University of Münster
Center for Quantitative Economics
Institute for Econometrics and Economic Statistics
Winter Term 2020/2021

Master Thesis

The Effect of Wind Turbines on House Prices in Germany - Evidence from a Machine Learning based Estimation Approach

Chairholder:	Prof. Dr. Mark Trede
Supervisor:	Manuel Stapper, M. Sc.

Submitted by:	Tobias Larysch
Field of Study:	Master Economics

Contents

List of Tables

List of Figures

Abbreviations

ASMD	Averaged standardized absolute mean difference
ATE	Average treatment effect
BCA	Baseline conditional average
BLP	Best linear predictor
CATE	Conditional average treatment effect
CI	Confidence interval
CLAN	Classification analysis
CV	Cross-validation
ENet	Elastic net
GATES	Sorted group average treatment effects
GHG	Greenhouse gas
HET	Heterogeneity coefficient
HTE	Heterogeneous treatment effect
IPTW	Inverse probability of treatment weighting
KS	Kolmogorov-Smirnov
LASSO	Least absolute shrinkage and selection operator
LIE	Law of iterated expectations
MaStR	Core Energy Market Data Register (*Marktstammdatenregister*)
ML	Machine learning
MSE	Mean squared error
OLS	Ordinary least squares
PS	Propensity scores
RCT	Randomized controlled trial
RF	Random forest
RSS	Sum of squared residuals
SGD	Stochastic gradient descent
VEIN	Variational estimation and inference

1 Introduction and Motivation

Fridays for Future, the Paris Agreement, carbon taxes - hardly any topic has been as present in the public and political debate in Germany in recent years as the issue of climate change. In an attempt to limit the ongoing global warming and temperature increase to a maximum of 2 °C, as stated in the Paris Agreement in 2015, Germany has committed itself to ambitious climate protection targets. Compared to 1990, the total greenhouse gas (GHG) emissions are to be reduced by 40 % until 2020, by 55 % until 2030 and by at least 80 % until 2050 (Umweltbundesamt, 2019). Up until now, Germany has already undertaken a lot of effort to reach those intended figures. The single most important sector for GHG emission reductions is energy generation from fossil fuels. By an extensive promotion of renewable energies via various regulatory instruments as, for example, relatively high fixed feed-in tariffs or feed-in priorities over electricity from conventional energy sources, it was possible to reduce the CO_2 emissions in this sector from 423.9 Mt in 1990 to 290.1 Mt in 2018 (Umweltbundesamt, 2020). At the same time, the share of renewable energy in total gross electricity consumption increased from 6.3 % in 2000 to 42.1 % in 2019 (BMWi, 2019). Nevertheless, Germany is expected to miss its self-imposed emissions targets in both 2020 and 2030. And although the target regarding electricity consumption from renewables in 2020 of 35 % was met (BMWi, 2020), the expansion of production capacities of renewable energies has slowed down in the last few years (BMWi, 2019). This is mostly due to a low number of new constructed wind power plants, especially onshore wind turbines. The number of new wind turbines in 2019 was actually the lowest in the last 20 years (Deutsche WindGuard, 2020). Since wind energy is by far the strongest driver of renewable energies in Germany, contributing more than 50 % to the overall share of renewables in total electricity consumption (BMWi, 2019), this decrease slows down the further expansion of renewable energy generation significantly.

One of the main reasons for this decline in the capacity expansion of energy from wind is that the construction of new turbines often meets with resistance. Although there is high public acceptance among citizens regarding wind energy in general (FA Wind, 2019), the development of new power plants is often accompanied by local protests and lawsuits due to perceived negative externalities induced by the respective turbines. For example, rotor blades make noise and may cast shadow flicker on nearby properties, the plants are illuminated to make them more visible to air traffic at night, and they may negatively affect the visual landscape they are built in. Therefore, in addition to the affordability of

the conducted measures, achieving the set emissions targets also depends on the affected citizens' acceptance of these measures to a large extent.

In addition to (only) perceived negative effects of wind turbines, a measure of actual negative externalities and costs induced by these is needed. Since the visual landscape or unaffectedness from wind turbines are no regular market goods, they are also not priced as such. In those situations, there are two ways used to determine such figures: either by stated preferences, e.g. via surveys, or by revealed preferences. One strategy to uncover revealed preferences is by estimating hedonic price models and then using the estimates on a specific attribute to gain insights about how this attribute is valued (Lang et al., 2014). Following this approach, a useful proxy of external costs of installed wind power plants can be constructed from their effects on house prices. If the existence of nearby wind power plants leads to a decrease in properties' values, this would constitute an empirically testable explanation for the local resistance often occurring when planning the construction of new turbines. At the same time, such an estimate could for example also be used to compensate affected citizens in order to reduce the faced opposition.

The empirical evidence assessing effects of wind turbines on nearby properties is mixed. For example, Hoen et al. (2011) use a difference-in-difference approach to model the price change of houses before and after the construction of wind turbines in the US and find no significant effects for both visibility of turbines and proximity to the respective properties. Lang et al. (2014) estimate a hedonic difference-in-difference model with distance bands as variables of interest for houses in Rhode Island and find no significant effects either. In contrast, using a similar approach for the Netherlands, Dröes and Koster (2016) estimate a price decrease of 1.4 % for houses within a 2 km radius and no effects for more than 2 km distance. Jensen et al. (2014) try to disentangle visual effects and effects from noise using techniques from spatial econometrics. They find a negative effect of up to 3 % resulting from visual pollution and a negative effect between 3 and 7 % from noise pollution. Furthermore, Gibbons (2015) studies the effects of wind turbines in England and Wales. The author uses a quasi-experimental difference-in-difference setting in which he compares a group for which newly constructed wind turbines are visible with a group in the same proximity for which the new power plants are hidden by the surrounding terrain. The analysis reveals significant negative effects of 5–6 % for houses within 2 km, of under 2 % between 2 and 4 km and no effects for larger distances than 4 km.

Focusing on studies for Germany, Sunak and Madlener (2016, 2017) find significant negative effects of nearby wind power plants on house prices for three cities in North Rhine-Westphalia using spatial hedonic pricing models and an indicator for the visual impact of wind turbines. Further, Frondel et al. (2019) estimate the effect of wind turbines on property prices with a linear regression model containing property features and locality characteristics as well as local and year fixed effects in a large sample covering all of Germany. The authors use multiple distance bands to the nearest wind turbine as their variables of interest and find significant negative effects, fading out with increasing distance and no effects after distances larger than 8 km. Additionally, they also report some evidence of effect heterogeneity. They use a machine learning (ML) model, which utilizes regression trees to estimate the treatment effects based on observed covariates. The results from this ML model are used to construct interaction terms in the linear model in a second step. They find stronger effects for houses built before 1950, and for properties in rural areas compared to houses in urban surroundings.

All in all, the empirical evidence is ambiguous, however the results are more strongly pointing towards negative effects of wind turbines on house prices. Moreover, very little is known about effect heterogeneity. The only study, which tries to provide evidence of varying effects, is by Frondel et al. (2019). Yet, the heterogeneity in the cited paper is modeled by simple interaction terms between two of the covariates and the treatment variables based on information from a preceding model. However, heterogeneity may work along several dimensions and hence may not be captured sufficiently by single interaction terms. Thus, a more systematic and detailed approach to the estimation of heterogeneous treatment effects is needed. This master thesis therefore aims to contribute to closing this gap in the literature. To do so, it is important to analyze if there exists substantial heterogeneity in the first place using a more systematic approach. Furthermore, this thesis aims at providing insights into how treatment effects vary among houses and which houses are affected the most and the least, and to what extent.

In order to find answers to the above stated questions and objectives, and thus to improve the understanding of the effects of wind turbines on property prices in more detail, this thesis proceeds as follows: Section 2 describes and explains the estimation strategy applied in the empirical analysis. Since this method heavily relies on machine learning techniques and algorithms, Section 3 provides a brief overview of the main ideas and principles of this field and introduces the algorithms used later on. Section 4 illustrates the different data sets used in the analysis and explains the conducted data cleaning and preprocessing steps.

Before the results are displayed and examined in Section 6, Section 5 discusses additional implementation details which must be considered with the use of the previously explained estimation approach. Section 7 summarizes the results and concludes.

2 Empirical Strategy

In order to answer the research questions stated in the previous section, a recently developed estimation strategy developed by Chernozhukov et al. (2018) is utilized. The authors provide an empirical approach to assess heterogeneous treatment effects (HTE) by estimating key features of the so-called conditional average treatment effect (CATE) function, which describes average treatment effects as a function of observed characteristics or covariates. These features are estimated in two steps: at first, machine learning algorithms are used to construct a proxy predictor of the CATE function, and secondly, this proxy predictor is used as input to weighted linear regressions in order to identify the parameters of interest. One of the strengths of this approach is that it makes no (possibly hard to prove) assumptions about unbiasedness or consistency of the proxy predictor. Moreover, in contrast to other methods relying on ML for causal effect estimation, the described strategy comes with statistically valid confidence intervals, and thus allows for valid statistical inference and can be used with any ML method, therefore it is not limited to a specific kind of algorithm.[1]

The key features of the CATE function the authors consider are its best linear predictor (BLP) using the ML proxy, sorted group average treatment effects (GATES), and the average characteristics of the most and least affected groups, called classification analysis (CLAN). This chapter describes the estimation procedure in detail and provides an explanation of the empirical approach used in this master thesis.[2]

2.1 Notation and CATE Proxy

Following the causal model of potential outcomes introduced by Rubin (1974), let $D \in \{0, 1\}$ be a binary variable indicating treatment status and Z be a vector of

[1] For methods which also rely on ML methods and algorithms, but are lacking valid confidence intervals see for example Foster et al. (2011); Imai and Ratkovic (2013); Nie and Wager (2017); Künzel et al. (2019). The estimation strategy developed by Athey and Imbens (2016) and Wager and Athey (2018) allows for statistically valid inference, however can only be used with a specific kind of tree-based algorithms.

[2] Sections 2.1 to 2.5 are taken from the project studies thesis submitted on April 04, 2020 with only minor changes and edits.

observed covariates. $Y(1)$ and $Y(0)$ denote potential outcomes under treatment and under no treatment. The baseline conditional average (BCA) function and the conditional average treatment effect function are defined as:

$$b_0(Z) := \mathbf{E}\big[Y(0) \mid Z\big] \tag{2.1}$$

$$s_0(Z) := \mathbf{E}\big[Y(1) \mid Z\big] - \mathbf{E}\big[Y(0) \mid Z\big]. \tag{2.2}$$

Let D be randomly assigned conditional on the covariates Z and let $p(Z)$ be the probability of being treated given the covariates, i.e. the propensity score. Further assume that $p(Z)$ is bounded away from zero and one:

$$D \perp\!\!\!\perp Y(1), Y(0) \mid Z \tag{2.3}$$

$$0 < p(Z) < 1. \tag{2.4}$$

Assumptions (2.3) and (2.4) are also referred to as the unconfoundedness and overlap assumptions in various settings (see for example Imbens and Wooldridge, 2009). Using the definitions of BCA and CATE from Equation (2.1) and Equation (2.2) the outcome function can be written as

$$Y = b_0(Z) + s_0(Z)D + U, \quad \mathbf{E}[U \mid Z, D] = 0. \tag{2.5}$$

The CATE proxy is denoted by $S(Z)$, the estimated BCA function by $B(Z)$.

In order to make inferential statements on the true CATE function $s_0(Z)$ based on the CATE proxy, $S(Z)$ has to be estimated in the first place. To do so, a separate model is fitted to the outcome in each of both groups using any kind of ML algorithm. In a second step, these models are used to predict the (hypothetical) outcomes under treatment and no treatment for the pooled sample of treated and control observations. Thus, the predicted outcome under no treatment corresponds to the estimated BCA function $B(Z)$. Lastly, the CATE proxy is constructed by taking the difference between the predicted outcome from the treatment group model and the predicted outcome from the control group model.

2.2 Best Linear Predictor

The first key feature of the CATE function Chernozhukov et al. (2018) consider is its BLP given the ML proxy. The solution to the problem of best linear approximation of $s_0(Z)$ using $S(Z)$ is given by

$$\text{BLP}\big[s_0(Z) \mid S(Z)\big] = \beta_1 + \beta_2\big(S(Z) - \mathbf{E}[S(Z)]\big), \tag{2.6}$$

with the coefficients

$$\beta_1 = \mathbf{E}\big[s_0(Z)\big] \text{ and } \beta_2 = \frac{\mathbf{Cov}\big(s_0(Z), S(Z)\big)}{\mathbf{Var}\big(S(Z)\big)}. \tag{2.7}$$

The authors show that these coefficients β_1 and β_2 solving

$$(\beta_1, \beta_2)' = \arg\min_{b_1, b_2} \mathbf{E}\big[s_0(Z) - b_1 - b_2 S(Z)\big]^2 \tag{2.8}$$

can be identified from the weighted linear regression:

$$Y = \alpha' X_1 + \beta_1\big(D - p(Z)\big) + \beta_2\big(D - p(Z)\big)\big(S(Z) - \mathbf{E}[S(Z)]\big) + \epsilon, \tag{2.9}$$

$$\text{with } X_1 = \big[1, B(Z), S(Z)\big]' \text{ and weights } w(Z) = \big[p(Z)(1 - p(Z))\big]^{-1}.^{3}$$

Since the parameter β_1 from this weighted regression is equal to $\mathbf{E}[s_0(Z)]$, it corresponds to the average treatment effect. β_2, on the other hand, corresponds to the coefficient of a simple linear regression of $s_0(Z)$ on $S(Z)$ and can be utilized to evaluate how well $S(Z)$ approximates $s_0(Z)$ as well as to gain insights about treatment effect heterogeneity. If $\beta_2 = 0$, the CATE proxy and the true function are completely uncorrelated. Additionally, β_2 would also be zero if there was no heterogeneity, and thus $s_0(Z)$ was a constant. In contrast to this, $\beta_2 \neq 0$ implies that there is substantial heterogeneity and that it can be predicted by the proxy $S(Z)$. Therefore, testing for heterogeneous treatment effects corresponds to testing the hypothesis that $\beta_2 \neq 0$.

2.3 Sorted Group Average Treatment Effects

In addition to the BLP of the CATE function, the authors provide a strategy to identify groups of observations, which are more or less affected by the treatment, and to estimate ATEs in those groups. To do this, the observations are first sorted according to the predicted ML proxy $S(Z)$ and then divided into k non-overlapping groups $G_1, G_2, ..., G_K$ of arbitrary sizes. They might for example be

[3] A proof for the above stated properties can be found in Appendix A.1.

chosen by quantiles of the proxy predictor. The parameters of interest are the average treatment effects in these groups, called sorted group average treatment effects (GATES) which are denoted by

$$\gamma_k = \mathbf{E}\big[s_0(Z) \mid G_k\big] \text{ for } k = 1, ..., K. \tag{2.10}$$

Since the observations are sorted by $S(Z)$, it seems reasonable to assume what Chernozhukov et al. (2018) call the monotonicity assumption. Monotonicity implies that the GATES increase or decrease with the groups, i.e.

$$\mathbf{E}\big[s_0(Z) \mid G_1\big] \leq \mathbf{E}\big[s_0(Z) \mid G_2\big] \leq \cdots \leq \mathbf{E}\big[s_0(Z) \mid G_K\big]. \tag{2.11}$$

These parameters can again be identified by a weighted linear regression using the same weights as in the estimation of the BLP coefficients from Section 2.2. Specifically, the regression has the following form:

$$Y = \alpha' X_1 + \sum_{k=1}^{K} \gamma_k \big(D - p(Z)\big)\mathbf{1}(G_k) + \nu. \tag{2.12}$$

As before, X_1 may contain a constant, the estimated BCA function $B(Z)$ and the CATE proxy $S(Z)$. $\mathbf{1}$ denotes the indicator function being 1 if an observation belongs to group k. Chernozhukov et al. (2018) show that the coefficients γ_k from the weighted linear regression stated in (2.12) can be interpreted as the average treatment effects in the respective kth group:

$$\gamma_k = \mathbf{E}\big[s_0(Z) \mid G_k\big].^4 \tag{2.13}$$

The resulting GATES can again be used to test for treatment effect heterogeneity by testing whether

$$\mathbf{E}\big[s_0(Z) \mid G_1\big] = \mathbf{E}\big[s_0(Z) \mid G_2\big] = \cdots = \mathbf{E}\big[s_0(Z) \mid G_K\big]. \tag{2.14}$$

Rejecting this hypothesis implies that the ATE is not the same in all groups and that there are at least some groups which differ in terms of their ATEs. Therefore, a rejection of (2.14) can be interpreted as evidence in favor of heterogeneous treatment effects.

[4] A proof of this statement is provided in Appendix A.2.

2.4 Classification Analysis

Building on the estimation results from the analysis of GATES as described in Section 2.3, the construction of groups can further be used to gain insights about the average characteristics of the corresponding observations. The authors suggest focusing on the most and least affected groups based on the ML proxy. Given the monotonicity assumption introduced in property (2.11), these are determined by the groups G_1 and G_K. This allows for the identification of observations who are affected the most and the least from the analyzed treatment. This approach is called classification analysis (CLAN). The target parameters are denoted by

$$\delta_1 = \mathbf{E}\big[g(Y, Z) \mid G_1\big] \text{ and } \delta_K = \mathbf{E}\big[g(Y, Z) \mid G_K\big] \tag{2.15}$$

and can be estimated by computing the averages of the observed variables for the observations from the groups G_1 and G_K separately. In a second step it can be tested if those average characteristics differ significantly between the two groups.

2.5 Estimation Uncertainty and Inference

The estimation strategies described in this section rely on splitting the data into two approximately equally sized subsamples to avoid overfitting the data, which else may cause problems using highly flexible ML algorithms. One half of the data, the auxiliary sample $Data_A$, is used to tune and train the ML models, and thus to construct the ML proxy, while the other half, the main sample $Data_M$, is used to actually estimate the parameters of interest. This approach comes with two sources of uncertainty regarding the estimates: The first one is estimation uncertainty conditional on the sample split. This is standard in any estimation procedure and already accounted for by reporting regular standard errors, confidence intervals and p-values. The second source arises from the uncertainty in the random splitting itself. Different data splits may yield different parameter estimates, and thus, conditional on the data, the estimates and confidence intervals can still be regarded as random variables. Chernozhukov et al. (2018) develop methods to also account for such splitting uncertainty, called variational estimation and inference methods (VEIN). They suggest using many different data splits and repeating the estimation procedure for each split in order to increase robustness of the results. Specifically, they recommend to report medians

for each estimated parameter over all data splits. Additionally, they propose to report confidence intervals of the following form:

$$\big[lower, upper\big] = \big[\overline{Med}(Lower_A \mid Data), \underline{Med}(Upper_A \mid Data)\big]. \tag{2.16}$$

$\overline{Med}$ and $\underline{Med}$ are defined as the upper and lower median:[5]

$$\underline{Med}(X) := \inf\big\{x : \Pr(X \leq x) \geq 1/2\big\} \tag{2.17}$$

$$\overline{Med}(X) := \sup\big\{x : \Pr(X \geq x) \geq 1/2\big\} \text{ and} \tag{2.18}$$

$$Med(X) := \big(\underline{Med}(X) + \overline{Med}(X)\big)/2. \tag{2.19}$$

Further, the confidence level is discounted from $1-\alpha$ to $1-2\alpha$ to account for the splitting uncertainty. The authors also develop adjusted p-values for hypothesis testing. Using those p-values, a null hypothesis is rejected at significance level α if for at least 50 % of the data splits the p-values conditional on the specific split are below $\alpha/2$:

$$\Pr\big(p_A \leq \alpha/2 \mid Data\big) \geq 1/2 \text{ or } p_{.5} = \underline{Med}\big(p_A \mid Data\big) \leq \alpha/2. \tag{2.20}$$

The p-values $p = 2p_{.5}$ are then called sample splitting-adjusted p-values. Applying these VEIN methods to adjust the obtained confidence intervals and p-values allows to account for the additional uncertainty induced by data splitting. Chernozhukov et al. (2018) are the first authors to develop such methods.

2.6 Choosing the Best ML Method

In addition to the estimation details, the authors also provide auxiliary information on how to choose which ML should be used for constructing the CATE proxy. For this purpose, they derive information criteria for both steps, estimating the BLP as well as estimating the GATES parameters, which can be used as an indicator on how well the proxy approximates the true CATE function.

Specifically, the authors propose to choose the ML algorithm which maximizes:

$$\Lambda := |\beta_2|^2 \mathbf{Var}[S(Z)], \tag{2.21}$$

with β_2 being the HTE coefficient from the BLP estimation. They further note that this is the same as maximizing the correlation between the ML proxy $S(Z)$ and the true CATE function $s_0(Z)$.

[5] For example, if X is uniform in {1,2,3,4,5,6}, then $\underline{Med}(X) = 3$ and $\overline{Med}(X) = 4$.

Alternatively, the best ML method can also be found by considering the following criterion:

$$\overline{\Lambda} = \mathbf{E}\Big[\sum_{k=1}^{K} \gamma_k \mathbf{1}(S \in I_k)\Big]^2 \tag{2.22}$$

$$= \sum_{k=1}^{K} \gamma_k^2 \Pr(S \in I_k), \tag{2.23}$$

with γ_k being the GATES parameters. Maximizing this term corresponds to maximizing the R-squared of a regression of the true function $s_0(Z)$ on the demeaned proxy $\overline{S}(Z)$, thus leaving out the constant term.

By simply applying multiple ML algorithms to the data and estimating all the parameters, these criteria can then be used to choose the method which yields the best approximation, and thus the most accurate and credible results.

3 Machine Learning: Overview and Algorithms

Since the empirical analysis conducted in the context of this master thesis heavily relies on models and algorithms from the field of machine learning, this section provides an introduction to and overview of this topic. At first, Section 3.1 introduces the main concepts and general ideas of ML as well as basic terminology. In general, there are no theoretical and practical results, which imply that a specific algorithm outperforms every other algorithm or method for any dataset (Athey and Imbens, 2019). Therefore, the analysis in this thesis utilizes multiple algorithms to construct the CATE proxy, and then the best performing method is chosen according to the criteria stated in Section 2.6. The remaining sections thus provide an overview of the specific algorithms and methods used later on.

