambiance From Airbnb Photos. *IEEE Transactions on Multimedia*, *20*(6), 1499–1511. https://doi.org/10.1109/tmm.2017.2769444

O'Sullivan, C. (2021, January 16). Finding and Visualising Interactions - Towards Data Science. *Medium*. https://towardsdatascience.com/finding-and-visualising-interactions-14d54a69da7c

Pech-Pacheco, J. L., Cristóbal, G., Chamorro-Martínez, J., & Fernandez-Valdivia, J. (2000). Diatom autofocusing in brightfield microscopy: a comparative study. In *International Conference on Pattern Recognition*. https://doi.org/10.1109/icpr.2000.903548

Potrawa, T., & Tetereva, A. (2022). How much is the view from the window worth? Machine learning-driven hedonic pricing model of the real estate market. *Journal of Business Research*, *144*, 50–65. https://doi.org/10.1016/j.jbusres.2022.01.027

Poursaeed, O., Matera, T., & Belongie, S. (2017). Vision-based real estate price estimation. *Journal of Machine Vision and Applications*, *29*(4), 667–676. https://doi.org/10.1007/s00138-018-0922-2

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1602.04938

Sagar. (2021, December 14). Laplacian and its use in Blur Detection - Sagar - Medium. *Medium*. https://medium.com/@sagardhungel/laplacian-and-its-use-in-blur-detection-fbac689f0f88

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, *41*(3), 647–665. https://doi.org/10.1007/s10115-013-0679-x

The amenities guests want. (2020, November 19). Airbnb. https://www.airbnb.com/resources/hosting-homes/a/the-amenities-guests-want-25

Urban, N. M., & Fricker, T. E. (2010). A comparison of Latin hypercube and grid ensemble designs for the multivariate emulation of an Earth system model. *Computers & Geosciences*, *36*(6), 746–755. https://doi.org/10.1016/j.cageo.2009.11.004

Valdez, P., & Mehrabian, A. (1994). Effects of color on emotions. *Journal of Experimental Psychology*, *123*(4), 394–409. https://doi.org/10.1037/0096-3445.123.4.394

Wilimitis, D. (2021, December 7). The Kernel Trick in Support Vector Classification - Towards Data Science. *Medium*. https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f

Zhang, S., Lee, D., Singh, P. V., & Srinivasan, K. (2016). How Much Is An Image Worth? An Empirical Analysis of Property's Image Aesthetic Quality on Demand at AirBNB. In *International Conference on Information Systems*. https://core.ac.uk/display/301370269?utm_source=pdf&utm_medium=banner&utm_campaign=pdf-decoration-v1

Zhang, S., Lee, D., Singh, P. V., & Srinivasan, K. (2021). What Makes a Good Image? Airbnb Demand Analytics Leveraging Interpretable Image Features. *Management Science*, *68*(8), 5644–5666. https://doi.org/10.1287/mnsc.2021.4175

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2018a). *Places365* [Dataset]. Massachusetts Institute of Technology. http://places2.csail.mit.edu/

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2018). Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(6), 1452–1464. https://doi.org/10.1109/tpami.2017.2723009

Appendices

Variable	Mean	Std Dev	Min	Max
price	215.6	135.0	77.0	1246.2
accommodates	2.9	1.3	1.0	16.0
air_conditioning	0.1	0.3	0.0	1.0
balcony	0.6	0.5	0.0	1.0
bathrooms	1.3	0.4	0.0	5.5
bbq	0.2	0.4	0.0	1.0
bed_linen	0.7	0.5	0.0	1.0
bedrooms	1.5	0.9	1.0	10.0
beds	1.8	1.5	1.0	33.0
child_friendly	0.2	0.4	0.0	1.0
coffee	0.8	0.4	0.0	1.0
cooking_basics	0.6	0.5	0.0	1.0
days_available	67.2	97.9	0.0	365.0
elevator	0.1	0.3	0.0	1.0
entire_home	0.7	0.4	0.0	1.0
gym	0.1	0.2	0.0	1.0
higher_price_words	2.7	2.4	0.0	18.0
host_greeting	0.3	0.5	0.0	1.0
host_identity_verified	0.9	0.3	0.0	1.0
host_listings_count	2.0	2.9	1.0	20.0
host_response_rate_100%	0.6	0.5	0.0	1.0
host_response_rate_<100%	0.1	0.4	0.0	1.0
host_response_rate_unknown	0.3	0.5	0.0	1.0
host_response_time_1+hour	0.3	0.5	0.0	1.0
host_response_time_<1hour	0.4	0.5	0.0	1.0
host_response_time_unknown	0.3	0.5	0.0	1.0
image_brisque	21.7	15.1	-28.4	117.1
image_clarity	573.0	772.0	5.1	4508.9
image_hue	83.6	15.7	5.8	141.8
image_indoor	0.8	0.4	0.0	1.0
image_label_bedroom	0.1	0.3	0.0	1.0
image_label_bow_window	0.1	0.3	0.0	1.0
image_label_dining_room	0.1	0.3	0.0	1.0
image_label_living_room	0.1	0.4	0.0	1.0
image_label_other	0.6	0.5	0.0	1.0
image_saturation	65.1	27.2	7.1	214.0
image_value	146.9	27.5	26.0	234.8
instant_bookable	0.2	0.4	0.0	1.0
latitude	52.4	0.0	52.3	52.4
long_stay	0.2	0.4	0.0	1.0
longtitude	4.9	0.0	4.8	5.0
lower_price_words	2.4	2.3	0.0	17.0
maximum_nights	446.1	501.4	1.0	1125.0
minimum_nights	2.7	1.1	1.0	4.5
non_basic_electroninc s	0.2	0.4	0.0	1.0
number_of_reviews	52.8	107.3	1.0	2310.0
parking	0.6	0.5	0.0	1.0
pets	0.1	0.3	0.0	1.0
private_entrance	0.5	0.5	0.0	1.0
review_scores_accuracy	4.8	0.2	1.0	5.0
review_scores_checkin	4.9	0.2	1.0	5.0
review_scores_cleanliness	4.8	0.3	1.0	5.0
review_scores_communication	4.9	0.2	1.0	5.0
review_scores_location	4.8	0.3	1.0	5.0
review_scores_rating	4.8	0.3	1.0	5.0
review_scores_value	4.6	0.3	1.0	5.0
safety	0.1	0.2	0.0	1.0
self_check_in	0.2	0.4	0.0	1.0
sentiment	0.4	0.1	-0.3	1.7
smoke_alarm	0.9	0.3	0.0	1.0
superhost	0.2	0.4	0.0	1.0
tv	0.8	0.4	0.0	1.0
view	0.1	0.2	0.0	1.0
white_goods	1.0	0.2	0.0	1.0

Figure A1- Descriptive statistics.

AppendixB-Sensitivity analysis

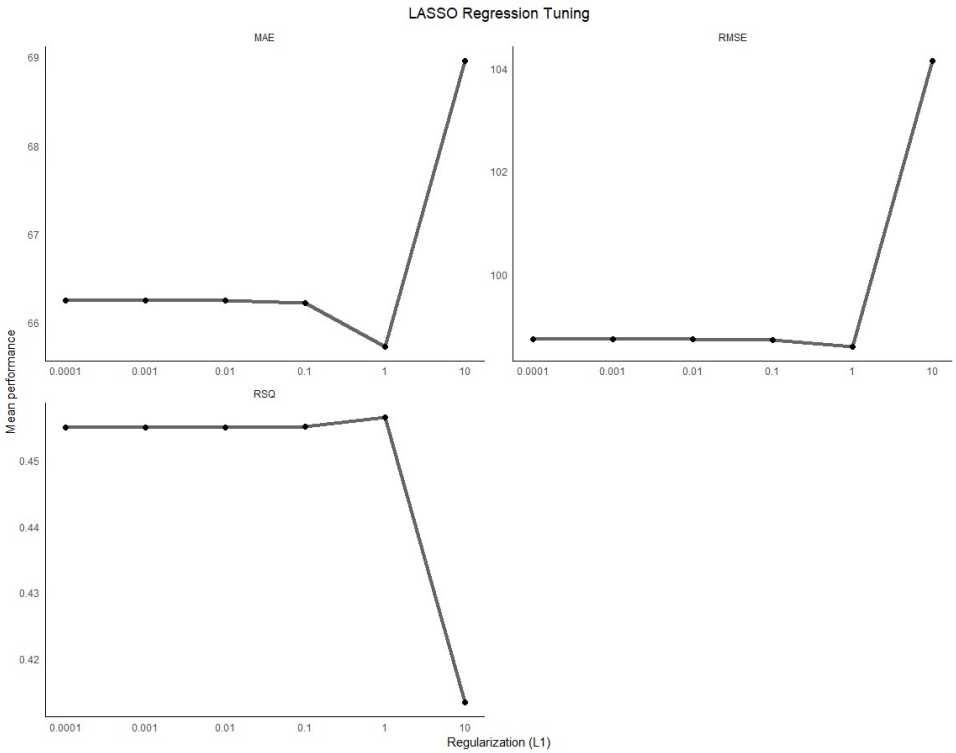

Figure B1- Lasso Tuning results.