3.1 Main Concepts and Ideas

The term machine learning refers to a broad set of techniques and methods used to estimate functions or detect patterns in data without explicitly programming rules or making assumptions about the functional form in advance (see for example James et al., 2013). ML can be divided into two main areas: supervised and unsupervised learning. Unsupervised learning comprises algorithms that are constructed to find distinct groups based on observed characteristics and to cluster the data accordingly, whereas supervised learning uses a set of covariates to predict an observed outcome. Supervised learning can further be divided into regression and classification problems, i.e. building models that either pre-

dict continuous outcomes or classify observations into distinct categories (Varian, 2014).[6] In contrast to the field of econometrics, which is mostly concerned with the consistent and unbiased estimation of parameters of interest in order to isolate causal effects, ML focuses primarily on prediction problems. One of its main goals is to construct, or train, models and estimators that generalize well and that have high predictive power, even for new, unseen data (Mullainathan and Spiess, 2017). Therefore, ML relies on out-of-sample predictive power as a goodness-of-fit measure rather than in-sample measures, as for example R-squared (Athey and Imbens, 2019). The data sets the ML models are fitted on are called training data or training sets, while the data used for assessing the out-of-sample performance is called test data or test set.

Since the functional form of the estimated models is usually not set in advance and has to be inferred from the data, ML algorithms often are able to fit highly flexible functions resembling the input data very closely. However, such models usually suffer from poor predictive performance since they do not only capture the systematic information provided by the covariates or predictors, but also pick up all the specific characteristics and noise as well (Mullainathan and Spiess, 2017). In ML parlance, this is called overfitting. In order to generalize well, a model needs to be able to produce predictions with only small errors for new data. It can be shown that the expected prediction error, more specifically the mean squared error (MSE), can be decomposed into a sum of three components: an irreducible error, the squared bias of the estimator, and its variance. Let Y be the target in a regression setting with $Y = f(x) + \varepsilon$, $\mathbf{E}[\varepsilon] = 0$ and $\mathbf{Var}[\varepsilon] = \sigma_\varepsilon^2$. $\hat{f}(x)$ denotes the fitted estimation function for Y. The expected test set MSE is then given by:

$$\mathbf{E}\big[(Y - \hat{f}(x))^2\big] = \mathbf{Var}[\hat{f}(x)] + \text{Bias}[\hat{f}(x)]^2 + \mathbf{Var}[\varepsilon]. \tag{3.1}$$

In ML, the term bias refers to the amount by which the average of an estimate deviates from the true mean, while the term variance specifies how sensitive the estimator is to the specific training sample (James et al., 2013). In order to construct models that produce decent estimates for new data, both the bias and the variance should be as small as possible. However, highly flexible models tend to have low bias at the expense of higher variance, while less complex models achieve a low variance, but result in higher bias. Therefore, minimizing the ex-

[6] Since the ML algorithms in the context of this master thesis are applied in regression settings only, they are also explained with regard to regression tasks. All of these methods however work analogously in classification settings with some tweaks and changes in the optimization functions and mathematical details, however the basic ideas and concepts still hold.

pected test MSE results in the conflict of simultaneously minimizing the bias and the variance of an estimator. Since a decrease in one property often leads to an increase in the other, this is also called the bias-variance trade-off. Whether a more flexible model would improve or worsen the out-of-sample prediction performance in comparison to a less flexible model therefore depends on the relative rate of change of the bias and variance.

In order to find models that are still able to detect possible nonlinearities, but do not suffer from too high variance, most ML algorithms use some form of regularization (Athey, 2018). Regularization can be thought of as a penalty term for excessive model complexity (Varian, 2014). The amount of regularization depends on the specific algorithm and is often determined by so-called hyperparameters. In contrast to the model parameters, which can be directly estimated from the data, such hyperparameters must be set manually in advance (Mullainathan and Spiess, 2017). These hyperparameters can either be chosen using heuristics or, which is standard in applied ML, can be explicitly determined in a data-driven manner, using out-of-sample predictive performance of the model (Athey and Imbens, 2019). One of the most frequently applied techniques to do so is called k-fold cross-validation (CV). In k-fold CV, the data is first split into k subsamples or folds. The hyperparameters are fixed and the model is fitted to all the data from $k-1$ folds. Subsequently, the model is used to predict the outcome for the data from the k-th fold, which is not used to train the model, and a measure of error between predicted and actual values is computed. The procedure is repeated until all folds have been used as test data sets. This iterative approach then results in k error measures which are averaged to obtain a single measure of error. Eventually, the whole process is repeated with different hyperparameter settings in order to find those parameters that result in the lowest error, and thus in the best predictive properties for unseen data. This way of empirically searching for the best hyperparameters by testing the models' out-of-sample prediction performance is also called hyperparameter tuning or model tuning. Common values for the number of folds k are 5, 10 or $n-1$, also called leave-one-out CV (Varian, 2014).

Based on the above stated points, Mullainathan and Spiess (2017) provide the following summary for most ML methods: a ML algorithm usually consists of a function class $\mathcal{F}$ and a regularizer $\mathcal{R}(f)$ specifying the complexity of a model. The estimation and selection of the final model constitutes a two-step approach: First, conditional on the chosen hyperparameters, the model is fitted by minimizing

some error or loss-function. And second, the optimal level of regularization is estimated using empirical out-of-sample testing and tuning.

3.2 Regularized Linear Models

The first class of ML methods considered in more detail and applied in the estimation later on are regularized linear models. The traditional method in econometrics used to estimate coefficients in linear models is by minimizing the sum of squared residuals, i.e. the squared deviations between predicted and actual values. This approach is known as ordinary least squares (OLS) estimation and, considering N observations and P predictors or regressors, yields the following minimization problem:

$$\hat{\beta}_{OLS} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2 \right\}. \tag{3.2}$$

A commonly used approach to regularize linear models is to impose complexity constraints on the size and the number of the coefficients β in order to penalize more complex models. The aim of this technique is to add a small bias to the model compared to standard linear regression in order to reduce the variance by more than the additional bias, which then results in a decrease of the total expected out-of-sample test error. The coefficients of such regularized linear models solve minimization problems of the following form (Athey and Imbens, 2019):

$$\hat{\beta} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{P} |\beta_j|^q \right\}. \tag{3.3}$$

λ denotes a hyperparameter controlling the amount of regularization. Two common choices for q are $q = 1$ and $q = 2$. For the case of $q = 2$ such linear models are also called ridge regression and the coefficients solve:

$$\hat{\beta}_{ridge} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{P} \beta_j^2 \right\}. \tag{3.4}$$

Due to the quadratic penalty added to the loss function, large coefficients are strongly penalized. This leads to a decrease in the magnitude of the coefficients, they are shrunk towards zero. Because of this property, regularized linear models that optimize objective functions as stated in Equation (3.3) are also called shrinkage methods (James et al., 2013). Shrinking the coefficients towards zero makes the fitted model less sensitive to the specific data, and thus the model less

likely to overfit. The hyperparameter λ determines the intensity of the coefficient shrinkage; high values of λ lead to stronger shrinkage. With $\lambda = 0$, the ridge regression simply corresponds to linear regression via OLS.

Another common choice for q is $q = 1$. In this case, the coefficients are obtained by solving the minimization problem

$$\hat{\beta}_{LASSO} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{P} |\beta_j| \right\}. \tag{3.5}$$

This approach is also called the least absolute shrinkage and selection operator (LASSO). Similar to ridge regression, the LASSO also leads to shrinkage of estimated parameters towards zero. However, while coefficients from ridge regressions can only approximate zero, coefficients obtained from LASSO regressions can be set to be exactly zero if the hyperparameter λ is sufficiently large (Hastie et al., 2009). Therefore, the LASSO can also be used for variable selection and may result in sparse solutions, i.e. coefficient matrices with many entries being exactly zero (Varian, 2014). This property makes the LASSO an attractive algorithm in very high dimensional settings with more variables than observations (Tibshirani, 1996). Due to the choices for q, the ridge constraint is also called L_2-penalty, while the constraint imposed on the minimization problem solved by the LASSO is called L_1-penalty.

However, compared to ridge regression, the LASSO has got some limitations and drawbacks. For example, if there are groups of correlated variables in the data, it usually selects one of those variables kind of randomly, without caring which of the variables actually enters the model. Furthermore, in settings with highly correlated predictors, ridge regression tends to outperform the LASSO (Zou and Hastie, 2005). In order to mitigate such issues but prevail the LASSO's advantages at the same time, Zou and Hastie (2005) developed the elastic net algorithm. This algorithm combines both the ridge regression L_2-penalty and the LASSO L_1-penalty. The obtained coefficients solve the following minimization problem:

$$\hat{\beta}_{enet} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{P} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{P} \left(\alpha\beta_j^2 + (1-\alpha)|\beta_j|\right) \right\}. \tag{3.6}$$

All the before mentioned methods, standard linear regression, ridge regression and the LASSO, can be seen as special cases of elastic net regression. With $\lambda = 0$, the elastic net simply becomes linear regression with OLS. For $\alpha = 1$, the elastic

net corresponds to ridge regression and for $\alpha = 0$ it equals the LASSO. Elastic net regression is also able to set coefficients to zero, and thus to perform variable selection (Varian, 2014). Doing so, the sparsity of the solution found by the elastic net increases with $\alpha \to 1$. In contrast to the LASSO however, elastic net does not suffer from randomly choosing a single variable from a group of correlated predictors and ignoring the rest. Instead, it assigns highly correlated predictors very similar coefficients and selects all variables from the specific group.[7] This characteristic is called groupwise selection. For $\alpha = 1 - \epsilon$ with very small $\epsilon > 0$, the solution is similar to the solution obtained using LASSO regression, but only the elastic net preserves the groupwise selection property (Friedman et al., 2010).

Due to its strengths and advantages compared to both ridge regression and LASSO regression, especially its high predictive power, the elastic net is chosen as the first ML algorithm to construct the CATE proxy in this thesis.

3.3 Tree-based Methods

In the next section, two ML algorithms are introduced which are based on decision trees. Thus, the basic concept and idea of decision trees is illustrated at first, followed by an explanation of two algorithms using such decision trees and additional techniques for the construction of prediction models: Random forests and the XGBoost algorithm.

3.3.1 Decision Trees

Decision trees were introduced by Breiman et al. (1984) to be used in both regression and classification settings, thus being called classification and regression trees. In contrast to the regularized linear models explained in the previous section, they belong to the family of non-parametric methods. Decision trees, and tree-based methods in general, are used to partition the covariate space of some input data into distinct, non-overlapping regions and estimate a simple model for the outcome in each of those, usually the average of outcomes over all observations in that specific region (Athey and Imbens, 2019).

To get an understanding of how the trees are constructed, or grown, assume that there is some input data with N observations consisting of a continuous outcome y and p predictors, i.e. (x_i, y_i) for $i = 1, 2, \ldots, N$ and $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$.[8]

[7] For negatively correlated predictors the coefficients differ only in terms of their sign.

[8] The notation used in this subsection borrows from Hastie et al. (2009).

The goal is to divide the whole dataset into J regions R_J, such that the sum of squared residuals (RSS) is minimized, i.e.:

$$\min \sum_{j=1}^{J} \sum_{i:x_i \in R_j} (y_i - \hat{y}_{R_j})^2. \tag{3.7}$$

Since it is computational infeasible to examine every single possible partition of the predictor space, a top-down approach called recursive binary splitting is used instead. It is referred to as top-down because the algorithm starts at the top of the tree, the so-called root, and from there splits the data into multiple subgroups using binary splitting along the covariates. At the root, the whole sample still belongs to a single region and the total RSS amounts to

$$RSS_{root} = \sum_{i=1}^{N} (y_i - \bar{y})^2, \tag{3.8}$$

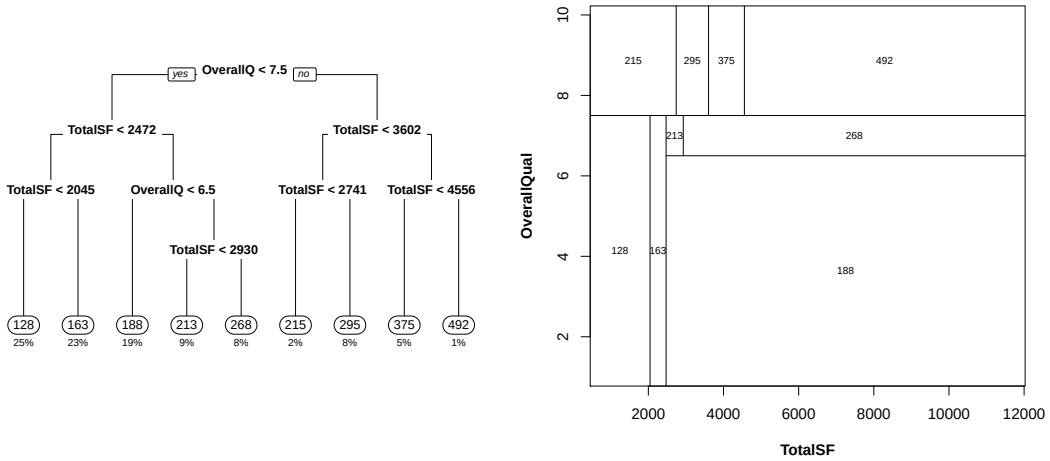
$$\text{with } \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i. \tag{3.9}$$

From there, the sample is split into two subsamples. These splitting points are called nodes or internal nodes. Previous nodes are referred to as parent nodes, succeeding nodes as child nodes. The final nodes the observations end in are called terminal nodes or leaves. At each node, the splitting criterion is chosen in such a way that the partition into the new regions results in the largest possible reduction in the RSS. Thus, at the root node the algorithm searches for the predictor X_j and a threshold value s to divide the data into two regions $R_1(j, s) = \{X \mid X_j < s\}$ and $R_2(j, s) = \{X \mid X_j \geq s\}$ such that

$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 \quad + \sum_{i:\, x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2 \tag{3.10}$$

is minimized, with $\bar{y}_{R_1}$ and $\bar{y}_{R_2}$ being the average outcomes in the two regions (Athey and Imbens, 2019). The threshold value s is found by first sorting the observations along each predictor and iteratively splitting the sample at each observed value. The reductions in RSS are calculated and finally, the specific split resulting in the largest RSS reduction is chosen. This binary splitting procedure is then repeated for any such constructed subsample, until some kind of stopping criterion is reached, which is typically after the number of observations in all leaves falls below a predefined minimum. At each node, the best split is chosen according to the reduction of RSS at this specific split, without looking for splits which might result in lower overall RSS at subsequent splitting points. Therefore,

recursive binary splitting is also called a greedy top-down approach (James et al., 2013). When the stopping criterion is reached and the tree is grown, predictions for new observations are then made by simply predicting the mean outcome of all training observations from the region R_j the new datapoint falls into (Athey and Imbens, 2019).



(a) Regression Tree **(b)** Partition Plot

Figure 1: Visualization of a regression tree and the according partition plot

Figure 1 shows an example of a simple regression tree predicting house prices in $ 1,000 using an indicator of the houses' overall quality (OverallQual), ranging from one to ten, and its total area in square feet (TotalSF).[9] The whole sample is first split based on the variable OverallQual and the threshold of 7.5, thus houses with higher quality are assigned right and houses with lower quality are assigned left (Figure 1a). The sample is then further split based on similar binary splitting rules. The terminal nodes contain the predicted house value for the (new) observations being assigned to the respective leaf. The percentage numbers below the leaves give the fraction of training observations assigned to the specific region. Figure 1b plots TotalSF against OverallQual and shows the partitioning and the predicted values resulting from the tree illustrated in Figure 1a.

Since both, the final predictions produced by decision trees as well as all splits depend on the splits constructed before, regression trees constitute a highly interactive function class (Mullainathan and Spiess, 2017). This may easily lead to overfitting when using many splits, and thus to very large trees. In general, the

[9] The data used for this illustration is a cleaned and reduced version of a dataset provided in the context of a well-known data science competition from the website www.kaggle.com and was retrieved via the following link: https://bit.ly/2OejeBI.

lower the number of splits, the lower the variance of the tree, but the higher the bias. One approach to decision tree regularization would be to stop further splitting the subsamples, when the RSS reduction from a considered split is smaller than some predefined threshold. However, this may lead to missed subsequent splits resulting in error reductions which are larger than this threshold (Hastie et al., 2009). To prevent this, at first a large, fully grown tree is constructed and then cut back by removing some of the internal nodes. This technique is called tree pruning. The most common approach to prune decision trees is by applying cost complexity pruning or weakest link pruning. Let T_0 denote a large tree, grown until some stopping criterion is met, e.g. a minimum terminal node size, and let $T \subset T_0$ denote a subtree obtained from pruning T_0. Further, let m denote the terminal nodes partitioning the sample into regions R_m, N_m the number of observations in leaf m and $|T|$ the number of terminal nodes in tree T. The average in-sample error in every leaf is then given by:

$$Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - \bar{y}_{R_m})^2. \tag{3.11}$$

Cost complexity pruning then finds the subtree $T \subset T_0$ that minimizes the cost complexity criterion given by:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|. \tag{3.12}$$

Here α constitutes a hyperparameter regularizing the complexity and size of the decision tree. Large trees have a higher number of terminal nodes $|T|$, and thus are penalized more strongly. Intuitively, the additional error reduction from further splitting, and hence a higher number of terminal nodes $|T|$, must be larger than the penalty resulting from that additional leaf. Therefore, the cost complexity criterion induces a trade-off between complexity as well as in-sample fit and the generalizability of the estimator (Hastie et al., 2009). Large α-values tend to produce smaller trees, while smaller values result in larger trees. For $\alpha = 0$, T will be equal to the fully grown tree T_0.

In general, decision trees have the advantage that they can easily be explained to others as well as interpreted due to their visual structure resembling a tree with binary decisions (see also Figure 1). The predicted values for new data are simply the average of training data outcomes in the leaf, in which the new observations fall into. However, single decision trees usually suffer from relatively low predictive power compared to other more sophisticated ML methods. Additionally,

although easily explainable, they have no causal interpretation. Covariates that appear at the first splitting points do not necessarily have a causal effect on the outcome, but may only be highly correlated with the outcome and other variables, which are themselves strongly associated with the target variable (James et al., 2013).

3.3.2 Random Forests

In order to increase the predictive power of decision trees, Breiman (2001) developed an algorithm called random forests (RF). This algorithm builds on a technique introduced earlier by Breiman (1996), called bagging, and applies it to decision trees. Assume there is a learning set $\mathcal{L}$, consisting of N observations with predictors $\boldsymbol{x_n}$ and outcome y_n and a predictor for y denoted $\varphi(\boldsymbol{x}, \mathcal{L})$. Further, suppose there is a sequence of learning sets $\{\mathcal{L}_k\}$, consisting of N independent observations drawn from the same distribution as $\mathcal{L}$. The goal of this approach is to use the sequence of sets $\{\mathcal{L}_k\}$ to construct a better predictor $\varphi(\boldsymbol{x}, \mathcal{L}_k)$ than the predictor from the single set $\varphi(\boldsymbol{x}, \mathcal{L})$. In a regression setting this can be achieved by replacing $\varphi(\boldsymbol{x}, \mathcal{L})$ with the expectation of the sequence of predictors learned from $\{\mathcal{L}_k\}$:

$$\varphi_A(\boldsymbol{x}) = \mathbf{E}_{\mathcal{L}}[\varphi(\boldsymbol{x}, \mathcal{L}_k)]. \tag{3.13}$$

Usually, there are no replicated learning sets from the same distribution as $\mathcal{L}$, thus Breiman (1996) proposes to imitate such replicates by repeatedly sampling N observations from the original learning set with replacement, known as bootstrapping.[10] Thus, the sequence of learning sets $\{\mathcal{L}_k\}$ is replaced by bootstrapped learning sets denoted $\{\mathcal{L}^{(B)}\}$, and the new predictor is the average of the bootstrapped predictors over all samples:

$$\varphi_B(\boldsymbol{x}) = av_B\ \varphi(\boldsymbol{x}, \mathcal{L}^{(B)}). \tag{3.14}$$

Since this new predictor is constructed using both bootstrapping and aggregating multiple predictors, this approach is called bootstrap aggregating, or bagging.

Breiman (1996) shows that improving the predictive power by the use of bagging critically depends on the stability of the procedure of constructing the predictors $\varphi(\boldsymbol{x}, \mathcal{L})$. If changes in $\mathcal{L}$ result in small changes of the predictor only, $\varphi_B(\boldsymbol{x})$ will be close to $\varphi(\boldsymbol{x}, \mathcal{L})$ and hence not generate large enhancements. Thus, the

[10] For further information regarding the bootstrap see for example Efron and Tibshirani (1994).

biggest improvements can be achieved for unstable procedures. In other words, algorithms with high variance would benefit the most from bagging. As stated in the previous section, decision trees, especially without pruning, are predictors with very high variance, but low bias, thus being a natural choice to be combined with bagging. Breiman (2001) therefore extends his idea of bagging to decision trees, resulting in a large number of single trees from bootstrapped samples forming a forest. The author shows that the generalization error and therefore the predictive power of the algorithm depends on two properties: low error trees and a low correlation between the individual trees. In order to achieve such low error trees, RFs utilize fully grown and unpruned trees, resulting in high variance, but low bias of the trees. All trees grown using bagging are identically distributed (i.d.). Therefore, the average over all trees is identical to the expectation of each single tree. Thus, the bias of each tree is the same as the bias of the bagged trees.[11] Since the expected MSE can be decomposed into an irreducible error, the squared bias and the variance (see also Section 3.1), bagging the decision trees increases the estimator's performance only if the variance decreases. The variance of the average of trees is given by

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2, \tag{3.15}$$

with B denoting the number of trees in the forest and ρ denoting their correlation. For an increasing B, the second term of Equation (3.15) approaches zero, and thus the increase in predictive power is limited by the correlation between the decision trees (Hastie et al., 2009). In addition to bagging, the RF algorithm therefore aims at further decorrelating the individual trees. Building on work and ideas from Amit and Geman (1997), Breiman (2001) proposes that the single trees should not consider all p predictors, but only use a random subset of $m \leq p$ predictors at each node. This is especially useful for decorrelating trees in datasets with one or few very strong predictors of the outcome. If every tree could access the whole predictor space, most of them would split the sample based on these variables early on, resulting in very similar trees (James et al., 2013). In regression settings, it is often recommended to set $m = p/3$ and grow trees until a minimum node size of 5 observations is reached. However, these parameters may also be treated as hyperparameters, and thus tuned using CV (Hastie et al., 2009). The higher m, the more predictors are used and the lower becomes the error of each single tree, but the higher gets the correlation between these. Thus, the choice of m again implies a trade-off. Since the trees should be uncorrelated, a very

[11] As a reminder: bias refers to the amount by which the average of an estimate deviates from the true mean. Since all trees are i.d., the bias is the same for each one of the trees. Thus, the average of all biases is just the bias of each single one of these trees.

large number of trees usually does not lead to overfitting. Therefore, the number of trees in a forest can either be set to be sufficiently large, or also be tuned (Breiman, 2001).