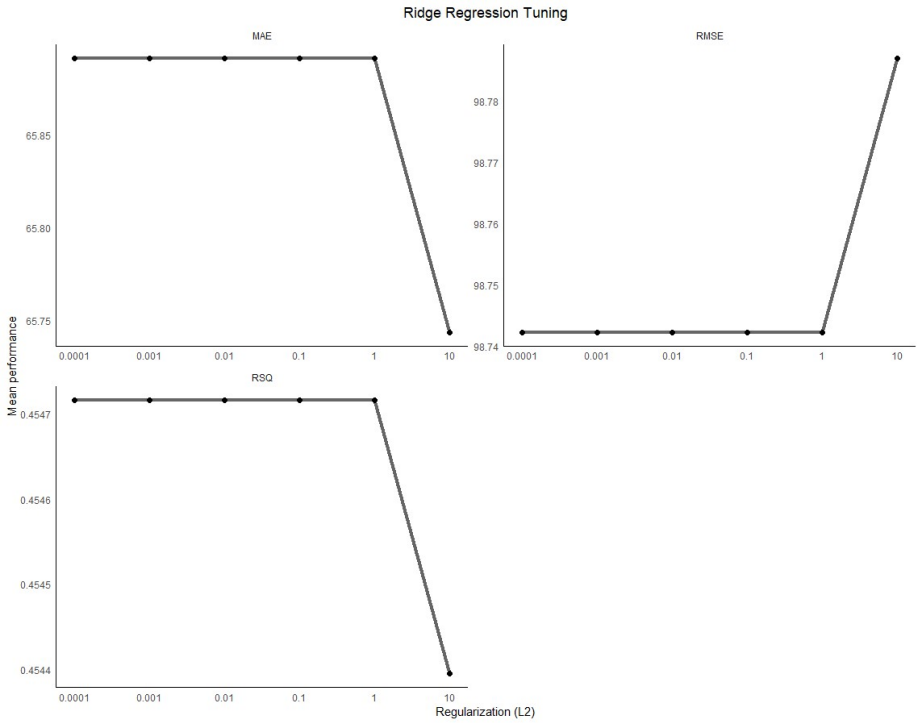

Figure B2- Ridge tuning results.

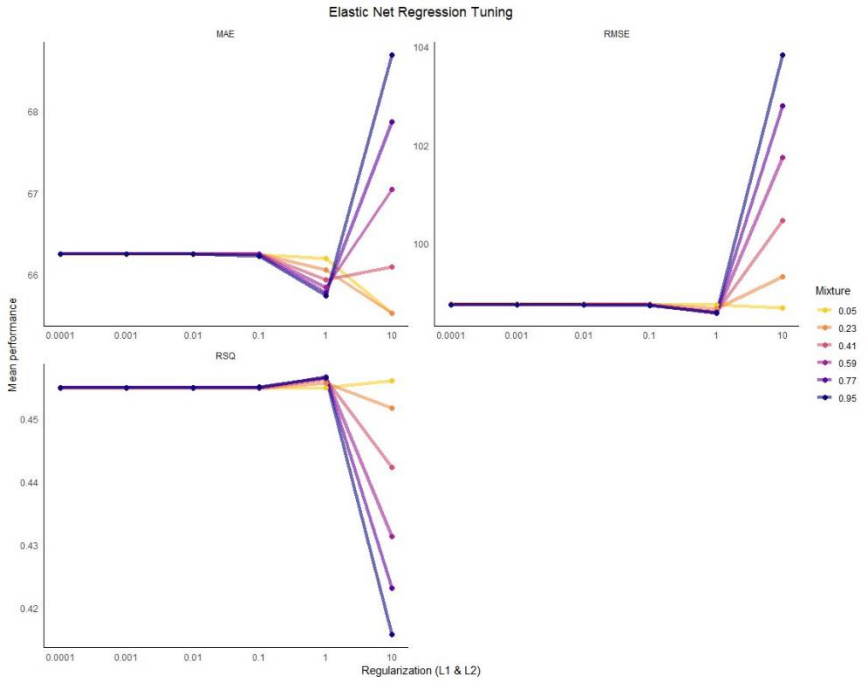

Figure B3- Elastic Net tuning results.

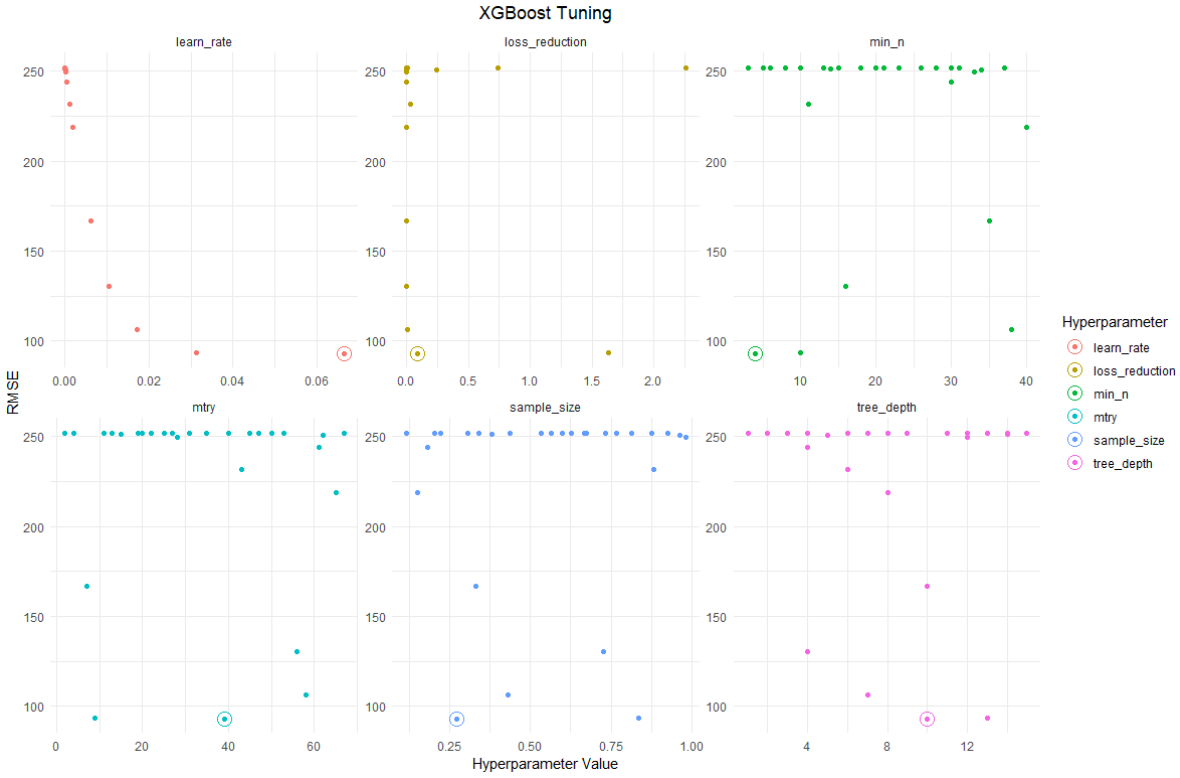

Figure B4- XGBoost tuning results

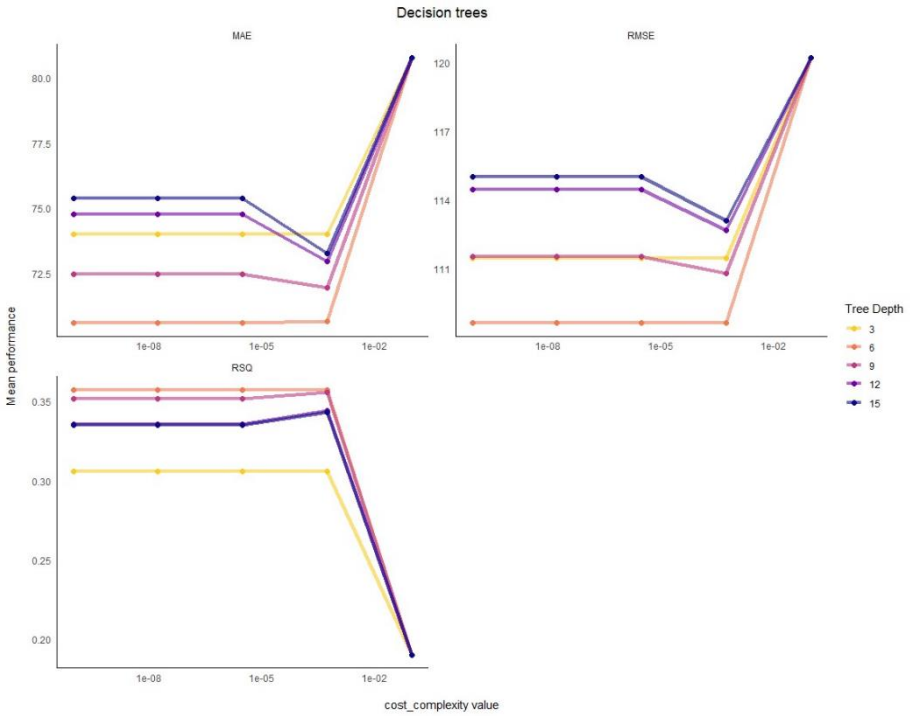

Figure B5- Decision Tree tuning results.