When making predictions with RFs, the algorithm simply outputs the average value of all predictions made by the single trees in the respective forest. Let Θ_b denote the characteristics of the b-th decision tree $T(\boldsymbol{x}, \Theta_b)$ in the RF with regard to the splitting variables and thresholds, the leaves, and the values in the leaves. In a RF with B trees the predictions are then given by:

$$\hat{f}_{RF}^{B}(\boldsymbol{x}) = \frac{1}{B} \sum_{b=1}^{B} T(\boldsymbol{x}, \Theta_b). \tag{3.16}$$

3.3.3 Gradient Boosting and XGBoost

Another technique to increase the predictive properties of a single regression tree was introduced by Friedman (2001). The author first proposed boosting as a general-purpose technique, which can be used in combination with any learner and any differentiable loss-function $L(y, F(x))$. The idea behind boosting is to approximate an optimal function

$$F^*(x) = \arg\min_{F(x)} \mathbf{E}[L(y, F(x))] \tag{3.17}$$

through an additive regression model constructed by sequentially fitting simple functions, called weak or base learners, to pseudo-residuals from a previous weak learner in M steps:

$$F(x) = \sum_{m=0}^{M} \beta_m h(x, a_m). \tag{3.18}$$

$h(x, a_m)$ denotes the base learners, fitted to predictors x with parameters a_m, and β_m is called the expansion coefficient. Friedman (2001) also introduces the special case of regression trees as base learners and a quadratic loss-function $L(y, F(X)) = (y - F(x))^2/2$. Here, β_m denotes the output values in the leaves, and $h(x, a_m)$ becomes $h(x, \Theta_m)$, with Θ_m being the characteristics of a base learner, i.e. the splitting rules as well as terminal nodes and values in the leaves.

Approximating $F^*(x)$ is described as an iterative process. Let $\{(x_i, y_i)\}_{i=1}^N$ denote data on predictors and outcomes from N observations. At first, the model is initialized with a constant γ which solves:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma). \tag{3.19}$$

In this case, γ is just the average of all observed outcomes. For each step $m = 1, \ldots, M$ at first the negative derivative of the loss-function is calculated with respect to the model at the current step:

$$\tilde{y}_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i))}\right]_{F(x)=F_{m-1}(x)}. \tag{3.20}$$

Using the above stated quadratic loss-function, $\tilde{y}_{im}$ corresponds to the difference between the observed and the predicted values from the model in the previous iteration, also called pseudo-residuals. In a next step, a new regression tree is fitted to these pseudo-residuals and R_{lm} terminal regions are defined. The output values γ_{lm} of each of the L leaves of this new regression tree are then determined by minimizing the loss in each leaf:

$$\gamma_{lm} = \arg\min_{\gamma} \sum_{x_i \in R_{lm}} L(y_i, F_{m-1}(x_i) + \gamma), \tag{3.21}$$

with F_{m-1} being the predicted value in step $m-1$. With the squared loss-function, the γ-values minimizing this objective are again simply the averages of values in the leaves as in standard regression trees.

This process of iteratively fitting new regression trees to the residuals from previous trees can easily lead to overfitting, and thus it requires some form of regularization as well. Friedman (2001) discusses two potential factors for regularizing boosted trees. Firstly, the risk of overfitting increases with the number of boosting iterations, i.e. in M. In addition to that, the author recommends using what he calls shrinkage. Instead of adding the γ_{lm}-values from equation (3.21) to the prediction from the previous step, each update of $F(x)$ is scaled by a scalar $0 < \nu \leq 1$, i.e. at each iteration $F_m(x)$ is updated in such a manner that

$$F_m(x) = F_{m-1}(x) + \nu\, \gamma_{lm}\, \mathbf{1}(x \in R_{lm}). \tag{3.22}$$

ν thus determines the rate at which the model's prediction is updated. In other words, ν determines how fast the model "learns" the specific characteristics of the

data and is therefore also called learning rate. The number of iterations M and the learning rate ν are hyperparameters and can be chosen by CV. Low values of ν usually require a higher number of boosting iterations to yield a sufficient fit to the data. Moreover, Friedman (2001) shows in simulations that low learning rates of $\nu \leq 0.1$ result in a better predictive performance of the estimator. However, since low ν requires high M, this approach increases the computational cost of the algorithm.

Expression (3.20) is also known as the gradient, therefore, this algorithm is also called gradient boosting. In a subsequent paper, Friedman (2002) further extends the idea of gradient boosted regression trees. Motivated by the work from Breiman (1996) on bagging and Breiman (2001) on RFs, the author adds an approach, which is similar to bagging, to gradient boosted trees. However, instead of considering a random subset of predictors at each node, only a random subset of $\tilde{N} < N$ observations is considered in every iteration. Thus, Friedman (2002) incorporates row subsampling instead of column subsampling into his algorithm. Due to the randomness induced by row subsampling, this algorithm is called stochastic gradient boosting. The smaller the fraction $f = \tilde{N}/N$ of considered rows for each tree, the higher the variance of the base learners. However, low f reduces computation time of the algorithm, since the trees are fitted to smaller subsamples. Friedman (2002) finds that stochastic gradient boosting improves the predictive power substantially compared to standard gradient boosting and recommends values of $0.5 \leq f \leq 0.8$ for regression settings.

A recently developed algorithm building on the idea of boosted regression trees called XGBoost (Chen and Guestrin, 2016) received a lot of attention in the last few years.[12] Similar to standard boosting, XGBoost builds additive models on the (pseudo-)residuals of previous estimators, however it utilizes non-standard regression trees. At first, the model is initialized with a constant prediction for every observation, and then regression trees are additively and iteratively trained on the residuals. When growing a new tree in iteration t, the algorithm uses a regularized loss-function:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{N} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t), \tag{3.23}$$

$$\text{with } \Omega(f_t) = \gamma T + \frac{1}{2}\lambda w^2. \tag{3.24}$$

[12] For example, many data science and prediction-related online competitions could be won using the XGBoost algorithm. The authors state that out of 29 competition winning solutions on the data science competitions website kaggle, 17 used XGBoost alone or in combination with other algorithms (Chen and Guestrin, 2016).

w denotes the output values from the leaves of the new tree, T the number of leaves in a tree. γ and λ are hyperparameters and regularize the complexity of the tree. The above stated equation is minimized using second-order Taylor approximation[13], which then leads to the following objective:

$$\mathcal{L}^{(t)} = \sum_{j=1}^{T} \left[\Big(\sum_{i \in I_j} g_i \Big) w_j + \frac{1}{2} \Big(\sum_{i \in I_j} h_i + \lambda \Big) w_j^2 \right] + \gamma T, \tag{3.25}$$

where g_i denotes the first derivative of the loss-function at x_i with respect to the output values, h_i denotes the second derivative, and w_j the output value from the j-th leaf. The values for w_j minimizing equation (3.25) are then given by:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}. \tag{3.26}$$

In regression settings squared-loss $\mathcal{L}(y_i, f(x)) = (y_i - f(x))^2/2$ is used, such that g_i are just the negative residuals, and $h_i = 1$ for all observations. Therefore, the optimal output values are given by the sum of squared residuals divided by the number of observations in each leaf plus the regularization parameter λ. If $\lambda = 0$, this is just the average of residuals in the leaf, for $\lambda > 0$, the optimal output values are shrunken towards zero.

Using the expression for w_j^* and plugging it into equation (3.25) yields a scoring function for each tree with characteristics q:

$$\mathcal{L}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \tag{3.27}$$

In addition to λ, γ is another regularization parameter and penalizes larger trees, hence it also prevents the trees from overfitting the data. Moreover, XGBoost also utilizes column or feature subsampling as used in RFs, as well as shrinkage by the use of a learning rate η as in stochastic gradient boosting developed by Friedman (2002).

As depicted before in the case of decision trees, it is computationally not feasible to examine every possible tree structure, therefore greedy binary splitting is used here as well. However, in contrast to other algorithms, XGBoost uses what the authors call an approximate greedy algorithm. Instead of looking for the optimal splitting value among all observed predictor values, the algorithm only

[13] For more details in the context of boosting see for example Friedman et al. (2000).

uses quantiles as possible splitting point candidates.[14] This approximate greedy split finding leads to a significantly reduced computation time, especially for large datasets. In combination with some other optimizations regarding computation, as for example utilizing and saving intermediate results in the CPU cache, this enables a very efficient and fast boosting algorithm. Due to these reasons and its success in recent years among many practitioners, XGBoost is used as an example of a boosting algorithm in this thesis.

3.4 Neural Networks

The last class of algorithms introduced in this section and applied in the empirical analysis of this thesis, are neural networks. While there is a high number of different architectures and methods to construct neural networks, this section focuses on feedforward neural networks. Such networks consist of at least three layers: an input layer, at least one hidden layer and an output layer. Each of these layers is composed of a number of elements called nodes or neurons and each node is connected to every node in the subsequent and the previous layer. Neural networks are able to perform classification as well as regression tasks. The number of output nodes is chosen according to the specific prediction problem at hand. Since the goal of regression is to predict a numerical value for some input data, in such settings the output layer consists of a single node (Hastie et al., 2009).

In a first step, the inputs from the training data are forwarded to the nodes in the hidden layers as linear combinations. Each input enters every node with a possibly different weight. In the neurons a weighted sum of the predictors is calculated and used as input to a so-called activation function. Moreover, in each node an additional constant is added to the weighted sum before entering the function, called bias. The value given by the activation function, also called a neuron's activation, is then again used as (weighted) input to the neurons in the next layer. The weights and biases are the model parameters which are learned from the training data (Hastie et al., 2009). To facilitate the understanding of how neural networks are designed, Figure 2 illustrates the architecture of a simple feedforward neural network consisting of an input layer, two hidden layers and an

[14] The quantiles evaluated as splitting thresholds do not correspond to regular quantiles, but are based on the sum of weights given to observations instead of the number of the observations in each bin. Those weights are found by another special algorithm developed by the authors called weighted quantile sketch. They are equal to h_i and thus equal 1 for all $n = 1, \ldots, N$ in regression settings. However, in classification tasks, this algorithm assigns higher weights to low confidence predictions, making such observations less likely to end up in the same leaf and receive the same output value (Chen and Guestrin, 2016).

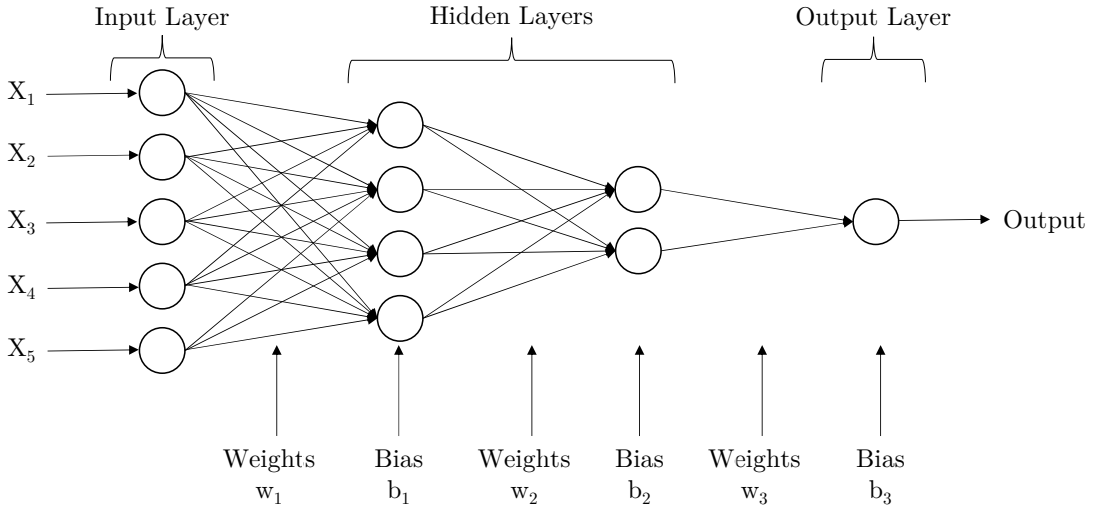


Figure 2: Feedforward neural network architecture with 2 hidden layers, taken and adapted from Nielsen (2015)

output layer with a single output neuron. There are numerous possible activation functions which can be used in the neurons of neural networks. The most common choice is the so-called sigmoid function which is just the logistic function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{3.28}$$

σ denotes the activation function, while z denotes the weighted sum of J inputs plus the bias b:

$$z = \sum_{j=1}^{J} (w_j x_j + b). \tag{3.29}$$

For regression tasks, the activation function in the output neuron is usually chosen to be the identity function (Hastie et al., 2009).

When learning the model parameters, i.e. the weights and biases for each link and neuron at each step, neural networks usually rely on two techniques: (stochastic) gradient descent and backpropagation. Gradient descent is a general-purpose technique to optimize differentiable functions in an iterative way. Let $C(v)$ denote a differentiable function, which is to be minimized. For simplicity, assume that $v = v_1, v_2$, however the following also holds for the general case. The change

in this function, induced by changes in its parameters, is given by the vector of partial derivatives of $C(v)$, also called the gradient vector and denoted ∇C:

$$\nabla C = \left(\frac{\partial C}{\partial v_1}, \frac{\partial C}{\partial v_2} \right)^T. \tag{3.30}$$

The actual change in C can then be expressed as:

$$\Delta C = \frac{\partial C}{\partial v_1} \Delta v_1 + \frac{\partial C}{\partial v_2} \Delta v_2 \tag{3.31}$$

$$\Delta C = \nabla C * \Delta v. \tag{3.32}$$

Now suppose that Δv is chosen such that:

$$\Delta v = -\eta \nabla C, \tag{3.33}$$

with η being a small, positive number. Then Equation (3.32) becomes

$$\Delta C = -\eta \left\| \nabla C \right\|^2. \tag{3.34}$$

Since $\left\| \nabla C \right\|^2 \geq 0$, this ensures that the change in $C(v)$ is always negative for changes in v_1, v_2 as stated in equation (3.33). Building on these results, in order to minimize the function $C(v)$, its parameters are iteratively updated according to the following equation:

$$v \rightarrow v' = v - \eta \nabla C. \tag{3.35}$$

The parameter η determines the amount of how much the parameters are updated, and thus controls the speed of the function being minimized as well as the number of steps needed to reach a minimum. Due to these properties, η is also called learning rate (Nielsen, 2015). This learning rate is usually chosen using CV.

In neural networks, the cost-function to be minimized most of the time corresponds to quadratic loss when used for regression tasks:

$$C(w, b) = \frac{1}{2n} \sum_x (y - f(x))^2, \tag{3.36}$$

with n being the number of observations and $f(x)$ the output from the neural net for given input data x. Since the objective function is a sum over all train-

ing observations, computing the gradient corresponds to computing the partial derivative for each observation, i.e.

$$\nabla C = \frac{1}{n} \sum_{x} \nabla C_x. \tag{3.37}$$

Calculating ∇C_x is computationally very expensive. In order to decrease the amount of computation, an approach called stochastic gradient descent (SGD) is used. Stochastic gradient descent works very similar to gradient descent, however, instead of using the whole training sample to compute the gradient, a randomly drawn subsample of observations is used to estimate the gradient. The parameters are updated according to the gradient obtained from these subsamples, also called batch or mini-batch (Nielsen, 2015). The idea behind SGD is that each mini-batch yields a noisy, but unbiased estimate of the gradient (Athey and Imbens, 2019). So, using SGD with m observations in each mini-batch, updating the parameters from equation (3.35) is done in the following way:

$$v \to v' = v - \frac{\eta}{m} \sum_{x \in m} \nabla C_x. \tag{3.38}$$

In order to compute the gradient of the loss-function, neural networks as described here usually use an algorithm called backpropagation (Hastie et al., 2009). Since each neuron receives activations from neurons in the previous layer, the actual output from the neural network depends on all the activations from all of the previous neurons. Let a^l denote the activations from neurons in layer l, σ the activation function, w_l the weights from layer $l-1$ to l and b^l the added bias in layer l. The activation of layer l, a^l, can then be written as

$$a^l = \sigma(w^l a^{l-1} + b^l). \tag{3.39}$$

Therefore, in order to compute the derivatives of the loss-function with respect to all weights and biases, at first the input is forwarded through the neural network to get the prediction for the specific observation, and the error is calculated. Then, this error is propagated backwards through the network, and at each step the partial derivatives with respect to the weights and biases are calculated using the chain rule. This is done until the first layer is reached, and the parameters are updated using the SGD algorithm described above (Hastie et al., 2009; Nielsen, 2015).[15]

[15] For mathematical details regarding the backpropagation algorithm and theoretical derivations see for example Nielsen (2015) or Hastie et al. (2009).

As other ML algorithms, due to their high flexibility, neural networks are prone to overfitting the training data, which leads to poor generalization. One way to penalize the flexibility and complexity of the fitted function is to use a form of regularization, which is similar to the approach utilized for the LASSO regression. Instead of minimizing the error between actual and predicted values, a regularization parameter is added to the loss function. Let C again denote the cost function, the new objective function then becomes:

$$C(w, b) = \frac{1}{2n} \sum_{x} (y - f(x))^2 + \frac{\lambda}{2n} |w_j|^2. \tag{3.40}$$

In the case of neural networks, this type of regularization is called weight decay. Here, λ is a hyperparameter used to define the intensity of the regularization (Nielsen, 2015; Venables and Ripley, 2013).

In addition to the SGD and backpropagation algorithm applied in order to optimize the loss function, there exist several other optimization strategies, which can be and are used to train neural networks. For example, building on SGD, an algorithm called resilient propagation (RProp) sets the learning rate, and thus the step size in adjusting the weights, adaptively (see for example Riedmiller and Braun, 1993; Anastasiadis et al., 2005). Other algorithms are modifications of Newton's method and use the second order derivatives or approximations of the second order derivatives to numerically find solutions for the optimization problem. The most common of such methods is the so-called BFGS method, named after the researchers who proposed it first (Fletcher, 1970; Broyden, 1970; Goldfarb, 1970; Shanno, 1970).[16]

4 Data Description and Preprocessing

In order to provide answers to the research questions stated in the introduction, data on house prices, on wind turbine locations in Germany and on factors potentially influencing both house prices and the construction of wind turbines in a specific area, to satisfy the unconfoundedness assumption from Section 2.1, is needed. Therefore, the empirical analysis conducted in this thesis draws on data from three different sources: The first dataset, RWI-GEO-RED, contains data on houses for sale in Germany, advertised on ImmoScout24, Germany's largest internet platform for real estate advertisements (Boelmann et al., 2019). This

[16] Due to the limited scope of this thesis and because the exact optimization algorithm is not needed to get a basic understanding of how neural networks work, these method are not explained in more detail here.

dataset is complemented by another data source, RWI-GEO-GRID, containing information on sociodemographic data on a square kilometer grid-level, collected by the commercial data provider microm GmbH (RWI and microm, 2019). The assignment of houses to the same square kilometer grid allows to combine both sets in order to obtain an extensive data base on factors influencing both house prices as well as the construction of wind turbines. These two data sources are provided by the Research Data Centre Ruhr at the RWI – Leibniz-Institute for Economic Research (FDZ Ruhr). Lastly, the third dataset contains information on wind turbines in Germany. It comprises information on the date of commissioning of each power plant as well as the exact location defined by geographic coordinates. The following section describes the three data sources in more detail and explains necessary data cleaning and preprocessing steps.

4.1 Real Estate Data and Sociodemographics

Starting with the real estate data, the RWI-GEO-RED dataset comprises information on the asking price as well as on all additional characteristics of the properties, which are advertised for sale on the real estate platform ImmoScout24. The publication dates of the advertisements range from 2007 up to March 2019. However, since the data on wind turbines is as of December 31, 2018, the real estate data is restricted to this date as well. The characteristics of the houses include for example the size of the living and plot area, the number of rooms, the year of construction or the category of the property. In addition to that, technical variables as for example the number of views on the website for each house or a unique ID are available as well. The exact location of each house is anonymized and instead mapped to multiple geographic areas, with the smallest area being a one square kilometer grid according to the European standard ETRS89-LAEA, but also to higher levels such as postcode area, municipalities, districts and labor market regions (Boelmann and Schaffner, 2019).[17]

Although the data is already cleaned to some extent by the FDZ Ruhr, there are still some faulty and implausible values. For example, some houses have a living or plot area of zero square meters (sqm) or a number of rooms or floors being zero. Moreover, some entries are unreasonably small, e.g. values of 0.01 for certain variables, as for example the number of rooms. Those faulty data points are replaced by missing values. Furthermore, asking prices below 20,000 €, living and plot areas below 40 sqm, and number of rooms below one are replaced by missing values as well, following Frondel et al. (2019). In addition to that,

[17] The full, unanonymized dataset containing the exact coordinates of the included houses can be accessed from a special data security room at the FDZ Ruhr via on-site access.

there are houses with the same ID occurring multiple times in the data due to reporting issues by ImmoScout24, or actual multiple occurrences on the platform. To account for these multiple appearances, only the latest observation of those is kept in the data, since an advertisement appearing later in time reflects an updated price, which is most likely to be closer to the properties' actual value or market price.