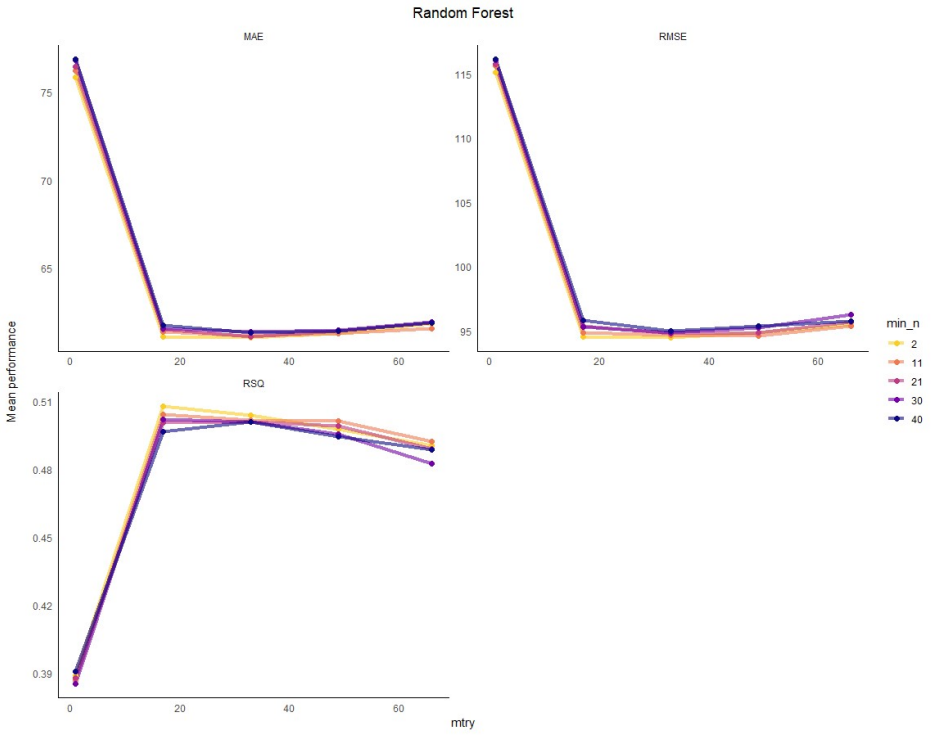

Figure B6- Random Forest tuning results

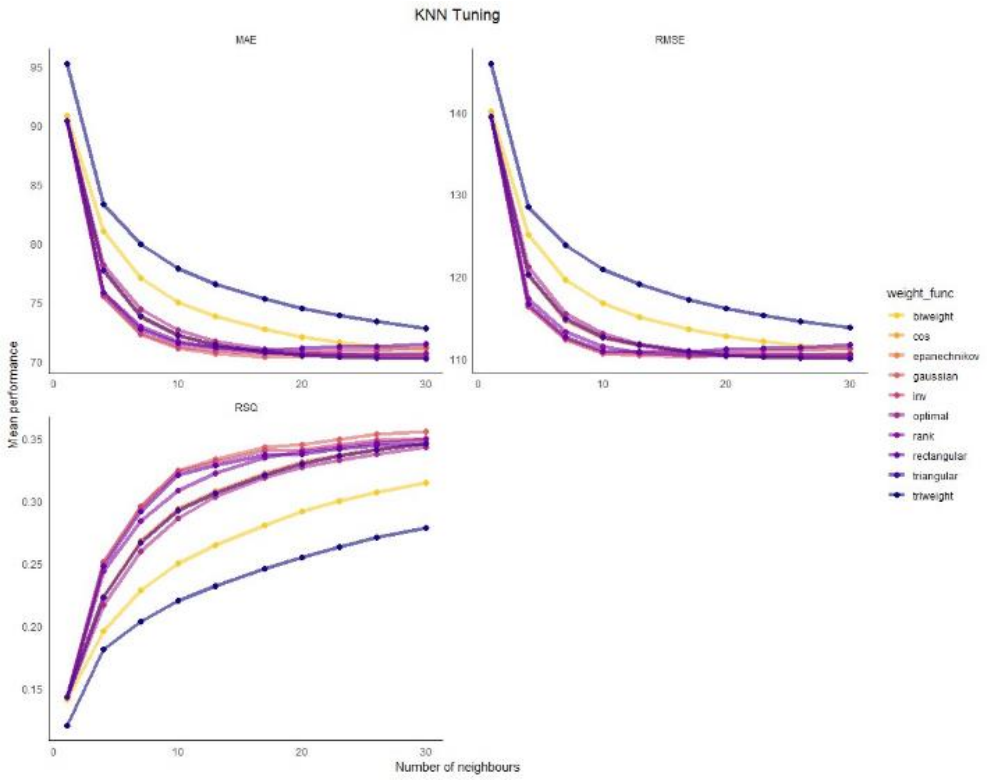

Figure B7- KNN tuning results.