In addition to the information contained in RWI-GEO-RED, the dataset RWI-GEO-GRID comprises socioeconomic data, which is used to complement the real estate data. Since the dataset is on the same one square kilometer grid as the real estate data, both data sources can easily be merged in order to obtain a single extensive dataset. RWI-GEO-GRID contains variables from four categories: household, mobility, building development and population. Additionally, each category comprises data on the number of households, enterprises, and buildings (Breidenbach and Eilers, 2018). The variables used for the analysis account for the relation of residential and commercial buildings, how densely a grid is populated, mobility related information, as for example the density of cars, and the demographic composition of the inhabitants living in each grid.[18] While RWI-GEO-RED contains data from 2007 up to March 2019, the grid data is available for 2005 and 2009 to 2017 only. In order to use these additional sociodemographics for all observations in the real estate data, information for the missing years is imputed. The data points for 2006, 2007 and 2008 are imputed by linearly interpolating the years 2005 and 2009, while information for 2018 is added using linear extrapolation. To do so, a linear regression model is fitted to the years 2015 to 2017 and then used to predict the values for 2018. The data on which the model is trained is restricted to the three most recent years in order to capture a possible (linear) time trend. Since earlier years are likely not to be too relevant for the extrapolated year, these are not included in the model to prevent it from picking up too much noise. The inter- and extrapolations are done for each grid separately.

Since one of the aims of this thesis is to analyze heterogeneity of treatment effects based on covariates and also to identify the most and least affected group by their covariates, missing values in the sample are not imputed. Instead, the sample used for the estimation should consist of those observations only for which there are information on all variables. However, the number of observations, for which data on all covariates is included in the combined dataset, is very small. Therefore, in

[18] Table A1 in Appendix A.3 lists all variables in the dataset and indicates which of these are used throughout this master thesis.

a first step, variables with a very high share of missing observations are excluded. A threshold of 60 % missing values is chosen to remove variables entirely. Table 1 displays the variables removed for this reason and their corresponding shares of missings.

Table 1: Share of missing observations

Variable	Share of Missings (in %)
Number of ancillary rooms	99.9
Heating costs	99.9
Elevator (0/1)	99.9
Rental income	97.3
Price of parking space	96.1
Energy efficiency rating	92.8
Last modernization	88.2
Energy consumption per year & sqm	84.2
Type of energy performance certificates	84.1
Parking space available (0/1)	83.0
Wheelchair-accessible (0/1)	81.3
Construction phase	76.0
Warm water included in energy consumption (0/1)	70.5
Quality of furniture	66.3
Usable floor space	63.3

Notes. (0/1) after the variables denote dummy variables.

In addition to that, the following variables are excluded as well, since they are not likely to have high explanatory power beyond the variables already included, and additionally suffer from a relatively high share of missing values: number of bedrooms (49.4 %), rented at sale (48.9 %), granny flat (48.6 %), cottage (31.3 %) and guest bathroom (26.8 %). Moreover, the variable indicating a property being advertised in 2007 is removed due to very little variation[19] and houses categorized as being castles are included in the category "other" for the same reason.[20] In a second step, for houses which appear multiple times in the data, some of the missing values in their latest appearance can be filled by forward filling information for the missing data points from previous advertisements of the same house. In a last step eventually, only those observations which contain data on all the

[19] Since the year dummies are mutually exclusive, the removed variable can be reconstructed from the other year dummies. Therefore, this is unproblematic.

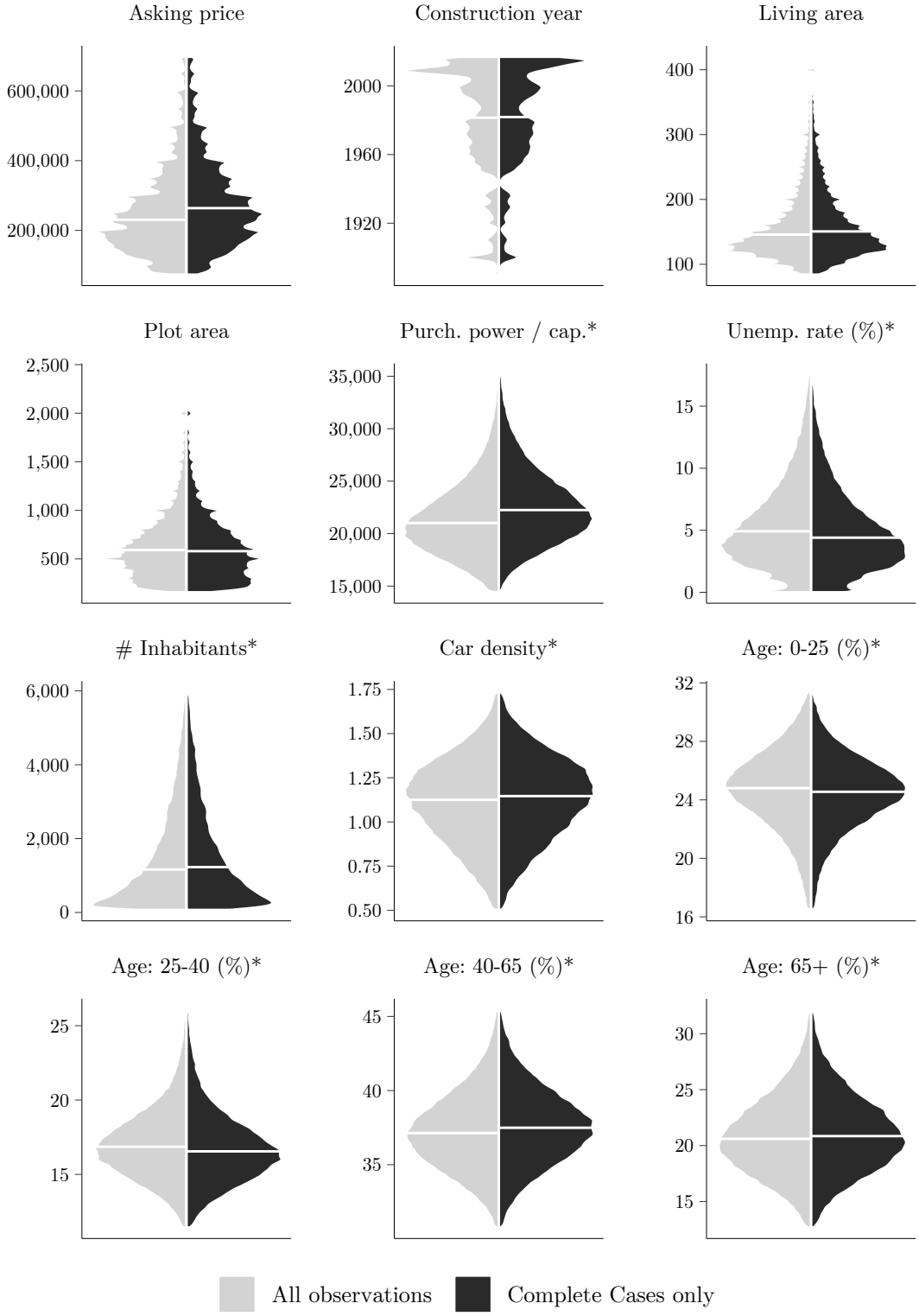
[20] As for the RWI-GEO-GRID dataset, Table A2 provides a list of all variables included in RWI-GEO-RED and indicates which of these are used in this thesis.

remaining variables in the dataset, i.e. only complete cases, are kept in the final sample, while all other observations are excluded.

A possible concern regarding this approach of excluding a considerable amount of data as done here, is that there is a risk of unintentionally removing information in a systematic way such that the remaining sample differs substantially from the original sample. This may lead to a selected sample which does not represent the studied population anymore, and thus to selection bias (see for example Imbens and Wooldridge, 2009). To make sure that this is not a problem in the present case, Table A3 in Appendix A.3 compares selected measures of the distribution of both the restricted sample and the original unrestricted sample. Looking at this table, it can be seen that almost all of the variables are very similarly distributed, indicating that there is not a systematic removal of information in the data. Additionally, Figure 3 provides a visual comparison of the distribution of selected continuous variables of both the restricted and the unrestricted sample using (modified) so-called beanplots (Kampstra, 2008). These plot the estimated density of a variable and its median represented by a horizontal line in one group on the left, and the estimated density and the median of the same variable in a second group on the right. This provides a simple and intuitive way of visually comparing distributions between groups and assessing how much both groups differ in terms of the presented variables. The densities for the variables look very similar and also the medians do not differ by large amounts in both groups. The beanplots in Figure 3 thus provide additional evidence that restricting the sample as described above does not lead to a selected dataset differing severely from the full sample.

4.2 Wind Turbines and Distances

In order to identify the nearest wind turbines and the corresponding distances to the properties at the time they are being advertised on ImmoScout24, information on wind power plants in Germany, namely their exact location and the date of commissioning, are needed as well. In January 2019, the so-called Core Energy Market Data Register (MaStR) was initiated by the Federal Network Agency. The MaStR constitutes a binding institutional register for all actors in the energy and gas sector in Germany. They are obliged to register and to provide information on all energy producing plants, including i.a. the location, date of commissioning and nominal production capacities. However, up until now the register is mandatory for plants which were taken into operation after July 2019 only. All other plants which started production earlier do not need to be registered until January 31, 2021 (Bundesnetzagentur, 2020b). Although these are already included in the



Notes. For improved visual representation, the lower and upper ends of the data are excluded from the plot. The following thresholds are used: asking price - 5 % and 95 % percentiles; construction year, living area, plot area, number of inhabitants - 3 % and 97 % percentiles; purchasing power / capita, unemployment rate (%), car density, age: 0-25 (%), age: 25-40 (%), age: 40-65 (%), age: 65+ (%) - 1 % and 99 % percentiles. Asterisks after variable names denote grid-level variables. The horizontal lines indicate the median in the respective group.

Figure 3: Beanplots

register due to migration from older registers and data sources, information on their location is still missing for most plants. Thus, although data on some wind turbines is already included, the MaStR cannot serve as a sufficient data source for the wind turbines in Germany. Therefore, to overcome this issue, the data on wind power plants was collected for each federal state separately. For most states, this information was freely accessible and downloadable, for Hamburg, Saxony, Lower Saxony, NRW and Rhineland-Palatine it was provided by the responsible regional authorities. Unfortunately, no data for Berlin was available, therefore houses from this state are excluded during the preprocessing procedure as well.[21]

Since there is no comprehensive, official data on wind turbines in Germany, different data sources also report differing total numbers of onshore wind turbines. For example, the aforementioned MaStR lists 28,154 wind turbines as of December 31, 2018 (Bundesnetzagentur, 2020a,b), while according to joint information from The German Wind Energy Association and the private company Deutsche WindGuard (2019) 27,765 onshore wind turbines were in operation in Germany at the end of 2018. A comparison of the number of all wind turbines combining the data for each state separately, as well as the installed capacity with those two data sources on the federal state-level can be found in Table A4 in Appendix A.3. The table reveals that the number of wind turbines for each state is either slightly lower compared to data from MaStR and Deutsche Windguard or lies in between both figures. There is no federal state for which the numbers deviate by a large amount. The same holds true when comparing the total installed capacity. Considering both these criteria, the employed dataset can be regarded as representative and (almost) complete for Germany, excluding Berlin.

In order to find the nearest wind turbines and calculate the distances between the advertised houses and them, the geographic coordinates are translated into a mutual coordination system, as the single datasets provided by the regional authorities use different coordination systems and projections. Although the exact location of the advertised properties is anonymized in the scientific use file provided for download, it can be accessed via a special data security room via on-site access at the FDZ Ruhr in Essen. Using this information and the data on the location of wind turbines, in a first step the 500 nearest wind turbines are identified for each house and the corresponding distances are calculated. In a second step, the date of the advertisement and the commission date of the wind

[21] An overview of the exact data sources and the regional authorities who provided the data is found in Appendix A.3 in Table A5.

turbine are compared. If the date of advertisement is later than the commission date of the respective power plant, the distance to this plant is saved as the actual distance to the nearest wind turbine. If the advertisement date is earlier than the commission date, the wind turbine was not yet in operation when the house was advertised on ImmoScout24, and thus it does not represent the nearest turbine at the time of advertising. In such a case, the second-nearest wind turbine is investigated in the same way, until a valid wind power plant is identified, and the corresponding distance is saved. However, for some of the wind turbines the exact date of commissioning is not available, but only the year, so it is not possible to tell whether they were already in operation when a property is advertised in the same year or whether they were constructed afterwards. Furthermore, there is a small number of wind turbines for which neither an exact date of commission nor the year is known. If the identified nearest wind turbine belongs to those two cases, it is explicitly tagged as such. In these cases, further wind turbines are examined in the same way until one is found, which can be clearly identified as being in operation when the house was put on ImmoScout24 and the corresponding distance is additionally saved.

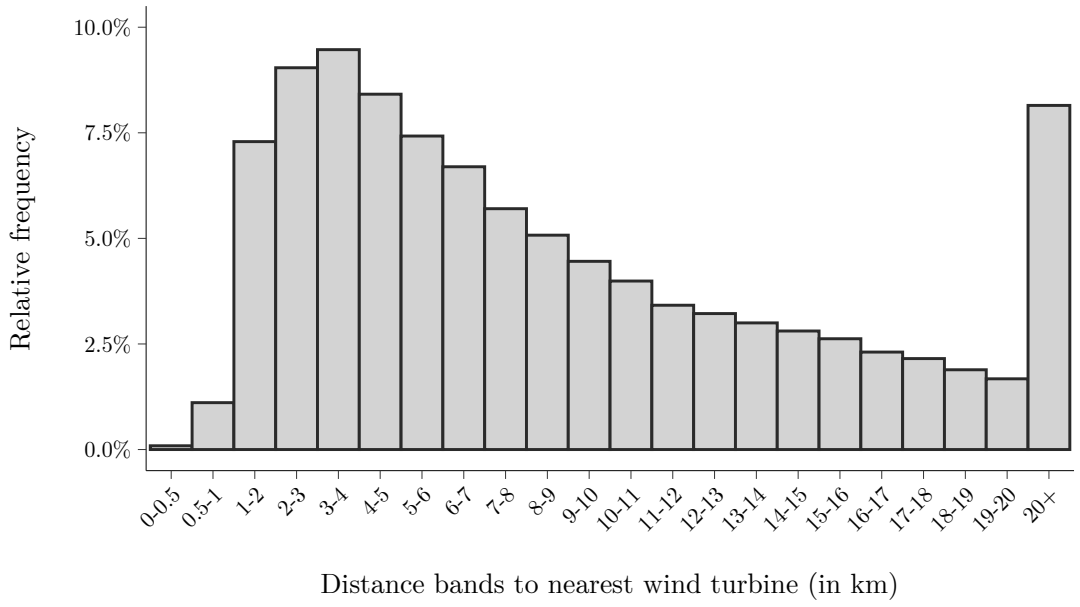


Figure 4: Relative frequencies of distance band categories

After the identification of the nearest wind turbines, the exact distances are discretized into distance bands in order to assure that the exact location of the properties cannot be reconstructed from the distances. The following bands are used: 0–0.5 km, 0.5–1 km, 1–2 km, 2–3 km, 3–4 km, 4–5 km, 5–6 km, 6–7 km, 7–8 km, 8–9 km, 9–10 km, 10–11 km, 11–12 km, 12–13 km, 13–14 km, 14–15 km, 15-16 km, 16–17 km, 17–18 km, 18–19 km, 19–20 km, 20+ km. Figure 4 plots the

relative frequency of each group against the distance bands. It can be seen that only a very small share of houses lies in the range of up to 1 km to the nearest wind turbine. A distance of between 3 and 4 km constitutes the largest group in the data. In addition to Figure 4, Figure 5 also plots the cumulative relative frequency for the distance bands as a step function. This plot reveals that more than 25 % of the properties lie in the range of maximum 4 km to the nearest wind turbine, slightly more than 50 % in the range of maximum 7 km and more than 75 % in the range of maximum 13 km. Moreover, Figure 5 also compares the cumulative relative frequencies of the restricted sample, which contains complete cases only, with the cumulative relative frequencies of the entire sample without excluding the incomplete entries. The two lines are almost indistinguishably similar to each other, providing further evidence that the conducted data cleaning steps did not result in a particularly selected sample.

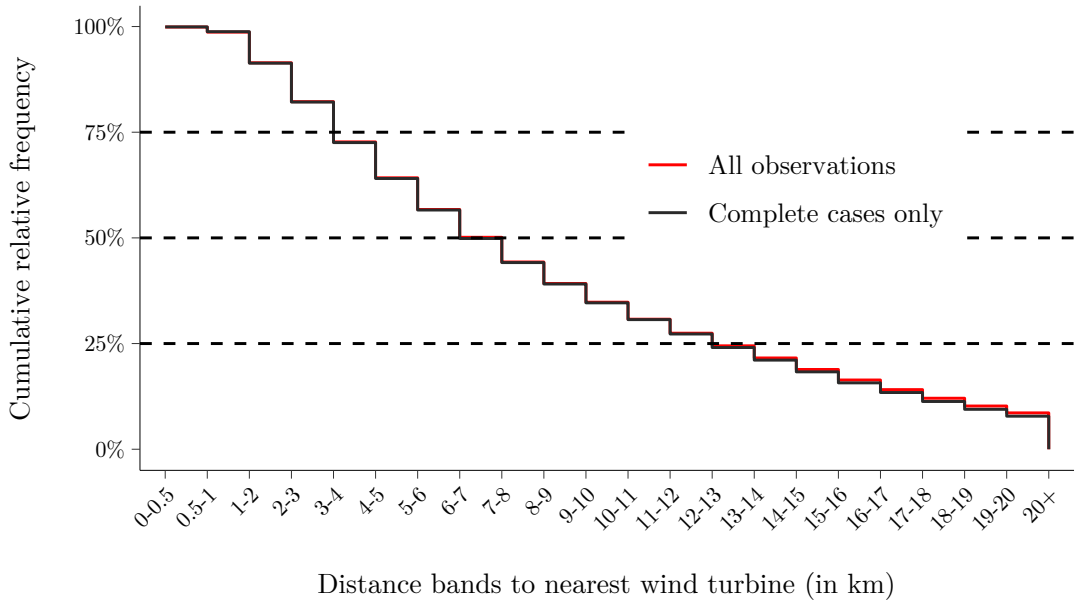


Figure 5: Share of observations within given distance bands

Figure 6 summarizes the entire data cleaning process in the form of a flowchart. The original dataset, as obtained from the FDZ Ruhr, which still includes multiple entries for the same houses, contains 12,419,820 observations. After restricting the data to the years 2007 to 2018, excluding data for Berlin due to missing data on wind turbines, and removing the duplicate IDs, the sample consists of 8,691,312 houses. Merging the calculated distance bands to the houses results in a loss of around one million houses due to missing information on their exact location. Excluding all those observations, which have missings in any of the remaining variables, and keeping the complete cases only leaves a sample of 943,927 properties. Since some of the grids in RWI-GEO-GRID contain only a

very small number of houses or households, some of the variables, as for example the purchasing power, are anonymized to prevent backtracing of these variables to single households. This results in a loss of additional 7,079 observations after merging the real estate data with this sociodemographic data. Eventually, the final sample after the data cleaning process consists of 936,848 observations.

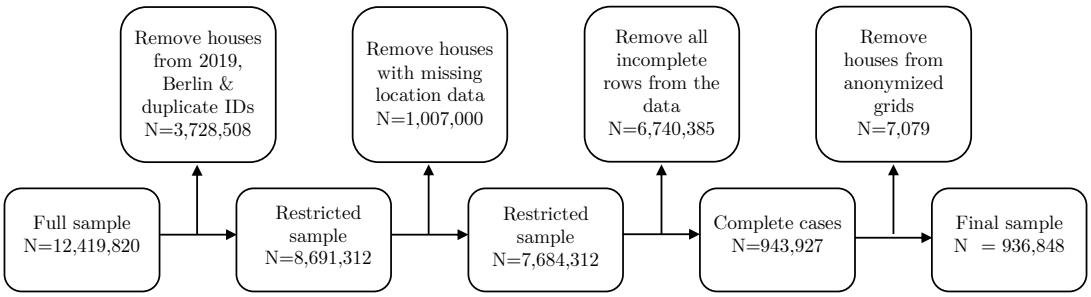


Figure 6: Data cleaning process

5 Implementation Details

Applying the method discussed in Section 2 in the context of the empirical analysis in this thesis comes with a few caveats and issues which have to be taken care of. As a reminder, Figure 7 summarizes the estimation procedure as proposed by Chernozhukov et al. (2018) in the form of pseudocode.

The first thing to notice is that the method requires a binary treatment indicator, and hence a clear identification of observations being treated and not being treated. Since the data used in this thesis contains the distance to the nearest wind turbine in form of distance bands, an appropriate distance threshold has to be chosen to divide houses into a treatment and a control group. As discussed in Section 1, Sunak and Madlener (2016) estimate the impacts on wind turbines on property prices in three cities in North Rhine-Westphalia. In a preliminary analysis, the authors construct indicators of visual impact levels and find that for houses, exposed to a visibility classified as "marginal", two wind turbines can be seen on average, and that these are located around 4,500 m from the nearest turbine. In the analysis by Frondel et al. (2019) the authors find significant negative effects on properties for house – power plant distances of up to 8 km. However, other studies find effects up to 2 km (Dröes and Koster, 2016) or up to 4 km only (Gibbons, 2015). Therefore, as a compromise between the cited studies and referring to Sunak and Madlener (2016), a distance of maximum 5 km is chosen as a treatment threshold. Thus, using the calculated distances and the derived distance bands as described in Section 4.2, houses in a range of maximum 5 km to the nearest wind turbine are considered being treated, while houses for which the

Algorithm 1:

Input: Data on units $i = 1, \dots N$ with covariates Z_i and a binary treatment indicator D_i

Result: BLP, GATES, CLAN

```
begin
    Fix the number of splits S and significance level α, e.g. S = 100 and
    α = 0.05
    Compute the propensity scores p(Z_i)
    Split the data S times into equally sized subsamples Data_A, Data_M
    for s = 1 ... S do
        for Algo in ML Algorithms do
            Tune and train Algo in Data_A to learn B(Z_i) and S(Z_i)
            Predict B(Z_i) and S(Z_i) in Data_M
            Construct k groups based on proxy S(Z_i)
            Estimate BLP parameters in Data_M
            Estimate GATES parameters in Data_M
            Estimate CLAN parameters in Data_M
            Compute performance measures
        end
    end
    Compute medians of parameters of interest, confidence intervals,
    p-values using VEIN methods and medians of performance measures
end
```

Figure 7: Algorithm as proposed by Chernozhukov et al. (2018)

nearest wind turbine is farther away are classified as control observations. For properties for which it could not be clearly stated whether the identified nearest wind turbine was already in operation when the advertisement was placed, the first wind turbine which is clearly identified as already in operation is taken into consideration. If the distance to the latter is smaller than 5 km the property belongs to the treatment group, regardless of a possibly even closer wind turbine. If this distance however is larger than the chosen threshold of 5 km, the treatment indicator is tagged as missing and the observation hence gets excluded from the data, since it is not possible to tell whether there was a wind turbine in the maximum distance range at the time of the advertisement.