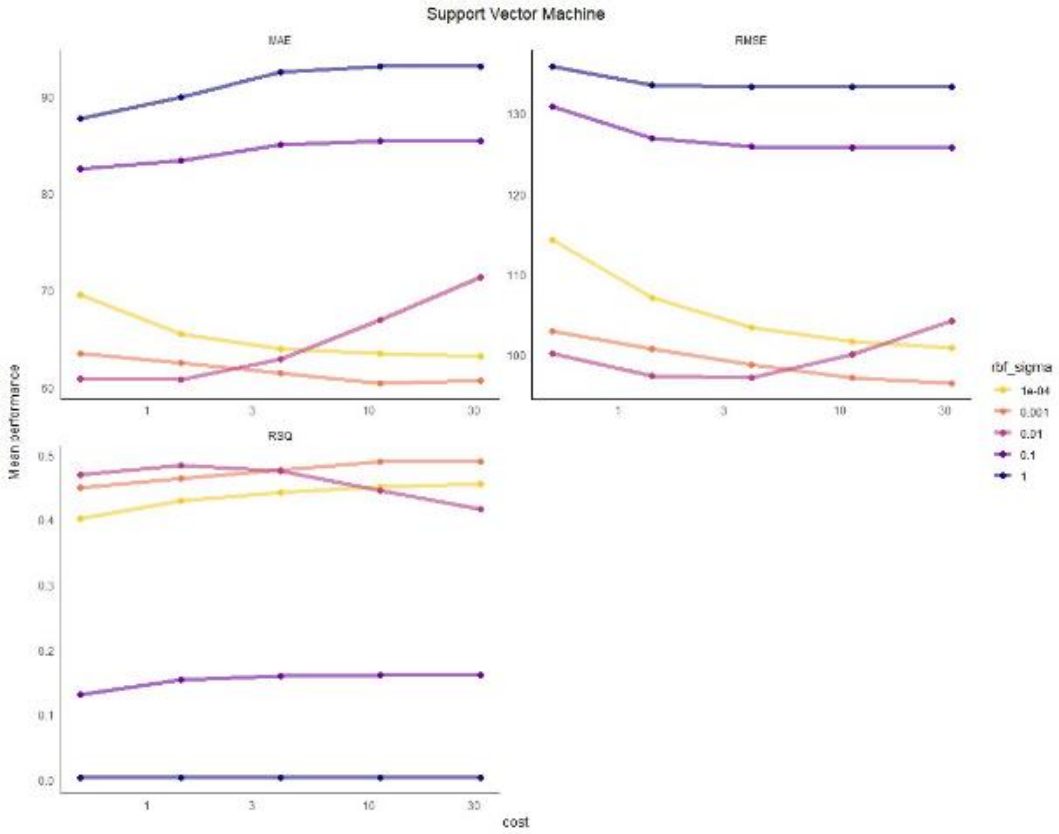

Figure B8- SVM tuning results

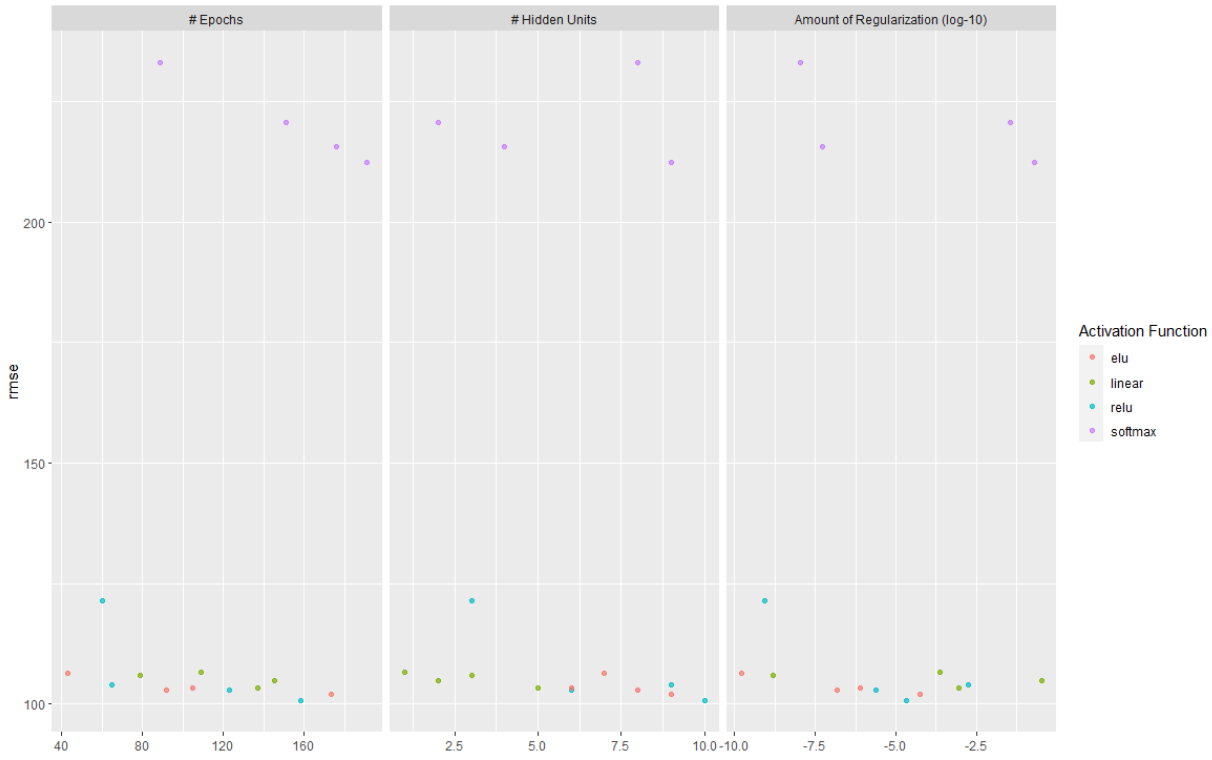

Figure B9- Neural Network tuning results.