During the actual estimation, the machine learning algorithms are used to construct the BCA $B(Z)$ and the proxy treatment effect $S(Z)$ for each split of the data. In their paper, Chernozhukov et al. (2018) suggest tuning each algorithm's hyperparameters separately for every data split. Since hyperparameter tuning is

done using a repeated procedure of fitting a model with fixed hyperparameters on some resampled data and evaluating the model's performance on a separate hold-out test set, hyperparameter tuning can be very computational expensive and time consuming. The most frequently used strategy for model tuning is via k-fold CV, as already mentioned in Section 3.1.[22] For a choice of $k = 5$, a model is trained 5 times to 80 % of the data and tested 5 times on 20 % for each set of hyperparameters. Thus, evaluating a single hyperparameter set requires each algorithm to be fitted 5 times which might lead to long computation times for large data sizes. This issue becomes even more pronounced the larger the number of folds k is set. With increasing k, the models need not only to be trained more often, but also on larger sized training sets, since for example for $k = 4$ the training data makes up 75 % of the data, while it corresponds to 90 % of the whole sample for $k = 10$ (see for example Wainer and Cawley, 2017). Due to the size of the dataset used in this thesis, repeating the tuning process for every split separately would result in a runtime of several weeks for the ML algorithms described in Section 3. Therefore, all of the ML algorithms are tuned extensively on the whole dataset prior to running the actual estimation. The best performing set of hyperparameters is then chosen and held fixed for all the splits. In addition to reducing computational time, this approach is reasonable, since, while different hyperparameters may also lead to different performances of models on different data sets, it is likely that (nearly) optimal hyperparameters for the whole dataset yield similarly good results for the respective subsamples as well, especially in the present case, where each subsample corresponds to 50 % of the entire sample and is randomly drawn. As the dataset used in this thesis is fairly large, a value of $k = 3$ is chosen to balance computational time and reliability of the obtained performance measures. Furthermore, k-fold CV may suffer from high variance due to the random splitting into subsamples. One commonly used approach to reduce this variance is by repeating the CV process, called repeated CV (Kim, 2009). In repeated CV, the data shuffling process is repeated prior to splitting the sample into k folds, thus resulting in different data splits. Repeated CV is therefore used here with two repeats and $k = 3$.

The simplest approach for hyperparameter optimization is performing an extensive search over a prespecified grid, which defines possible values for each parameter. This procedure is called grid search. However, a full grid search often results in a very large number of parameter combinations, and thus is neither efficient nor even feasible in many cases, especially for algorithms with many hyperparam-

[22] There are also other resampling approaches commonly used for hyperparameter tuning, e.g. drawing bootstrap samples from the data. However, since CV is the most widely used approach it is also used in this thesis and described here.

eters. As an alternative to grid search Bergstra and Bengio (2012) proposed a more efficient way of hyperparameter optimization called random search. Instead of testing all possible combinations, random search randomly draws hyperparameters from a prespecified grid and evaluates only a fixed number of combinations. This makes the approach more efficient, especially in high-dimensional hyperparameter spaces with a low effective dimensionality, i.e. in settings where only some hyperparameters are actually relevant. Due to the design of grid search, only a few values are tested for each dimension, while random search picks a different value for each draw and each dimension.[23] Thus, even though the same number of combinations is tested, random search evaluates more distinct values in each dimension compared to grid search. Because of these reasons, the ML algorithms in the present work are tuned using random search over prespecified grids.[24]

The analysis in this thesis is conducted using the statistical programming language R (R Core Team, 2020). Tuning and training the ML models as well as the predictions in the main sample are done using the *caret* package (Kuhn, 2008). *caret* provides a unified interface for various ML methods and convenience functions for model training. Therefore, the library is especially suited for projects relying on multiple algorithms, as the present work. It does so by providing wrapper functions, which themselves call functions from other R libraries, in which the algorithms and techniques are actually implemented. The algorithms used in the empirical analysis of this thesis are implemented in the libraries *elasticnet* (Zou and Hastie, 2020) for elastic net, *ranger* (Wright and Ziegler, 2017) for random forests, *xgboost* (Chen et al., 2020) for XGBoost and *nnet* (Venables and Ripley, 2002) for neural networks. Since some algorithms are sensitive to the scaling of the input data, e.g. regularized linear models as explained in Section 3.2, all predictors are normalized to be in a range between 0 and 1 before tuning and training. Moreover, as recommended by, for example, LeCun et al. (2012), before training the neural networks, the outcome variable is transformed to be between 0 and 1 as well.

[23] For example, using grid search with two parameters and nine draws evaluates only three distinct values for each hyperparameter, while random search chooses nine distinct values for both parameters (for a visual representation of this example see also Bergstra and Bengio, 2012).

[24] In addition to grid search and random search there are also more sophisticated strategies for hyperparameter optimization. Such strategies often exploit techniques from Bayesian statistics and can be roughly seen as informed searches using the results from previous combinations for the choice of the next parameter combinations (see for example Shen et al., 2011; Kuhn, 2014; Snoek et al., 2012). However, such techniques are out of the scope of this thesis, and hence neither described nor used.

As discussed above, all algorithms are tuned using repeated 3-fold CV on the entire sample via random search, evaluating 50 parameter combinations. The only exception is the random forest algorithm. Probst et al. (2019b) show in a simulation study on tunability of different ML algorithms that although tuning RFs, as implemented in the *ranger* package, results in some slight improvements regarding their predictive power, these improvements are rather small and come at large computational costs. Moreover, RFs are known to work well out-of-the-box using default settings without much tuning (see also Probst et al., 2019a). Therefore, the RF implemented in the *ranger* package is applied using its default parameters. Referring back to the notation from Section 3, the tuning parameters for the algorithms, and their respective names in the *caret* package, are as follows: α ("fraction") and λ ("lambda") for elastic net; the number of boosting iterations ("nround"), the maximum depth of a single tree ("max_depth"), the learning rate parameter used for iteratively updating the prediction η ("eta"), the regularization parameter γ penalizing a larger number of leaves ("gamma"), the ratio of columns considered for each tree ("colsample_bytree"), the minimum sum of instance weights, i.e. the Hessian, needed in a node to keep splitting the data ("min_child_weight")[25], and the share of rows considered at each tree ("subsample") for XGBoost[26]; the number of hidden neurons ("size"), and the weight decay parameter λ ("decay") for the neural net; the number of variables considered at each split ("mtry"), and the minimum number of observations in terminal nodes ("min.node.size") for RF.[27] The tuning grids from which the hyperparameters are randomly drawn are prespecified in *caret*. Table 2 displays the possible values for all hyperparameters as well as the best performing hyperparameters found via the above stated tuning process and the default parameters for the RF.

Looking at Algorithm 1 in Figure 7 again, before splitting the sample into an auxiliary and a main sample, the propensity scores are to be computed. The estimation strategy was originally developed for randomized controlled trials (RCT). In such experiments the propensity scores, i.e. the probability of being treated,

[25] For regression tasks, this simply corresponds to the minimum number of observations in each terminal node, see also Section 3.3.3 and Chen and Guestrin (2016).

[26] *caret* does not allow to tune the L_2 regularization parameter λ, therefore its default value of 1 is used.

[27] *caret* does not allow to tune the number of trees grown in a forest since this parameter does not have a large impact on the forest's performance. Probst and Boulesteix (2017) show that the biggest improvements are made growing the first 100 trees and only small gains are achieved after this. However, growing more trees does not increase the risk of overfitting due to the low correlation between the single trees. Thus, it is common to grow a rather large number of trees, possibly even more than needed, to avoid limiting the algorithm's ability to pick up the relevant characteristics and patterns from the data. Thus, *caret*'s default of 500 trees is confidently used here as well.

Table 2: Hyperparameter grids and chosen values

Algorithm	Hyperparameter	Range		Best
		Min	Max	
Elastic Net	fraction	0	1	0.9518
	lambda	0.00001	10	0.00008
XGBoost	nround	1	1000	577
	max_depth	1	10	8
	eta	0.001	0.6	0.1687
	gamma	0	10	8.6623
	colsample_bytree	0.3	0.7	0.6726
	min_child_weight	0	20	3
	subsample	0.25	1	0.7392
Neural Net	size	1	20	14
	decay	0.00001	10	0.0143
Random Forest	mtry	1	# cols	$\sqrt{\#\text{ cols}}$
	min.node.size	1	20	5

Notes. The hyperparameters listed in the last column are found using repeated 3-fold CV, except the parameters for the RF, which correspond to the default parameters for regression settings, as implemented in the *ranger* package.

are usually set by the researchers, and thus known by definition. In the present analysis however, the dataset stems from observational data rather than from RCTs, hence these scores are not known. Following Deryugina et al. (2019), who used the same empirical strategy in a health economics related context estimating the effect of acute fine particulate matter exposure on mortality in the US, as well as Chernozhukov and co-authors themselves, who applied the strategy to the example of the gender wage gap[28], the propensity scores are estimated. The process follows the construction of the CATE proxy, thus a model for the propensity scores is estimated on the auxiliary sample and then the scores are predicted and further utilized in the main sample. Estimating propensity scores can be framed as a prediction problem, hence using ML models for this step might improve the estimation due to their superior predictive performance (Mullainathan and Spiess, 2017). Prior research on this topic indicates that ML can indeed outperform the most widely used approach of using logistic regression. For example, McCaffrey et al. (2004) compare the performance of propensity scores estimated via boosted trees with scores from logistic regression and find that the

[28] The application example changed in a later version of the paper, and thus cannot be found in the NBER Working Paper Series version, however the version can be found on arXiv, Chernozhukov et al. (2017).

first outperform the latter. Lee et al. (2010) also compare the performance of decision trees as well as bagged and boosted tree algorithms with logistic regression in a simulation study and obtain similar results. Therefore, in addition to standard logistic regression, RFs and XGBoost are considered for the estimation of propensity scores as well. One of the most important aspects of propensity scores is that they should account for differences between groups, usually a treatment and a control group, in observational settings to remove systematic differences and selection bias (Austin, 2011). Therefore, in order to choose the algorithm used to estimate the propensity scores in the empirical analysis, an experiment is conducted based on their ability to remove group differences. In a first step, the XGBoost algorithm is tuned on the whole sample in the same manner as the models for the proxy predictor using repeated 3-fold CV. The RF is again used with default parameters. In a second step, to mimic the actual estimation approach, the data is split into two equally sized subsamples, of which one sample is used to train the models and the other one to actually predict the propensity scores. The obtained scores are then used to weight the sample using inverse probability of treatment weighting (IPTW). In IPTW, treated observations are assigned a weight of $1/p_i$ and control group observations get a weight of $1/1 - p_i$ with p_i being the estimated propensity score for observation i (Austin and Stuart, 2015). If the propensity scores are estimated correctly, the weighted groups should be equal or at least close to being equal in terms of their covariates. In order to compare the groups after weighting, three different statistics are used and compared as balance diagnostics. Following Lee et al. (2010), the absolute differences between groups are calculated for the covariates and standardized by dividing by their pooled standard deviation. The resulting standardized differences are then averaged and referred to as the average standardized absolute mean difference (ASMD) (Stuart et al., 2013). This statistic is not affected by the sample size or the covariates' scale or units of measurement. A standardized difference of 0.1 or less is usually seen as acceptable and not a concern (Austin, 2011). Therefore, in addition to the ASMD, the number of covariates exceeding this threshold is taken into consideration as well. Furthermore, the so-called Kolmogorov-Smirnov (KS) test statistic is reported. It is defined as the maximum distance between the empirical cumulative distribution functions of continuous variables between two groups (Austin and Stuart, 2015). Similar to the ASMD, the average KS statistic over the continuous covariates is considered. Additional to the ASMD, the KS aims at comparing not only the means, but also takes into account higher order parameters of the distribution. To mitigate the possibility of obtaining a particular well-suited data split induced by random splitting, the experiment is

repeated ten times. Table 3 displays the average results over all ten runs for the logistic regression, XGBoost and RF.[29]

Table 3: Covariate balance before and after propensity score weighting

Criterion	Unadjusted	Logit	Random Forest	XGBoost
ASMD	0.1243	0.0248	0.0937	0.0605
KS	0.0880	0.0280	0.0672	0.0527
# Unbalanced	29.1	1.0	24.9	17.4

Notes: ASMD is the average of standardized absolute mean differences over all covariates. KS denotes the Kolmogorov-Smirnov statistic. # Unbalanced refers to the number of variables which exhibit a standardized absolute mean difference of more than 0.1. Smaller values indicate better balance after propensity score weighting for all three criteria. The reported results are means over 10 runs of the experiment, each with a different data split.

The unadjusted samples exhibit on average modest imbalances as indicated by the ASMD value of 0.1243. Weighting the observations with the estimated propensity scores improves the balance considerably for all algorithms. The propensity scores estimated via logistic regression reduce the ASMD to only 0.0280, which constitutes the lowest ASMD across models. XGBoost yields good results as well with an ASMD value of 0.0605, and also the RF algorithm manages to keep the ASMD for the adjusted sample below the threshold of 0.1 (0.0937). This ranking of algorithms is also represented by the number of unbalanced variables, i.e. variables with an absolute standardized difference of more than 0.1. In the unadjusted sample on average 29.1 variables are unbalanced. This number is reduced to only a single variable using logistic regression. Although also reducing the number of unbalanced variables, the RF and XGBoost still leave on average 24.9 and 17.4 covariates with larger absolute differences than 0.1, thus both are clearly outperformed by the logistic regression model. A very similar pattern is observed for the results of the KS statistic, where logistic regression yields the smallest value with 0.0280, followed by the XGBoost (0.0527) and the RF algorithm (0.0672).

These numerical indications can also be seen graphically in Figure 8 for one of the ten runs. Each grid of the plot compares the absolute standardized difference in means for the unadjusted and the propensity score weighted sample. The thresholds of 0.1 and -0.1 are indicated by the dotted lines. Compared to the RF and XGBoost, the adjusted means are on average closer to zero for the logistic regression. Moreover, the superior performance of this model in terms of the

[29] The comparison of the different algorithms for the propensity score estimation as well as the plots in Figure 8 are created using the *cobalt* package (Greifer, 2020).

Notes. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.

Figure 8: Balance grid

ASMD and the number of unbalanced variables after weighting becomes clear here as well. All in all, while all methods are able to improve balance in the adjusted sample compared to the unadjusted one, the logistic regression clearly outperforms both the RF and XGBoost algorithm. This is also consistent for all considered balance criteria. Therefore, logistic regression is used throughout the empirical analysis to predict the propensity scores. In order to avoid very large observation weights in the weighted regressions for the BLP and GATES parameters, observations with propensity scores below 0.05 and above 0.95 are removed from the estimation sample following Deryugina et al. (2019). This leads to the exclusion of on average 6.8 % of the utilized main samples.

As stated in Section 2.1, Chernozhukov et al. (2018) assume independence of treatment conditional on the observed covariates. The extensive dataset described in Section 4.1 contains a lot of factors which are likely to be determinants of house prices. At the same time, via the RWI-GEO-GRID dataset, the data base is extended by locality characteristics, which further influence property prices, while at the same time are very likely to have an impact on the construction of wind turbines in a specific location. For example, it is reasonable to assume that wind turbines are less likely to be placed in wealthier areas compared to poorer ones. This would be accounted for by the grid-level data on purchasing power. However, there might be further factors which are correlated with the placing of wind turbines, i.e. with treatment assignment, and the outcome, i.e. house prices. For example, there might be different local policies or restrictions, which differ between municipalities and influence the construction of wind power plants as well as the property prices. This would violate the unconfoundedness assumption and lead to unreliable results. To combat this issue, fixed effects are included in the estimation procedure. Following Frondel et al. (2019), these are included on the municipality-level.[30] The simplest approach to fixed effects is to include dummy variables for every level of the grouping variable. However, since the data comprises information on houses in 10,501 different municipalities, and since for many municipalities there are only very few observations, including a full set of dummy variables would result in a very large and sparse input matrix. This would lead to reduced predictive power for some ML algorithms and increases the number of predictors in the data drastically (Kuhn and Johnson, 2019). Therefore, the fact that fixed effects can also be implemented using the so-called within-transformation (see for example Gormley and Matsa, 2014), is

[30] The authors also provide evidence in favor of the unconfoundedness assumption after controlling for the locality characteristics obtained from the RWI-GEO-GRID data and municipality fixed effects by estimating multiple regression specifications and placebo regressions.

utilized in this thesis. The within-transformation computes the mean for each variable in each group and then subtracts these means from the levels for each observation. In order to ensure independence of the auxiliary and the main sample, the within-transformation is done for every split separately. At first, the means are computed on the auxiliary sample and used to demean this data. The predictive models are trained on this transformed sample. In a second step, the same means are then subtracted from the variables in the main sample and predictions are produced from this transformed sample. To get the predictions back to the original scale, the mean house prices in the corresponding municipalities from the auxiliary sample are again added to the predictions before constructing the CATE proxy. In order to ensure that all the municipalities which are present in the main sample also appear in the auxiliary sample, stratified data splitting on the municipality-level is used. This way, houses from all municipalities are present in both samples. That approach however comes with two possible problems: For municipalities, for which the data contains only a single observation, the stratified splitting does not work as intended, since the same house is not allowed to end up in both samples. Moreover, the within-transformation makes sense only if there are at least a few houses in a municipality. In the extreme case of only two houses in a municipality, the transformed house in the auxiliary sample would have values of zero for all variables after demeaning, since it is the only observation used for calculating the means. In order to avoid such cases and to have meaningful transformed data, municipalities with less than ten observations are excluded. This leads to the exclusion of 13,750 houses.[31] One typical issue in fixed effects estimation is that time-invariant attributes or variables which are common among all observations in a group, i.e. municipality, are removed from the model, thus inference on these is not possible. However, the estimation is a two-step procedure, and the first step is purely a prediction step, therefore there is no interest in estimated parameters, but rather in unconfoundedness, as described earlier. Thus, this is not considered a problem in the present case. In addition to the municipality fixed effects, year fixed effects are included as well. Since the time span of the data is eleven years only, year fixed effects are simply included as year dummies before performing the within-transformation (Gormley and Matsa, 2014).

[31] As an alternative to fixed effects on municipality-level, it would also be possible to use fixed effects on square kilometer grid-level. This would allow to control for unobserved factors on an even more detailed level. However, using the same approach of excluding grids with less than ten observations would result in the exclusion of 185,791 houses.

With respect to all of the previously discussed details and aspects, Figure 9 summarizes the implementation details and presents the adjusted estimation algorithm, as applied in this thesis.

Algorithm 2:

Input: Data on units $i = 1, \dots N$ with covariates Z_i and a binary treatment indicator D_i

Result: BLP, GATES, CLAN

```
begin
    Fix the number of splits S and significance level α, e.g. S = 100 and
    α = 0.05
    Split the data S times into equally sized subsamples Data_A, Data_M
    for s = 1 ... S do
        Train logistic regression in Data_A for propensity scores*
        Predict propensity scores in Data_M*
        Compute means for within-transformation in Data_A*
        Apply within-transformation to Data_A and Data_M*
        for Algo in ML Algorithms do
            Tune and train Algo in Data_A to learn B(Z_i) and S(Z_i)
            Predict B(Z_i) and S(Z_i) in Data_M
            Reverse within-transformation for predictions*
            Construct k groups based on proxy S(Z_i)
            Estimate BLP parameters in Data_M
            Estimate GATES parameters in Data_M
            Estimate CLAN parameters in Data_M
            Compute performance measures
        end
    end
    Compute medians of parameters of interest, confidence intervals,
    p-values using VEIN methods and medians of performance measures
end
```

Notes. Asterisks denote steps which are different from the original implementation algorithm as depicted in Figure 7.