Appendix C- Statistical tests

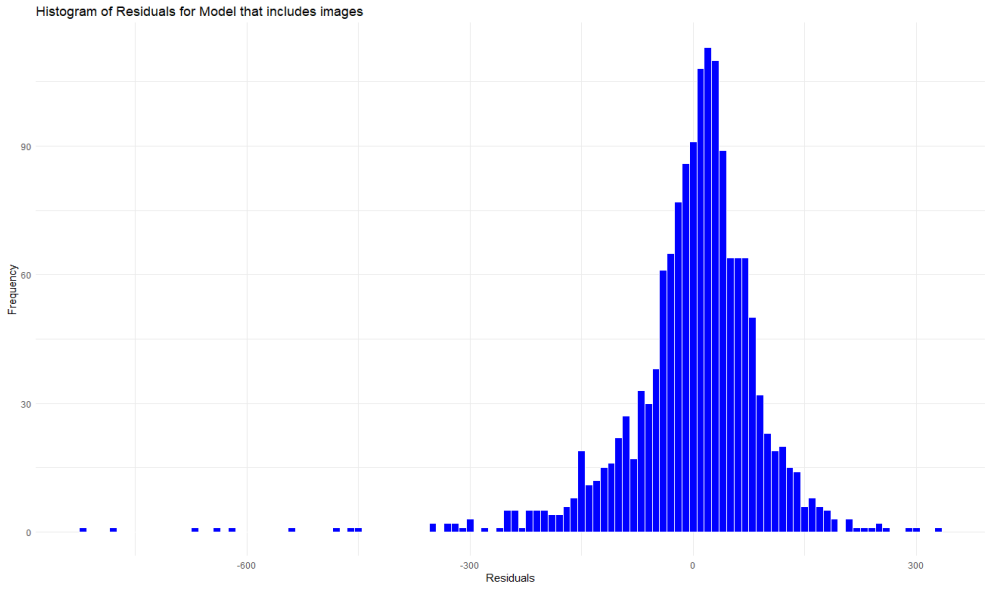

Figure C1- Histogram of residuals.

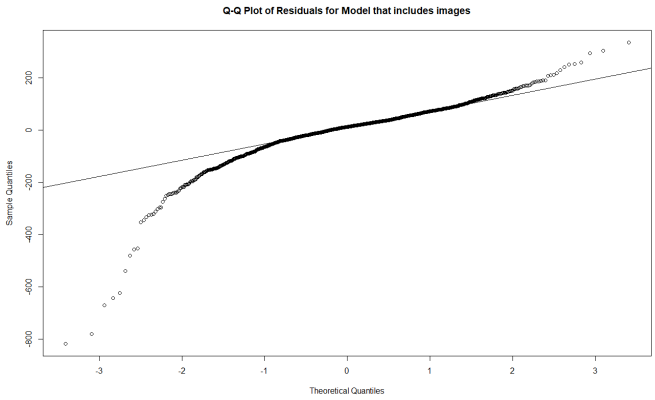

Figure C2- QQ-plot of residuals.