Figure 9: Adjusted estimation algorithm

6 Results

Before presenting and discussing the results obtained during the empirical analysis, summary statistics for the sample used during the estimation are displayed in Table 4. This is done to give an overview and intuition about the average values of the variables, which is useful in order to put the results from the classification analysis into perspective later on. Using a treatment threshold of 5 km to the

Table 4: Summary statistics

Variable	Means		
	Full sample	Treatment	Control
Asking price	341,360	270,276	382,578
Year of advertisement	2013.5	2013.7	2013.4
Year of construction	1976.4	1976.1	1976.6
Living area	174.1	168.3	177.4
Lot area	683.9	730.8	656.7
Number of floors	2.2	2.1	2.3
Number of rooms	6.1	6.0	6.2
Number of bathrooms	1.9	1.9	1.9
Protected building (0/1)	0.01	0.01	0.02
Basement (0/1)	0.62	0.57	0.66
Property condition	4.9	4.9	4.9
Farmhouse (0/1)	0.01	0.01	0.01
Bungalow (0/1)	0.04	0.04	0.03
Semidetached house (0/1)	0.15	0.14	0.16
Single-family house (0/1)	0.48	0.52	0.46
Multi-family house (0/1)	0.11	0.10	0.12
Terraced house (0/1)	0.13	0.12	0.14
Category: Other (0/1)	0.05	0.05	0.04
Car density*	1.1	1.2	1.1
Purchasing power / capita*	22,962	21,871	23,594
Unemployment rate (%)*	5.2	5.7	4.9
Number of inhabitants*	1,848	1,583	2,002
Housing blocks (%)*	6.9	5.4	7.8
Skyscrapers (%)*	3	2.1	3.5
Age: 0–25 (%)*	24.4	24.4	24.5
Age: 25–40 (%)*	16.9	16.3	17.2
Age: 40–65 (%)*	37.6	38.2	37.3
Age: 65+ (%)*	21.0	21.0	21.0
# Observations	854,041	313,460	540,581

Notes. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables. For improved readability of this table, dummy variables constructed from the categorical variables federal state, category of the advertised house, and type of heating are excluded. The full table including all variables can be found in Table A6 in Appendix A.3.

nearest wind turbine, the treatment group consists of 313,460 houses, and thus 36.7 % of the entire sample. The control group comprises the remaining 540,581 houses or 63.3 %. Looking at the asking prices in both groups, it is noticeable that the house prices are substantially lower in the treatment group. Together with a lower purchasing power of around 21,871 € per capita in the treatment group compared to almost 23,000 € in the control group, this indicates that wind turbines might in fact rather be built in poorer areas as hypothesized in Section 5. Furthermore, they are more likely to be placed in less densely populated areas, as the average numbers of inhabitants (1,583 and 2,002, respectively) indicate. The average plot area is bigger for treated properties which is likely to directly stem from the fact that the treatment group consists of houses in less densely populated areas.[32] Other property characteristics, as for example the year of construction, living area, or number of floors, rooms and bathrooms, are similar in both groups. The same holds true for many of the remaining grid variables like the car density or the demographic composition of inhabitants.[33]

6.1 Estimation Results

As discussed in Section 3, four different algorithms are used to construct the CATE proxy: elastic net (ENet), random forests, XGBoost, and single-layer feed-forward neural networks. Table 5 displays the performance metrics for each of these algorithms. The first row contains the performance criterion Λ calculated in the BLP estimation following Equation (2.21), the second row the criterion $\overline{\Lambda}$ calculated from the GATES estimation as defined in Equation (2.23). As a reminder: maximizing Λ corresponds to maximizing the correlation between the true CATE function $s_0(Z)$ and the ML proxy $S(Z)$, while maximizing $\overline{\Lambda}$ is the same as maximizing the R-squared from regressing $s_0(Z)$ on the demeaned proxy $\overline{S}(Z)$. Thus, the higher these criteria are, the better does the respective algorithm approximate the CATE function. Considering Λ, the ENet achieves a figure of 482,478,151, and thus outperforms the other algorithms by far. The RF (159,749,816) seems to perform a bit better than XGBoost (123,526,210), while the neural network lags far behind (16,032,440). In terms of the second criterion, $\overline{\Lambda}$, ENet (71,684,309) again produces the highest number. Moreover, the neural network (41,131,787) predicts the CATE function slightly better than the tree-based algorithms with Λ-values of 29,304,652 for the RF, and 25,459,158 for XGBoost. All in all, the elastic net results by far in the highest figures for both criteria, and hence the proxy constructed via ENet provides the best ap-

[32] Of course, this could also be the other way round: The lower number of inhabitants per grid could be a result of properties with larger plot areas.

[33] For a complete overview see also Table A6 in Appendix A.3.

proximation to the true CATE function. Therefore, this algorithm is considered producing the most reliable and credible results, and is chosen as the preferred specification. Although it clearly dominates the other techniques in terms of both criteria, the estimation results from the tree-based algorithms RF and XGBoost are presented in the further course of this section as well, as recommended by Chernozhukov et al. (2018). The estimates obtained via the neural network are excluded due to its overall poor performance.

Table 5: Performance metrics for ML algorithms

	Elastic Net	Random Forest	XGBoost	Neural Net
BLP - Λ	482,478,151	159,749,816	123,526,210	16,032,440
GATES - $\bar{\Lambda}$	71,684,309	29,304,652	25,459,158	41,131,787

Notes. The reported results are medians over 100 splits. Higher numbers indicate better performance for both criteria.

The results from the BLP estimation are presented in Table 6. It shows the medians for the coefficients, confidence intervals (CI), and p-values computed using the VEIN methods described in Section 2.5. Thus, although originally being 95 % CIs, their nominal significance level is reduced to 90 % to account for the splitting uncertainty. The reported p-values are sample splitting adjusted p-values. The coefficients for the BLP of the CATE function using the proxy constructed via ENet are -6,195 for the intercept β_1, and 0.316 for the slope β_2. The CIs for the parameters are far from zero, and both are significant at any conventional significance level, indicated by the adjusted p-values of 0.000. Since β_1 identifies the ATE, these results imply that being treated, i.e. a house being located in the range of 5 km to the nearest wind turbine, reduces the average asking price by around 6,200 €. Further, the result for β_2 indicates that first, there exists heterogeneity in treatment effects, and second, this heterogeneity is also successfully predicted by the elastic net algorithm. Similar to these results, the XGBoost algorithm yields a significant and negative ATE as well, however with a coefficient of -2,657 it is lower than the ATE estimated via ENet. In contrast to that, the RF does not identify a negative ATE. The estimated coefficient of -21 is further very close to zero and not statistically significant. Overall, the estimated ATEs are rather small, keeping in mind that the average asking price in the sample is around 340,000 € (Table 4). As for ENet, the heterogeneity coefficients β_2 (HET) are significantly different from zero for both tree-based algorithms as well, again implying that there exists heterogeneity in the treatment effects of wind turbines on property prices. However, both are considerably smaller compared to the HET coefficient derived from ENet with 0.143 for RFs and 0.072 for XGBoost. This again illustrates that both algorithms provide a much worse

approximation to the true CATE function, and that the elastic net captures the effect heterogeneity by far the best. Nonetheless, the results are consistent across algorithms in the sense that they all provide evidence in favor of effect heterogeneity.

Table 6: BLP estimation results

	Elastic Net	
	ATE (β_1)	HET (β_2)
Coefficients	-6,195	0.316
Confidence bands	(-7,499 ; -4,868)	(0.252 ; 0.379)
Adjusted p-values	[0.000]	[0.000]
	Random Forest	
	ATE (β_1)	HET (β_2)
Coefficients	-21	0.143
Confidence bands	(-1,079 ; 993)	(0.130 ; 0.157)
Adjusted p-values	[0.810]	[0.000]
	XGBoost	
	ATE (β_1)	HET (β_2)
Coefficients	-2,657	0.072
Confidence bands	(-3,706 ; -1,611)	(0.064 ; 0.079)
Adjusted p-values	[0.000]	[0.000]

Notes. The reported results are medians over 100 splits. The CIs are at the 90 % significance level. Adjusted p-values are from t-tests with H_0: estimated coefficient is equal to zero.

Table 7 presents the results for the estimation of the group average treatment effects. The groups are constructed based on the deciles of the proxy $S(Z)$, thus there are ten groups. Assuming a negative effect of wind turbines on house prices, as suggested by the existing literature (see also Section 1), the most affected group, i.e. the group with the largest negative effect, is identified by the observations in the first decile, while the least affected group is given by the properties in the last decile. Table 7 shows the estimated effects for the most and least affected groups as well as the difference between both groups. Starting again with the elastic net, the point estimate for the least affected group still suggests a small negative treatment effect of -3,760, however not significant, as indicated by the CI, which includes zero, and the adjusted p-value of 0.163. The ATE in

Table 7: GATES estimation results

	Elastic Net		
	Least affected (γ_{10})	Most affected (γ_1)	Difference ($\gamma_{10} - \gamma_1$)
Coefficients	-3,760	-20,419	16,031
Confidence bands	(-7,950 ; 494)	(-24,379 ; -16,521)	-
Adjusted p-values	[0.163]	[0.000]	[0.000]
	Random Forest		
	Least affected (γ_{10})	Most affected (γ_1)	Difference ($\gamma_{10} - \gamma_1$)
Coefficients	5,124	-13,028	18,410
Confidence bands	(1,491 ; 8,717)	(-16,571 ; -9,552)	-
Adjusted p-values	[0.010]	[0.000]	[0.000]
	XGBoost		
	Least affected (γ_{10})	Most affected (γ_1)	Difference ($\gamma_{10} - \gamma_1$)
Coefficients	-1,434	-12,956	11,900
Confidence bands	(-4,721 ; 1,885)	(-16,431 ; -9,446)	-
Adjusted p-values	[0.167]	[0.000]	[0.000]

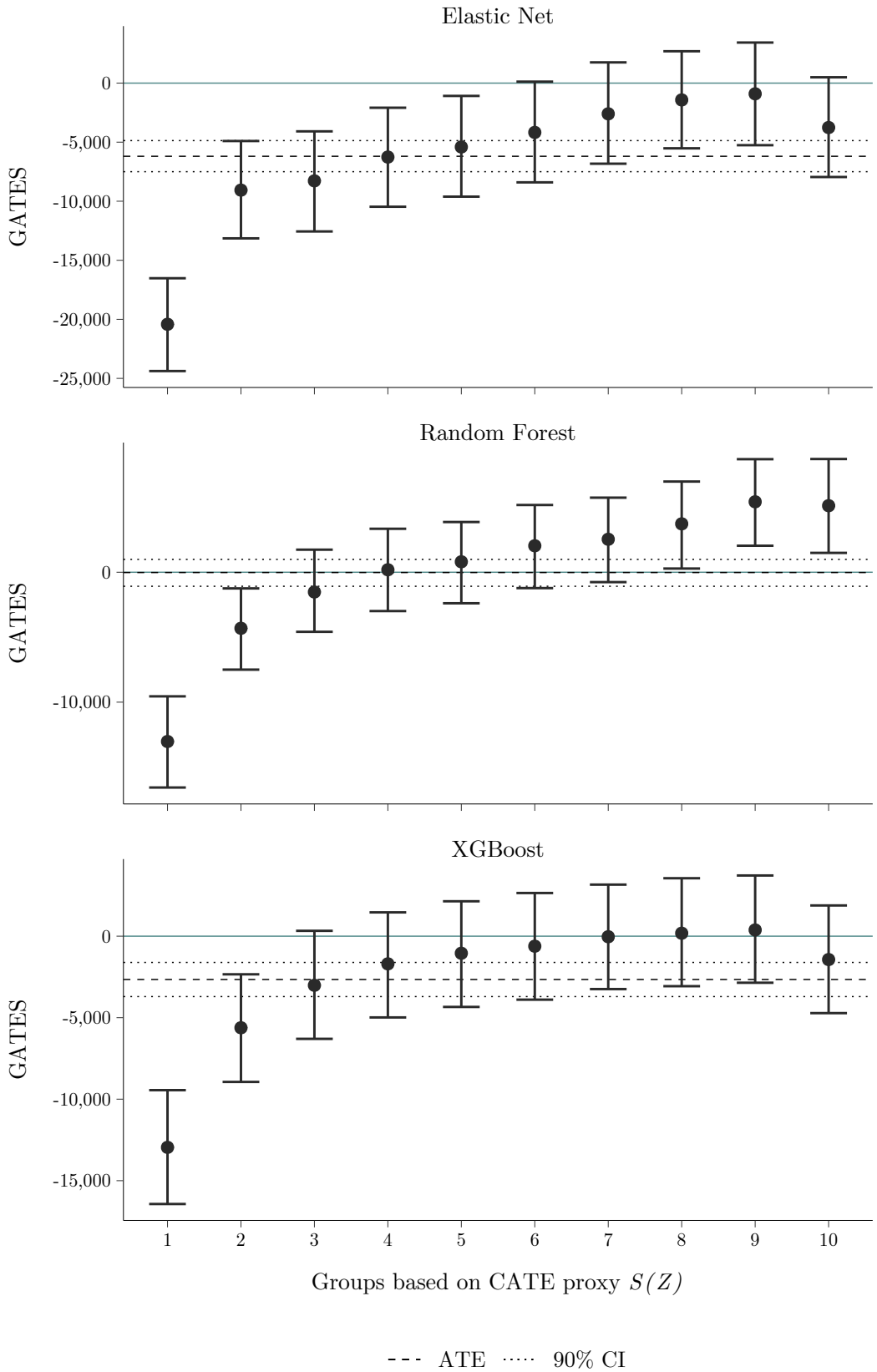
Notes. The reported results are medians over 100 splits. The CIs are at the 90 % significance level. Adjusted p-values are from t-tests with H_0: estimated coefficient is equal to zero. Since a negative effect of wind turbines on house prices is assumed, the most affected group is the group with the largest negative effect proxy, i.e. the first decile.

the most affected group however is substantial in both statistical significance and its magnitude. It is significant at any conventional significance level given the adjusted p-value of 0.000 and amounts to -20,419. The difference between both groups is estimated to be 16,031, i.e. houses in the most affected group react to wind turbines in the range of up to 5 km with a price reduction, which is around 16,000 € higher compared to the least affected group. In line with the ENet results, XGBoost also does not find significant positive treatment effects, even in the least affected group. The point estimate is negative (-1,434), although again not significant. Additionally, the estimated group average treatment effect in the first decile is significant and negative with -12,956. The estimated difference is 11,900 and significant as well. The RF is the only algorithm which finds significant positive treatment effects in the least affected group with a significant point

estimate of 5,124 and an adjusted p-value of 0.010. However, in line with the other two algorithms, an estimation result of -13,028 reveals negative effects in the most affected group as well. Due to the positive estimate in the last decile, the difference between the groups is 18,410, and thus the largest among all three algorithms. Therefore, all algorithms find large and significant differences between the most and least affected groups as well as substantial negative effects for the most affected group, which range from -12,956 to -20,419. Only the RF finds positive treatment effects at all, which is however not confirmed by the other two models.

Additional to the results on the HET coefficient β_2 from the BLP estimation discussed earlier, the GATES hence provide further evidence in favor of treatment effect heterogeneity. This is also supported by an F-test on the equality of all the GATES coefficients in the deciles: the adjusted p-values derived from the elastic net, random forest and XGBoost are all very close to zero, thus the hypotheses that these do not differ from each other is rejected. Furthermore, Figure 10 plots the GATES for all ten groups and the three algorithms considered in more detail.[34] The point estimates are illustrated by the dots, while the 90 % CIs are represented by the errorbars. The dashed and dotted lines denote the ATEs and the upper and lower bounds of the 90 % CIs obtained from the BLP estimation. The pattern of the estimated parameters is almost as expected given the monotonicity assumption introduced in Section 2.3, i.e. the GATES are increasing with the groups based on the CATE proxy $S(Z)$. The only exceptions are the estimates in the least affected groups using elastic net and XGBoost, which are slightly lower than the coefficients for the second-least affected groups, however both parameters are not statistically different from zero, as indicated by their CIs. All algorithms find strong negative and significant treatment effects in the most affected group, i.e. for the properties in the first decile of the CATE proxy. As discussed above, only the approach using the RF estimates significant positive effects, while the other two algorithms do not find any positive impact of nearby wind turbines on house prices. Furthermore, the estimations building on elastic net and XGBoost suggest significant negative treatment effects up to group 5 and group 2, respectively. In many groups the GATES differ significantly from each other as demonstrated by the non-overlapping CIs of the respective groups. The estimated coefficients in the first decile even differ significantly from all other group average treatment effects for all three algorithms. All in all, Figure 10 is seen as additional evidence of treatment effect heterogeneity.

[34] The full estimation results for all GATES coefficients including CIs and adjusted p-values can be found in Appendix A.3 in Table A8.



Notes. The dots represent the point estimates for the ATEs in the respective groups. Errorbars denote 90 % CIs. Point estimates and CIs are medians over 100 splits.

Figure 10: GATES results

Continuing with the classification analysis, Table 8 presents the arithmetic means for the analyzed variables in the most and the least affected groups derived from the estimation based on elastic net. Since the elastic net provides the best approximation to the CATE function, as discussed above, the CLAN results are reported for this algorithm only for reasons of improved readability. However, although less pronounced, the results are qualitatively similar for the other algorithms, if not explicitly stated differently. Moreover, the full results can be found in Appendix A.3 in Table A9. The difference of the means between the most and least affected group is statistically significant for every variable, even if it is rather small in absolute size, which is probably due to the large number of observations in the sample. Therefore, CIs and p-values are not explicitly reported here.

Regarding the covariates, the elastic net finds large differences in the average number of inhabitants per grid. Properties in the first decile of the proxy are located in grids with on average 2,473 inhabitants, while the average number of inhabitants is only 1,896 in the least affected group. This indicates that prices of houses in more densely populated areas are affected more strongly than house prices in grids with a lower population density. This is also supported by the results on the shares of semidetached and terraced houses, which are lower in the most affected groups (4 % and 3 % in decile 1, 24 % and 30 % in decile 10), as well as by the results on multi-family houses, which is higher in this group (22 % and 7 %, respectively). In addition to that, another large difference is found in the average asking prices of houses in both groups. The mean price of a property in the most affected group is over 600,000 €, and thus more than twice as high as the average price in the least affected group (289,705 €). This is also reflected by the fact that the share of villas (19 %) in the most affected group is almost seven times as high as the overall share in the entire sample (3 %), while the share in the last decile is only 0.2 %, and thus substantially smaller than the overall mean. These findings however are not surprising. All of the results discussed until this point stem from a linear specification using the absolute house prices as the outcome variable throughout the prediction and estimation steps. Therefore, it is reasonable to assume that the size of the treatment effects is strongly driven by the level of the house prices. This may lead to somehow misleading results, since although the absolute effect size might be larger for more valuable properties, the relative effect size might actually be fairly small and much higher for less expensive properties. Another concern with this linear approach is that a large amount of the effect heterogeneity may therefore get absorbed by the levels of the asking prices.

Table 8: CLAN results

Variable	Elastic Net	
	Most	Least
Asking price	612,362	289,705
Year of construction	1965.8	1969.9
Protected building (0/1)	0.04	0.01
Purch. power / capita*	24,232	22,654
Unemployment rate (%)*	5.6	5.2
# Inhabitants*	2,473	1,896
Skyscrapers (%)*	4.1	3.7
Housing blocks (%)*	8.6	9.1
Age: 0-25 (%)*	24.2	24.4
Age: 25-40 (%)*	17.6	17.2
Age: 40-65 (%)*	37.1	37.1
Age: 65+ (%)*	21.1	21.3
Farmhouse (0/1)	0.01	0.01
Bungalow (0/1)	0.02	0.03
Semidetached house (0/1)	0.04	0.24
Single-family house (0/1)	0.42	0.27
Multi-family house (0/1)	0.22	0.07
Terraced house (0/1)	0.03	0.30
Category: Other (0/1)	0.07	0.05
Villa (0/1)	0.19	0.002
Property condition	5.2	5.1

Notes. The reported results are medians over 100 splits. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.

In order to combat this concern, additional specifications using a logarithmic-transformed outcome are estimated. The approach and strategy are identical to the estimation procedure described earlier, the only difference is that the asking prices are log-transformed prior to the estimation.[35] This logarithmic specification facilitates the interpretation of the treatment effects, since the estimated parameters then approximate relative changes instead of effects in absolute asking

[35] The results from the proofs provided in Appendices A.1 and A.2 still hold in this case. The only difference is that the treatment effect is no longer defined as $s_0(Z) := \mathbf{E}[Y(1) \mid Z] - \mathbf{E}[Y(0) \mid Z]$, but instead as $s_0(Z) := \mathbf{E}[log(Y(1)) \mid Z] - \mathbf{E}[log(Y(0)) \mid Z]$. Therefore, all the steps and operations are still valid and work as explained. Hence, β_1 and β_2 still define the BLP of s_0 given S(Z), and γ_k still estimate the GATES parameters.

prices. Since only the dependent variable is log-transformed and all predictors enter the model in their linear form, the regressions are log-linear models. In such models, an estimated coefficient β has the interpretation that a unit change in the respective predictor is accompanied by (approximately) a $100 * \beta\,\%$ change in the outcome. Since the treatment indicator is binary, a one-unit change corresponds to a change in treatment status, and thus the parameters β_1 from the BLP and γ_k are just the approximated (group) average treatment effects in percent. The exact percentage change can be calculated from the parameters using the formula $\%\Delta y = 100 * (e^{\beta} - 1)$ (see for example Wooldridge, 2016).

Table 9: BLP estimation results - log specification

	Elastic Net	
	ATE (β_1)	HET (β_2)
Coefficients	-0.007	0.328
Confidence bands	(-0.010 ; -0.005)	(0.257 ; 0.403)
Adjusted p-values	[0.000]	[0.000]
	Random Forest	
	ATE (β_1)	HET (β_2)
Coefficients	0.002	0.068
Confidence bands	(-0.0003 ; 0.003)	(0.053 ; 0.083)
Adjusted p-values	[0.175]	[0.000]
	XGBoost	
	ATE (β_1)	HET (β_2)
Coefficients	-0.007	0.050
Confidence bands	(-0.009 ; -0.005)	(0.022 ; 0.080)
Adjusted p-values	[0.000]	[0.001]

Notes. The reported results are medians over 100 splits. The CIs are at the 90 % significance level. Adjusted p-values are from t-tests with H_0: estimated coefficient is equal to zero.

The results of the BLP estimation for the logarithmic specifications, again utilizing ENet, RFs and XGBoost to construct $S(Z)$, are shown in Table 9. As before, the ENet and XGBoost algorithms reveal significant negative ATEs of -0.7 %, while the RF-based approach results in a positive point estimate this time, however still insignificant and very close to zero. Log-transforming the asking prices

does not seem to affect the ability to approximate the CATE function when using ENet, as indicated by similar parameters for β_2 (0.328 and 0.316, respectively). The approximation using the tree-based methods however gets worse. While the linear specification yielded parameters of 0.143 for the RF and 0.072 for XGBoost, the log-transformed outcome models result in coefficients of only 0.068 and 0.050, respectively. However, although these are smaller than in the linear models, all HET coefficients are again different from zero and highly significant, which is again taken as evidence in favor of the existence of treatment effect heterogeneity.

Analogous to Table 7, Table 10 presents the results on the GATES estimation using the logarithmic specifications. The preferred model building on ENet again does not identify statistically significant positive treatment effects in the least affected group. The average treatment effect in the first decile on the other hand is both highly significant and also substantial in size with -2.0 %. This results in a difference of 2.1 percentage points between the groups. The XGBoost yields similar results. Although it reveals negative point estimates even in the last decile of the proxy, it is not significant with a p-value of 0.363. Furthermore, in line with ENet, a significant negative ATE is found in the first decile, which amounts to -1.5 %. Overall, the point estimates from XGBoost are smaller in size and less divergent with a difference of only 1.1 percentage points between the first and the last group. In contrast to ENet and XGBoost, which find negative effects only, the estimation results from the RF again suggest substantial positive treatment effects of up to 1.9 % in the least affected group. However, the ATE in the most affected group is estimated to be -1.5 % and significant as well. The large spread across the groups also results in the largest estimated difference for all algorithms of 3.1 percentage points.

The results are mostly consistent across algorithms. All models reveal significant negative treatment effects in the most affected group. The RF is the only model, which suggests positive effects for parts of the sample, however this is neither confirmed by XGBoost nor by ENet. Furthermore, all algorithms reject the hypothesis of no heterogeneity in treatment effects. This is again visualized in Figure 11. The first panel plotting the GATES from the ENet model reveals results, which are similar to the linear specification: significant negative group average treatment effects are found up to the fifth decile, while all other groups are not significantly affected, indicated by the CIs including zero. The XGBoost panel reveals only little variation in the estimated GATES, as already indicated by Table 10. The properties in the first four groups are estimated to

Table 10: GATES estimation results – log specification

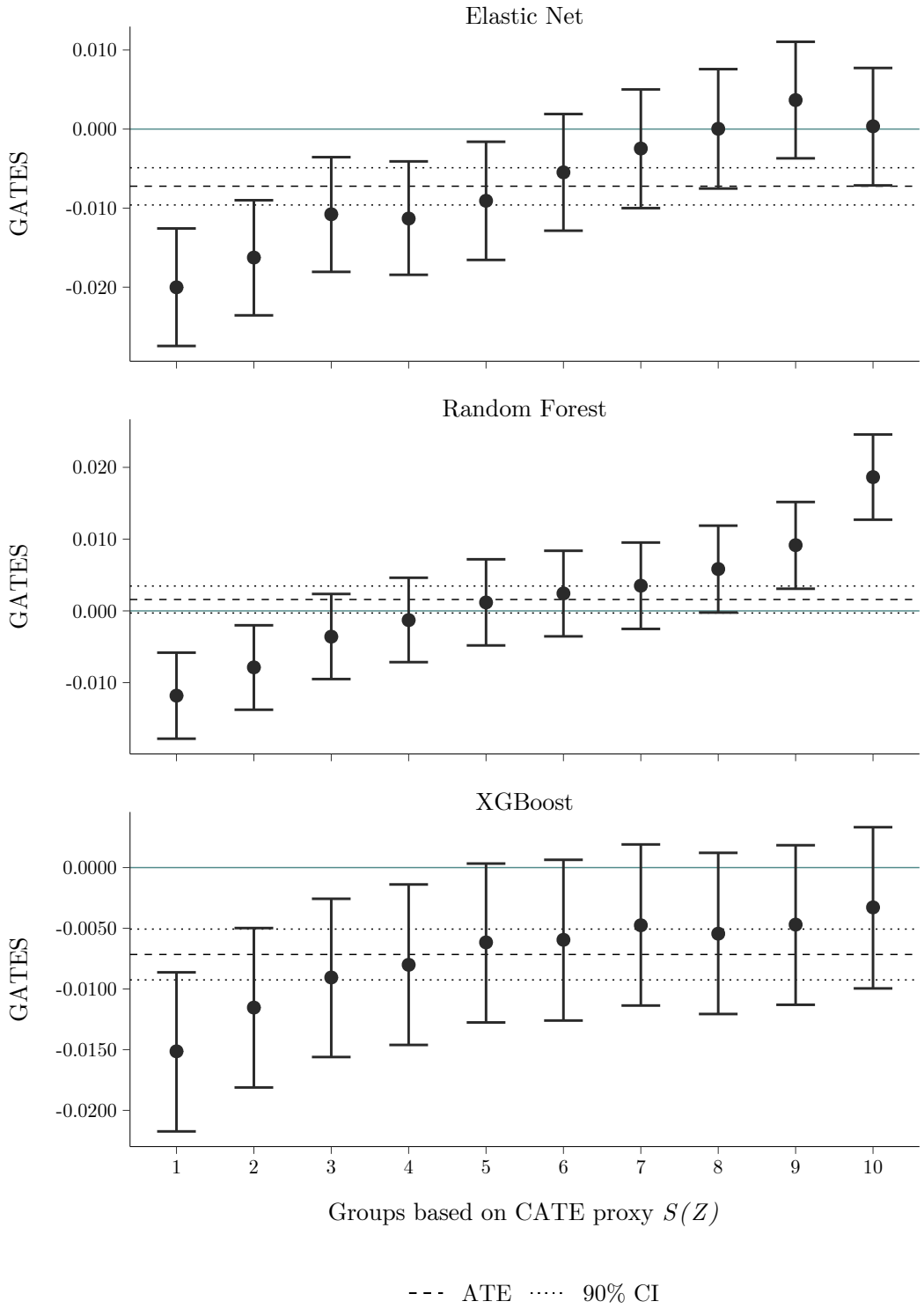
	Elastic Net		
	Least affected (γ_{10})	Most affected (γ_1)	Difference ($\gamma_{10} - \gamma_1$)
Coefficients	0.000	-0.020	0.021
Confidence bands	(-0.007 ; 0.008)	(-0.027 ; -0.013)	-
Adjusted p-values	[0.363]	[0.000]	[0.000]
	Random Forest		
	Least affected (γ_{10})	Most affected (γ_1)	Difference ($\gamma_{10} - \gamma_1$)
Coefficients	0.019	-0.012	0.031
Confidence bands	(0.013 ; 0.025)	(-0.018 ; -0.006)	-
Adjusted p-values	[0.000]	[0.000]	[0.000]
	XGBoost		
	Least affected (γ_{10})	Most affected (γ_1)	Difference ($\gamma_{10} - \gamma_1$)
Coefficients	-0.003	-0.015	0.011
Confidence bands	(-0.010 ; 0.003)	(-0.022 ; -0.009)	-
Adjusted p-values	[0.376]	[0.000]	[0.029]

Notes. The reported results are medians over 100 splits. The CIs are at the 90 % significance level. Adjusted p-values are from t-tests with H_0: estimated coefficient is equal to zero. Since a negative effect of wind turbines on house prices is assumed, the most affected group is the group with the largest negative effect proxy, i.e. the first decile.

suffer from a slight decrease in asking prices, while no effects are found for all remaining groups. Moreover, none of the GATES differ significantly from the overall ATE, as indicated by the overlapping CIs of the GATES and the ATE from the BLP estimation. The GATES from the RF show the above discussed differences compared to the ENet: while statistically significant negative effects are detected in the first two deciles as well, significant positive point estimates begin in the ninth decile and increase further in the last group. Although not as clearly as Figure 10, Figure 11 still provides evidence in favor of existing treatment effect heterogeneity as well. The estimated ATEs in the first decile still differ from ATEs in many other groups for ENet and RFs. The GATES are only in the case of the XGBoost-based approach not significantly different from each

other. However, since especially the elastic net provides a much better approximation to the CATE, as also implied by the much larger HET parameter β_2 from the BLP estimation, the results still strongly point towards the existence of heterogeneity, especially when considered in combination with the results from Table 10. The lacking heterogeneity from the XGBoost is very likely to originate from the poor approximation to the CATE, as indicated by the very small HET coefficient of 0.050, and thus not seen as strong evidence against treatment effect heterogeneity.

Lastly, the CLAN results from the logarithmic models are presented in Table 11. As before, only the classification analysis from the elastic net is shown here. Again, the full results for all algorithms can be found in Appendix A.3 in Table A11. The first thing to notice is that the asking prices differ much less in the most and least affected groups compared to the linear specification. This implies that higher-priced or more valuable properties do not necessarily suffer more strongly from nearby wind turbines when examining effects relative to the properties' values instead of treatment effects in absolute values. In fact, the classification analysis implies that the relative price decrease for houses in a range of 5 km to the nearest wind turbine is even stronger for less expensive houses. Such a finding would imply undesired allocation effects. More valuable properties are more likely to be owned by wealthier owners and owners with higher income. Thus, if less valuable properties' prices would additionally be reduced by a larger amount, wind turbines would affect less wealthy people disproportionately more. However, this result is not consistent across algorithms (Table A11) and is also not supported by the average grid-level purchasing power, which is roughly identical, or even slightly higher in the most affected group. Moreover, the CLAN from the log-linear model reveals further heterogeneity in the covariates: while the average year of construction in the least affected group is slightly above the overall sample average with 1983.5, the mean construction year in the most affected group is much lower (1944.1). Therefore, especially old houses are more severely affected by the proximity to wind turbines compared to newer houses. Moreover, the share of protected historic buildings is twice as high in the most affected group (4 % and 2 %, respectively). This difference might simply be a result from the much lower average construction year, however it may also indicate, that houses, whose appearance is in sharp contrast to the more modern look of wind turbines and the landscape they are built in, undergo a larger price reduction. Furthermore, the large difference in the average number of inhabitants per grid becomes apparent in the log-linear specification as well with an average of 2,596 in the first, and of only 1,693 in the last decile. Thus, the difference is even more pronounced in



Notes. The dots represent the point estimates for the ATEs in the respective groups. Errorbars denote 90 % CIs. Point estimates and CIs are medians over 100 splits.

Figure 11: GATES results – log specification

this specification, and further also consistent across algorithms. As before, this is also in line with the results on the shares of semidetached, multi-family,, and terraced houses, and is additionally supported by a higher share of housing blocks in the most affected grids (8.6 % and 7.7 %, respectively). These results reveal that properties in more densely populated areas are affected on average much more strongly than houses in more rural areas. One possible explanation is that owners and buyers of such properties in more rural areas might simply be more used to the look of wind turbines and their visual impact on the landscape, and do not perceive them negatively. The share of farmhouses, on the other hand, is higher in the least affected group. This might reflect the fact that wind turbines are sometimes co-owned by farmers or are built on properties owned by farmers. However, such wind turbines are often only small plants with a nominal capacity of less than 30 kW, and such power plants are excluded from the data on wind turbines due to data privacy reasons. Therefore, this might also result from the same reasons as above, namely that farmers and people living in or buying farmhouses simply do not perceive wind turbines as negatively as, for example, people living in cities, in which wind turbines constitute rather unusual sights.

Another noticeable difference is that values of properties, which are in better condition, indicated by the average property condition of 4.2 out of 10 in the last decile compared to 6.2 out of 10 in the first decile, are stronger affected by nearby wind turbines. This seems intuitive since there are more urgent and striking issues than possibly being somewhat close to a wind power plant for owners and potential buyers of properties which are in poor condition or in need of renovation. Interestingly, there are no substantial differences regarding the average demographic composition of grids in both groups. One could hypothesize that younger people might be more aware of climate change and GHG emissions related problems, as indicated for example by the Fridays for Future movement. Doing so, one would expect to find a higher willingness-to-accept nearby wind turbines in order to promote environmentally friendly energy generation among younger people, which in turn would result in a less negative treatment effect of nearby wind turbines on the house prices. However, this hypothesis cannot be confirmed with the data at hand and the results found here.

To summarize, all three algorithms, elastic net, random forest and XGBoost, provide evidence in favor of substantial heterogeneity of the effects of wind turbines on house prices for properties located in a range of maximum 5 km to the nearest wind power plant. The ENet- and XGBoost-based models estimate rather small, but significant ATEs, while the RF does not find a significant ATE. However,

Table 11: CLAN results – log specification

Variable	Elastic Net	
	Most	Least
Asking price	12.6	12.8
Year of construction	1944.1	1983.5
Protected building (0/1)	0.04	0.02
Purch. power / capita*	23,797	22,606
Unemployment rate (%)*	5.7	5.1
# Inhabitants*	2,596	1,693
Skyscrapers (%)*	3.5	3.6
Housing blocks (%)*	8.6	7.7
Age: 0-25 (%)*	24.2	24.5
Age: 25-40 (%)*	17.7	17.0
Age: 40-65 (%)*	37.0	37.5
Age: 65+ (%)*	21.2	21.1
Farmhouse (0/1)	0.01	0.03
Bungalow (0/1)	0.02	0.04
Semidetached house (0/1)	0.07	0.18
Single-family house (0/1)	0.39	0.45
Multi-family house (0/1)	0.25	0.10
Terraced house (0/1)	0.13	0.07
Villa (0/1)	0.05	0.06
Category: Other (0/1)	0.07	0.06
Property condition	6.2	4.2

Notes. The reported results are medians over 100 splits. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.

according to the criteria to choose the best ML algorithm provided by Chernozhukov et al. (2018), the elastic net outperforms both tree-based algorithms. Furthermore, this is also supported by the β_2 parameter from the BLP estimation, which suggests that ENet is the most suitable and best predictor of treatment effect heterogeneity in the presented work. Since the elastic net hence provides by far the best approximation to the true CATE function, it is seen as the preferred algorithm in this case and the estimation results from models building on CATE proxies constructed via ENet are deemed the most credible ones.

Regarding the specification of the outcome variable, i.e. the properties' asking prices, it is reasonable to assume that in a standard linear-linear model these prices absorb much of the treatment effect heterogeneity. In order to avoid this, and also to facilitate the interpretation of the estimated effects, log-linear models are additionally estimated. Using such models, ENet and XGBoost again find small ATEs of -0.7 %, while the ATE estimated via RFs is not statistically significant. Furthermore, all algorithms provide evidence in favor of effect heterogeneity as well. Due to the same reasoning as above, the elastic net is again considered providing the most reliable and credible results. The parameters of the GATES estimation utilizing ENet reveal substantial negative treatment effects of -2.0 % in the most affected group. Such negative treatment effects are moreover observed up to the fifth decile, i.e. for 50 % of the analyzed properties. While the RF-based models suggest positive treatment effects of up to 1.9 %, this is not confirmed by any of the other algorithms. Both do not find positive impacts of wind turbines at all and instead provide evidence in favor of negative treatment effects only. Therefore, although not entirely consistent across algorithms, there is strong evidence for a negative effect of wind turbines on the prices of nearby houses.

Furthermore, the classification analysis reveals interesting differences in the composition of the most and least affected group. Houses located in on average more densely populated areas are more strongly affected. The same holds true for much older houses, protected historic buildings and properties in a better condition. Moreover, slightly less valuable properties tend to experience a more severely price reduction when wind power plants are located in their proximity, albeit this is not consistent across algorithms and cannot be stated with certainty. All in all, the difference in the estimated GATES indicate that there is a difference of up to 3.1 percentage points regarding the treatment effect between properties in the most and least affected groups.

The obtained results are only partly in line with the findings from Frondel et al. (2019). On the one hand, the authors estimate larger negative treatment effects for houses which are built before 1950. This is supported by the results from the empirical analysis in this thesis. Properties in the most affected group are on average built in 1944, while the average construction year in the least affected group is 1984. On the other hand, Frondel et al. (2019) also find stronger negative treatment effects for houses in rural areas, measured as being further away than 10 km from the nearest city center. This finding, however, is diametral to the results obtained here. These reveal much stronger impacts of wind turbines on

prices of properties, which are on average located in much more densely populated grids. This is also consistent with other covariates, as for example higher shares of large housing blocks or multi-family houses in the most affected group, and also across all algorithms used to construct the CATE proxy. Moreover, in contrast to Frondel et al. (2019), the estimated effects are substantially smaller here. While the authors find effects of up to -7.1 % for houses in a range of 1 km to the nearest wind turbine and even -20.9 % for houses built before 1950, the largest negative effects found here are -2.0 % in the most affected decile. Given the extremely high prices of houses in general, compared to almost all other purchase decisions, such a moderate treatment effect of a few percents appears to be more reasonable, especially assuming that the main effect stems from visibility effects and negatively perceived landscape disturbances (Gibbons, 2015).

6.2 Robustness Checks

In addition to the main estimations, of which the results are discussed in the previous section, two further specifications are estimated as robustness checks. For both these robustness checks only the preferred estimation approach is used, i.e. the CATE proxy is constructed via the elastic net algorithm and the asking prices are log-transformed, thus the coefficients can be interpreted as approximate percentage changes.

The first additional estimation is related to the sizes of the treatment effects. In the main specifications depicted above, the properties are divided into 10 groups based on deciles of the CATE proxy. However, it might be possible that the properties in the first decile, i.e. the most affected group, are not evenly affected by the negative treatment effect of -2.0 %, but that this effect is rather driven by a small fraction of this group, which reacts very strongly, while the majority of the properties are affected to a much lesser extent. In order to verify or reject this hypothesis, the estimation is repeated, however, the main sample is not divided into 10 groups only, but into 50 groups. As expected, the number of groups has no impact on the best linear predictor, thus the BLP estimation yields the same results, and is therefore not presented again here. The results from the GATES estimation are reported in Table 12.

These still reveal no positive treatment effects in the least affected group, here identified by the 2 % of observations with the largest CATE proxy values. Albeit positive (0.004), the point estimate is close to zero and far from statistical significance with an adjusted p-value of 0.199. On the contrary, the GATES coefficient for the most affected group is the largest negative effect identified up to now with

Table 12: GATES estimation results – 50 groups

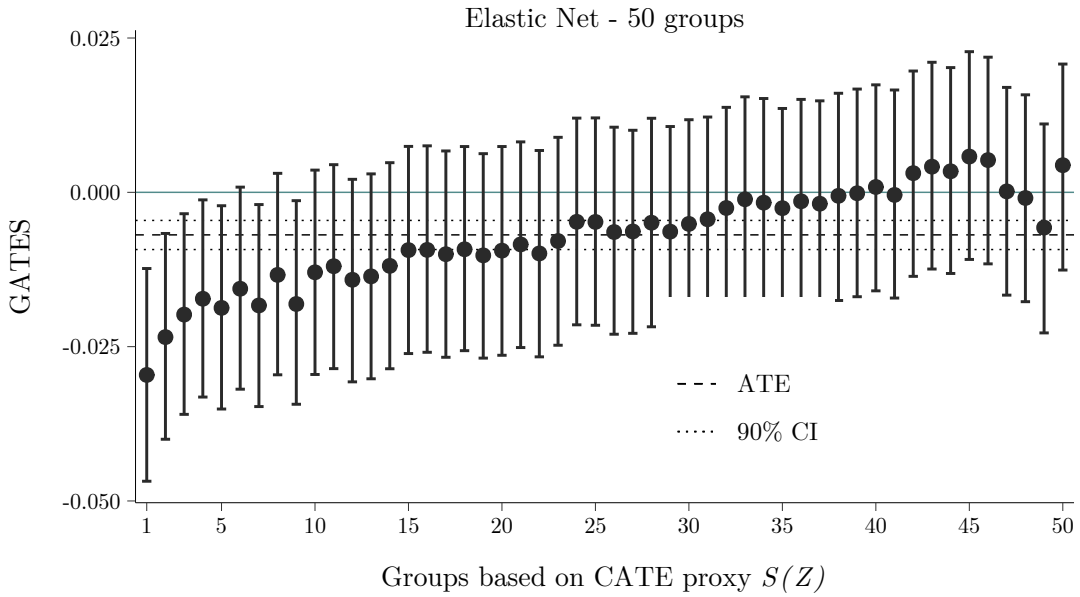
	Elastic Net		
	Least affected (γ_{50})	Most affected (γ_1)	Difference ($\gamma_{10} - \gamma_1$)
Coefficients	0.004	-0.030	0.035
Confidence bands	(-0.013 ; 0.021)	(-0.047 ; -0.012)	-
Adjusted p-values	[0.199]	[0.002]	[0.006]

Notes. The reported results are medians over 100 splits. The CIs are at the 90 % significance level. Adjusted p-values are from t-tests with H_0: estimated coefficient is equal to zero. Since a negative effect of wind turbines on house prices is assumed, the most affected group is the group with the largest negative effect proxy, i.e. the first decile.

-3.0 %, and significant at any conventional significance level as well. This also yields a large difference of treatment effects between the most and least affected group of 3.5 percentage points. The results depicted in Table 12 thus at least point towards the above stated hypothesis that the negative treatment effect of -2.0 % in the most affected decile of properties is driven by a part of more severely affected houses.

Figure 12 provides further insight into the composition of the group average treatment effects. In the same way as in Section 6.1, the figure shows the estimated GATES coefficients for all 50 groups and the accompanying 90 % CIs as well as the results from the BLP estimation. In line with the results using 10 groups, the specification with 50 groups does not find significant positive treatment effects in any of the groups. Moreover, it is notable that the estimated CIs are much wider compared to the CIs from Section 6.1. This is most probably due to the much lower number of observations in each group, which does not allow to estimate the GATES coefficients as precisely as in the approach with 10 groups only. Furthermore, the plot reveals the hypothesized pattern, albeit not as pronounced as expected: the estimated effect in the most affected 2 % of the properties is the largest in absolute size, however there is no clear gap or jump from the first to the second group. Therefore, while the group average treatment effect in the first decile still differs between properties within this group, there is no small number of houses which clearly drive this effect, while the remaining houses are affected to a much lesser extent.

Since the less affected groups are fairly small, but do not differ much in terms of their estimated treatment effects, there is a risk of accidentally comparing the most affected group to a particularly selected group, which might not be represen-



Notes. The dots represent the point estimates for the ATEs in the respective groups. Errorbars denote 90 % CIs. Point estimates and CIs are medians over 100 splits.

Figure 12: GATES results - 50 groups

tative for most of the unaffected properties, when conducting the classification analysis. Therefore, during the CLAN, the most affected group is not simply compared to the least affected group, but rather to the least affected 50 % of the sample. This is a reasonable approach since the GATES estimation does not reveal any significant effects for the least affected 50 % of the analyzed properties, neither did the main specification in Section 6.1 nor the robustness check conducted here. Table 13 displays the results from the respective classification analysis.

The results are qualitatively very similar to the results obtained in Section 6.1, but much more pronounced. The average year of construction is again substantially lower in the most affected group, even compared to the 10 % most affected properties, and amounts to 1917.9. 7 % of the properties are protected historic buildings, which constitutes a very large fraction compared to the sample mean of only 1 %, and which is also much higher than in the first decile. The same holds true for the number of inhabitants, increasing from 2,596 to 3,833, also in line with the shares of housing blocks (12.0 %), of semidetached houses (4 %), multi-family houses (33 %), and even skyscrapers (4.8 %).