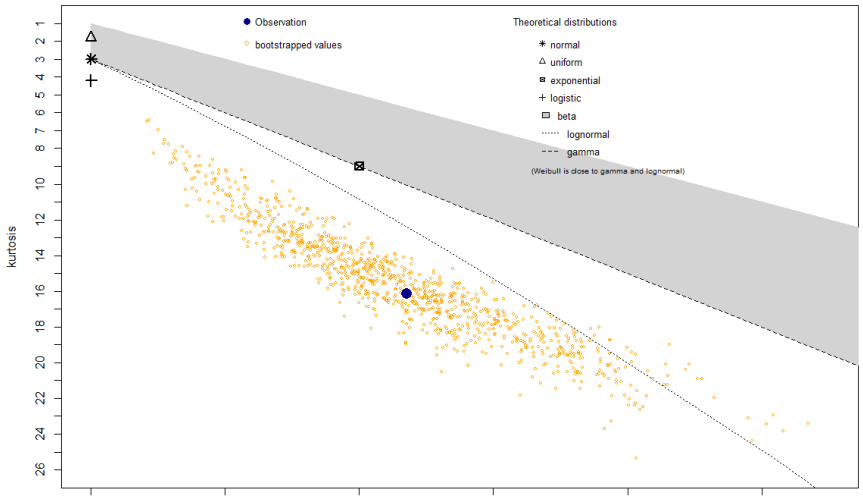

Figure C3- Cullen and Frey graph of residuals

Appendix D-Interpreting Models

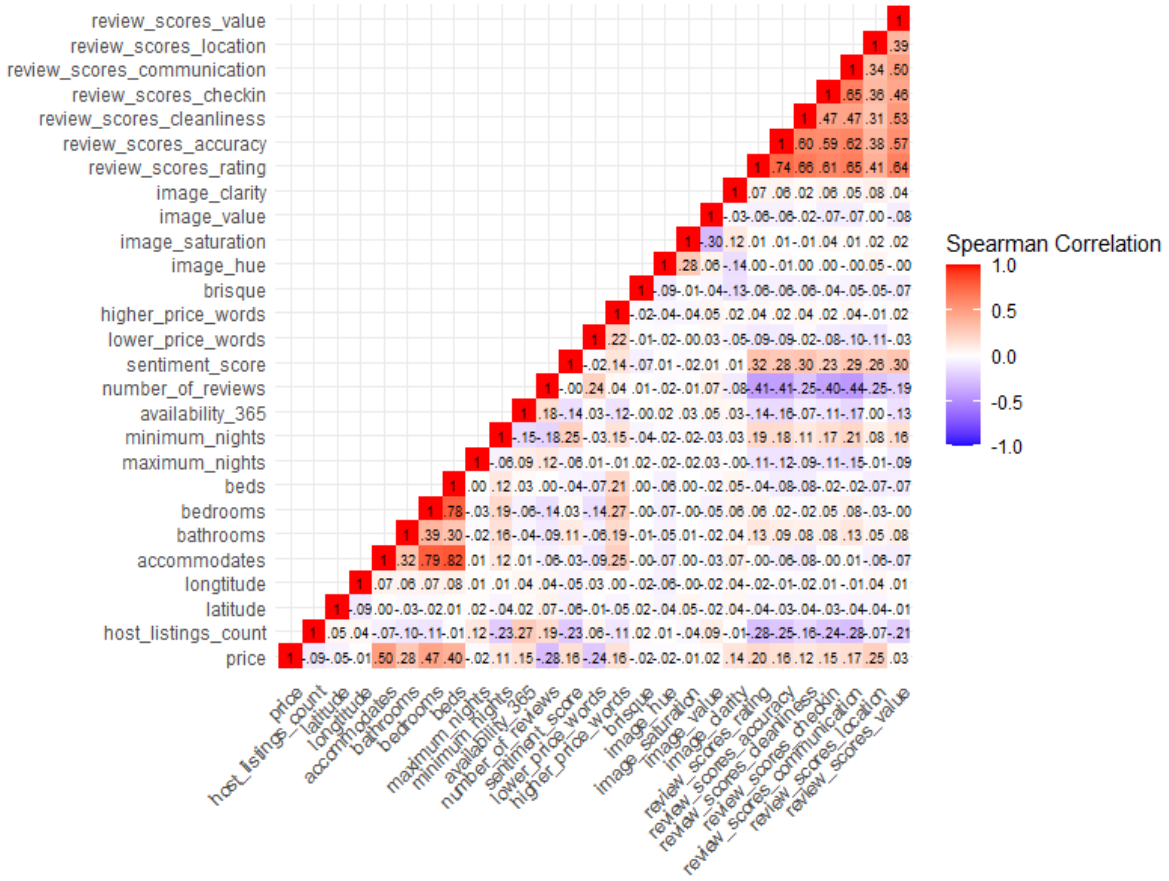

Figure D1- Correlation plot for continuous features.

Figure D2-SHAP values