All in all, the obtained results do not clearly provide evidence in favor of the hypothesis, that the negative treatment effects are largely driven by a small fraction of the overall sample. While the most affected 2 % of the properties are

Table 13: CLAN results – 50 groups

Variable	Elastic Net	
	2 % Most	50 % Least
Asking price	12.7	12.6
Year of construction	1917.9	1986.1
Protected building (0/1)	0.07	0.01
Purch. power / capita*	23,762	22,648
Unemployment rate (%)*	6.6	5.1
# Inhabitants*	3,833	1,711
Skyscrapers (%)*	4.8	3.0
Housing blocks (%)*	12.0	6.9
Age: 0-25 (%)*	24.2	24.6
Age: 25-40 (%)*	18.8	16.8
Age: 40-65 (%)*	36.2	37.7
Age: 65+ (%)*	20.6	21.0
Farmhouse (0/1)	0.01	0.01
Bungalow (0/1)	0.01	0.04
Semidetached house (0/1)	0.04	0.19
Single-family house (0/1)	0.28	0.50
Multi-family house (0/1)	0.35	0.08
Terraced house (0/1)	0.11	0.11
Villa (0/1)	0.08	0.03
Category: Other (0/1)	0.10	0.04
Property condition	6.2	4.3

Notes. The reported results are medians over 100 splits. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.

more strongly affected with an estimated ATE of -3.0 %, there is no clear cut or gap between the effects for this group and the preceding groups. Thus, while there appears to be some heterogeneity in treatment effects between properties within the most affected decile, it is not as distinct as one would expect given the just stated hypothesis. However, the results from the classification analysis are consistent with the results from the main specification, by which they add to the robustness and credibility of the results.

One potential limitation of this thesis and the utilized empirical strategy is that it is difficult to clearly divide the sample into a treatment and a control group.

Assuming that the estimated negative impacts of wind turbines on the properties' asking prices are most likely and mainly due to visibility effects (Gibbons, 2015), the distance to the nearest wind turbine can at most serve as a proxy for this visibility. Therefore, depending on the choice of the treatment distance threshold, properties without wind turbines in sight might unintentionally end up in the treatment group, and vice versa. As a brief reminder: in the first step of the exploited estimation strategy, separate models are fitted to both the treatment and the control group. These models are then used to predict (hypothetical) outcomes under treatment and no treatment for all observations in the main sample, and the difference between these predictions is further used in the weighted linear regressions as the treatment effect proxy. If a substantial number of properties in the treatment group would actually not be treated, this would result in upward shifted predictions for the treatment group model. The difference between predictions would then be underestimated, and hence the CATE proxy, and with it the treatment effects, would be biased downward. The second robustness check therefore concerns the choice of the maximum treatment threshold. As discussed in Section 5, based on findings from the literature on the effects of wind turbines on house prices, a maximum distance of 5 km to the nearest wind turbine is chosen for houses to be assigned to the treatment group in the main specification. In addition to that, the estimation procedure utilizing the elastic net algorithm is repeated with a reduced treatment threshold of 3 km.

Table 14: BLP estimation results – 3 km threshold

	Elastic Net	
	ATE (β_1)	HET (β_2)
Coefficients	-0.010	0.064
Confidence bands	(-0.014 ; -0.007)	(-0.065 ; 0.179)
Adjusted p-values	[0.000]	[0.001]

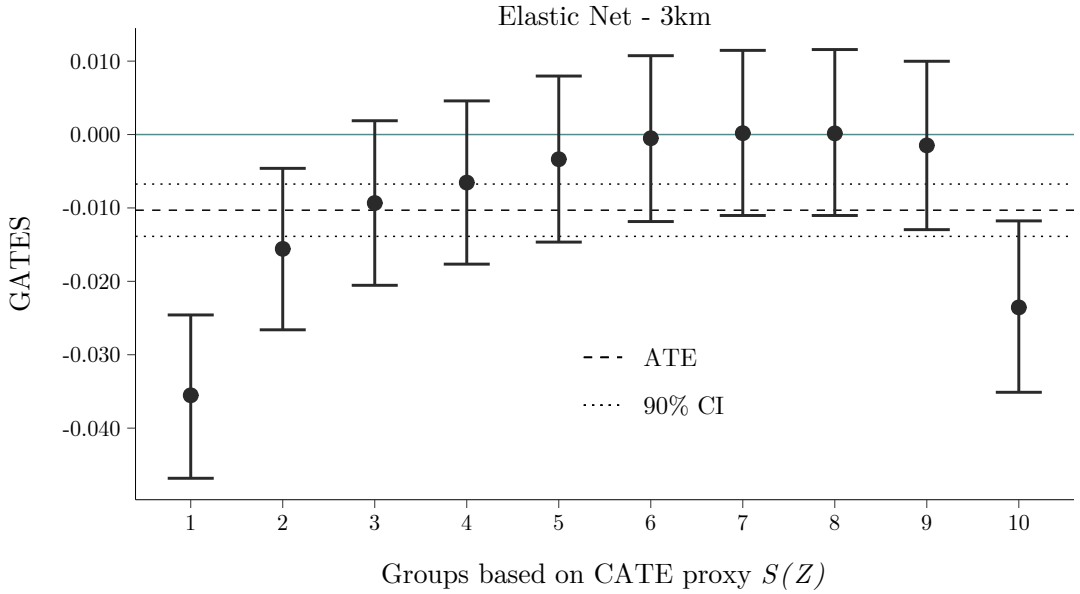
Notes. The reported results are medians over 100 splits. The CIs are at the 90 % significance level. Adjusted p-values are from t-tests with H_0: estimated coefficient is equal to zero.

Using this threshold reduces the number of treated properties from 313,460 (36.7 %) to 150,858 (17.9 %).[36] The results from the BLP estimation are reported in Table 14. While the estimated ATE in the main specification above was -0.7 %, the ATE for a range of 3 km is slightly higher and amounts to -1.0 %. This is as

[36] Using the same propensity score trimming thresholds of below 0.05 and above 0.95 would result in the exclusion of 25.3 % of the main sample on average. Therefore, the trimming thresholds are adjusted to propensity scores of below 0.01 and 0.99 only. This removes on average 5.0 % of the properties from the estimation samples.

expected, assuming both that a closer distance as a threshold reduces the risk of non-treated houses ending up in the treatment group, and that treatment effects increase with a smaller distance between properties and wind turbines, as for example found by Frondel et al. (2019) or Gibbons (2015). The HET coefficient β_2 however decreased drastically from 0.328 to only 0.064. Furthermore, the confidence interval and the adjusted p-value reveal ambiguous findings. While the former is rather wide and includes zero, the p-value in contrast indicates that the estimated parameter is significantly different from zero. This difference is due to the way the CIs and the p-values are calculated. A closer look into the results for each single data split reveals that the estimated β_2 coefficients differ by a large extent, ranging from -0.916 to 0.691. Since they are roughly symmetrically spread around zero, the median over all these splits is close to zero as well, as also indicated by the reported parameter in Table 14. The same holds true for the upper and lower bounds of the CIs. However, since a large fraction of the estimated negative coefficients are far from zero and significant, while the same is true for a large fraction of the estimated positive coefficients, this leads to many very small p-values. In fact, 39 out of 44 negatively estimated coefficients are different from zero at the 10 % significance level, while this is the case for 43 out of the remaining 56 positive HET coefficients. Therefore, the median over all the p-values, and the adjusted p-values are very small as well. Hence, the p-value in this specific case is misleading and the CI is more accurate and trustworthy, and thus the null hypothesis that β_2 is equal to zero, cannot be rejected. Referring back to Section 2.2, testing this hypothesis corresponds to testing whether there is heterogeneity in treatment effects, and whether this heterogeneity can be predicted by the ML algorithm used to construct the proxy $S(Z)$ at the same time. Therefore, the wide CI implies that either there are no HTEs on properties in the range of maximum 3 km to the nearest wind turbine, or that the heterogeneity is simply not captured by the elastic net model.

Figure 13 plots the estimated GATES coefficients and the corresponding 90 % CIs. The average treatment effect in the most affected group amounts to -3.5 %, and thus is larger compared to the ATE in the first decile using the 5 km threshold (-2.0 %). The remaining GATES parameters up to the least affected decile are very close to the estimated coefficients from Figure 11. However, in contrast to all the results obtained until now, the ATE in the last group becomes negative again and is also statistically significant. This result is not as expected, since this group constitutes the decile, which contains the properties with the largest proxy treatment effects $S(Z)$. Given the monotonicity assumption from Section 2.3, the GATES should increase with the groups constructed from the proxy as well.



Notes. The dots represent the point estimates for the ATEs in the respective groups. Errorbars denote 90 % CIs. Point estimates and CIs are medians over 100 splits.

Figure 13: GATES results – 3 km treatment threshold

Thus, the results obtained here constitute a clear violation of this assumption. Chernozhukov et al. (2018) further note that monotonicity holds if the proxy $S(Z)$ is consistent for the true CATE function $s_0(Z)$. Thus, the violation of this assumption can be seen as evidence that the elastic net algorithm in fact yields a poor approximation to the CATE when a treatment threshold of 3 km instead of 5 km is used. Therefore, it seems reasonable to conclude that the insignificant β_2 coefficient is more likely to result from a poor approximation, than to reject the hypothesis of the existence of treatment effect heterogeneity.

7 Conclusion

Throughout the introduction of this master thesis it became clear that Germany has already made great efforts to reduce its greenhouse gas emissions in order to mitigate climate change and to limit the ongoing global warming. The largest reductions in GHG emissions could be achieved in the sector of electricity generation via an extensive promotion of renewable energies. In the most recent years, however, the capacity expansion of energy from renewables has slowed down. This is especially caused by a drastic decline in the number of new wind turbines. The construction of new wind power plants is often accompanied by severe local protest and lawsuits due to perceived negative externalities. Empirical research aimed at quantifying these negative externalities utilizing hedonic price models. By estimating the impact of wind turbines on nearby properties' prices, most of

these studies find negative effects. Up to now, there exist only a single study which investigates heterogeneous treatment effects in this context. The authors find stronger effects on property values for houses in rural areas and for properties built before 1950. However, the literature still lacks a systematic, data-driven evaluation of potential heterogeneity. Therefore, this thesis provides additional details and insights into such heterogeneous effects.

In order to systematically analyze the effects of wind turbines on prices of nearby properties, a particularly suitable empirical strategy, developed by Chernozhukov et al. (2018), was explained in Section 2 of this thesis. The described approach is data-driven in the sense that (non-parametric) machine learning algorithms are used to flexibly model the relationship between observed covariates and the outcome, without prespecifying or restricting the functional form of the fitted models. The strategy is described as "agnostic", since it does not impose any assumptions on the ML models, which are only difficult or even impossible to verify. Specifically, the estimation approach does not even assume consistency or unbiasedness of the estimators. In a first step, a proxy of the treatment effects is constructed, which is then postprocessed to obtain average treatment effects. Moreover, the method also identifies the most and least affected groups, the ATEs in those groups, and the characteristics of the corresponding observations. The approach even comes with statistically valid confidence intervals and adjusted p-values, and thus can be used to make inferential statements about the obtained results.

Since this empirical strategy heavily relies on ML algorithms, Section 3 provided an overview of the main ideas and techniques of this field and introduced and explained the algorithms utilized in this thesis. First, the elastic net algorithm is chosen from the class of regularized linear models. Furthermore, Breiman's random forests (2001) and the recently developed XGBoost algorithm (Chen and Guestrin, 2016) are used as more sophisticated techniques, both building on decision trees as base learners. Lastly, single hidden-layer feedforward neural networks are adopted to construct the treatment effect proxy.

Section 4 described the datasets, this thesis' empirical analysis draws upon, in more detail. They comprise information on house prices and properties' characteristics, on locality characteristics on a one square kilometer grid, and on wind turbines in Germany. Furthermore, this data is used to identify the nearest wind turbines for all houses in the sample as well as to calculate the distance between both. Moreover, necessary data cleaning and preprocessing steps were depicted.

Closely connected to the data, Section 5 discussed various implementation details, which have to be taken into consideration when applying the previously illustrated estimation strategy in this work's specific context. The use of random search in combination with repeated cross-validation to tune the algorithms' hyperparameters is justified from a theoretical and practical point of view. The empirical approach was originally developed for randomized controlled experiments with one treated and one control group, and hence requires a binary treatment indicator. Since only the distances between properties and the respective nearest wind power plants are known, a maximum distance for houses to be regarded as treated had to be defined. Relying on findings from the literature, a threshold of 5 km was chosen. Moreover, in RCTs the propensity scores are known by the design of the experiment. Since the data used in this paper is observational data, the propensity scores have to be estimated. An experiment comparing logistic regression, random forests and XGBoost and their ability to produce propensity score estimates, which balance the treatment and the control group, was conducted. Logistic regression resulted in the most balanced samples after weighting with the estimated scores, as indicated by multiple balance diagnostics, and was therefore chosen for the propensity score estimations. Lastly, the inclusion of municipality-level fixed effects in order to control for possible unobserved factors, which might influence the placing of wind turbines and property prices at the same time, was motivated. The within-transformation approach and its advantages over the use of dummy variables for each group level in this specific context was further discussed.

Section 6 then presented and discussed the obtained empirical results. Utilizing criteria provided by Chernozhukov et al. (2018) to assess the performance of the employed ML algorithms, the elastic net algorithm yielded the best approximation to the actual conditional average treatment effect function. While no evidence was found that the prices might increase, when a wind turbine is located in a range of 5 km to the properties, all algorithms identified significant negative treatment effects. Building on the preferred specification, negative treatment effects of up to -2.0 % for the most affected group were estimated. Further analyses revealed even slightly higher effects of -3.0 % for the 2 % most affected properties in the sample. Moreover, all estimations consistently provided evidence in favor of the existence of heterogeneous treatment effects. The classification analysis comparing characteristics of the most and least affected properties revealed interesting differences between groups: properties, which are located in more densely populated areas, are affected more strongly. The same holds true for much older houses, protected historic buildings and properties in a better condition. These

results were consistent across algorithms and the number of groups in the GATES estimation as well.

One potential limitation of this thesis and the utilized approach lies in the identification of the treatment and the control group. Assuming that negative impacts of wind turbines on house prices are mostly due to visibility effects (Gibbons, 2015), the employed distances between properties and the nearest wind power plants in this thesis can only serve as a proxy for this visibility. Dröes and Koster (2016) argue that people often visit multiple different locations in their neighborhood, and thus see the wind turbines somewhat regularly, even if the plants cannot be seen directly from their houses. However, the robustness check in Section 6.2, using an alternative treatment threshold of 3 km, illustrated that, although the results were partly as expected, i.e. larger ATEs were found, the exact choice of the treatment distance actually does seem to be important in this case, as indicated by the much worse CATE approximation.

All in all, while some of the results obtained here are in line with the only other study on treatment effect heterogeneity of wind turbines on house prices, as for example that older houses are much more affected, others are not, and even contradict each other, as the conflicting results on which houses are more strongly affected, either those in rural areas or in more densely populated areas. Therefore, further research is needed to provide additional insights into how treatment effects vary between properties. Due to the above stated reasons, such studies would likely benefit from clearly identified visibility indicators of wind turbines instead of utilizing distances between properties and wind turbines. The results from those studies could help to further enhance the understanding of local citizens' rejection of new wind power plants and could be used to adequately and monetarily compensate affected citizens in order to reduce local resistance and promote acceptance, and hence the further expansion of wind turbines and renewable energies in general.

References

Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation* 9(7): 1545–1588. doi:10.1162/neco.1997.9.7.1545.

Anastasiadis, A.D., Magoulas, G.D., and Vrahatis, M.N. (2005). New globally convergent training scheme based on the resilient propagation algorithm. *Neurocomputing* 64: 253–270. doi:10.1016/j.neucom.2004.11.016.

Athey, S. (2018). The impact of machine learning on economics. In A. Agrawal, J. Gans, and A. Goldfarb (editors), *The economics of artificial intelligence: An agenda*, pages 507–547. University of Chicago Press. doi:10.7208/chicago/9780226613475.003.0021.

Athey, S. and Imbens, G.W. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences of the United States of America* 113(27): 7353. doi:10.1073/pnas.1510489113.

Athey, S. and Imbens, G.W. (2019). Machine Learning Methods That Economists Should Know About. *Annual Review of Economics* 11(1): 685–725. doi:10.1146/annurev-economics-080217-053433.

Austin, P.C. (2011). An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. *Multivariate Behavioral Research* 46(3): 399–424. doi:10.1080/00273171.2011.568786.

Austin, P.C. and Stuart, E.A. (2015). Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. *Statistics in Medicine* 34(28): 3661–3679. doi:10.1002/sim.6607.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research* 13(1): 281–305. doi:10.5555/2188385.2188395.

BMWi (2019). Zeitreihen zur Entwicklung der erneuerbaren Energien in Deutschland. Link: https://www.erneuerbare-energien.de/EE/Redaktion/DE/Downloads/zeitreihen-zur-entwicklung-der-erneuerbaren-energien-in-deutschland-1990-2018.pdf?__blob=publicationFile&v=24. Accessed: 22.10.2020.

BMWi (2020). Erneuerbare Energien. Link: https://www.bmwi.de/Redaktion/DE/Dossier/erneuerbare-energien.html. Accessed: 22.10.2020.

Boelmann, B., Budde, R., Klick, L., Schaffner, S., RWI, and Scout24 AG c/o Immobilien Scout GmbH (2019). RWI-GEO-RED: RWI Real Estate Data (Scientific Use File) - apartments for sale. Version: 1. Essen: RWI – Leibniz Institute for Economic Research. Dataset. doi:10.7807/immo:red:wk:suf:v1.

Boelmann, B. and Schaffner, S. (2019). FDZ Data description: Real-Estate Data for Germany (RWI-GEO-RED) - Advertisements on the Internet Platform ImmobilienScout24. *RWI Projektberichte.* Essen: RWI - Leibniz Institute for Economic Research.

Breidenbach, P. and Eilers, L. (2018). RWI-GEO-GRID: Socio-economic data on grid level. *Jahrbücher für Nationalökonomie und Statistik* 238(6): 609–616. doi:10.1515/jbnst-2017-0171.

Breiman, L. (1996). Bagging predictors. *Machine learning* 24(2): 123–140. doi:10.1007/BF00058655.

Breiman, L. (2001). Random forests. *Machine learning* 45(1): 5–32. doi:10.1023/A:1010933404324.

Breiman, L., Friedman, J., Stone, C.J., and Olshen, R.A. (1984). *Classification and regression trees.* CRC press.

Broyden, C.G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics* 6(1): 76–90. doi:10.1093/imamat/6.1.76.

Bundesnetzagentur (2020a). Aktuelle Einheitenübersicht. Link: https://www.marktstammdatenregister.de/MaStR/Einheit/Einheiten/OeffentlicheEinheitenuebersicht. Accessed: 22.10.2020.

Bundesnetzagentur (2020b). Marktstammdatenregister. Link: https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_Institutionen/DatenaustauschundMonitoring/Marktstammdatenregister/MaStR_node.html. Accessed: 22.10.2020.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. Association for Computing Machinery.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y.,

and Li, Y. (2020). *xgboost: Extreme Gradient Boosting*. R package version 1.0.0.2.

Chernozhukov, V., Demirer, M., Duflo, E., and Fernández-Val, I. (2018). Generic Machine Learning Inference on Heterogenous Treatment Effects in Randomized Experiments. *National Bureau of Economic Research Working Paper Series* 24678. doi:10.3386/w24678.

Chernozhukov, V., Demirer, M., Duflo, E., and Fernández-Val, I. (2017). Generic machine learning inference on heterogenous treatment effects in randomized experiments. v2. arXiv: 1712.04802 [stat.ML].

Deryugina, T., Heutel, G., Miller, N.H., Molitor, D., and Reif, J. (2019). The Mortality and Medical Costs of Air Pollution: Evidence from Changes in Wind Direction. *American Economic Review* 109(12): 4178–4219. doi: 10.1257/aer.20180279.

Deutsche WindGuard (2019). Status des Windenergieausbaus an Land in Deutschland - Jahr 2018. Varel: Deutsche WindGuard GmbH.

Deutsche WindGuard (2020). Status des Windenergieausbaus an Land in Deutschland - Jahr 2019. Varel: Deutsche WindGuard GmbH.

Dröes, M.I. and Koster, H.R.A. (2016). Renewable energy and negative externalities: The effect of wind turbines on house prices. *Journal of Urban Economics* 96: 121–141. doi:10.1016/j.jue.2016.09.001.

Efron, B. and Tibshirani, R.J. (1994). *An introduction to the bootstrap*. CRC press.

FA Wind (2019). Umfrage zur Akzeptanz der Windenergie an Land. Berlin: Fachagentur Windenergie an Land e.V.

Fletcher, R. (1970). A new approach to variable metric algorithms. *The computer journal* 13(3): 317–322. doi:10.1093/comjnl/13.3.317.

Foster, J.C., Taylor, J.M.G., and Ruberg, S.J. (2011). Subgroup identification from randomized clinical trial data. *Statistics in Medicine* 30(24): 2867–2880. doi:10.1002/sim.4322.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software* 33(1): 1–22. doi:10.18637/jss.v033.i01.

Friedman, J., Hastie, T., Tibshirani, R. et al. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* 28(2): 337–407. doi:10.1214/aos/1016218223.

Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics* pages 1189–1232. doi:10.1214/aos/1013203451.

Friedman, J.H. (2002). Stochastic gradient boosting. *Computational statistics & data analysis* 38(4): 367–378. doi:10.1016/S0167-9473(01)00065-2.

Frondel, M., Kussel, G., Sommer, S., and Vance, C. (2019). Local cost for global benefit: The case of wind turbines. *Ruhr Economic Papers* 791. doi:10.4419/86788919.

Gibbons, S. (2015). Gone with the wind: Valuing the visual impacts of wind turbines through house prices. *Journal of Environmental Economics and Management* 72: 177–196. doi:10.1016/j.jeem.2015.04.006.

Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of computation* 24(109): 23–26. doi:10.2307/2004873.

Gormley, T.A. and Matsa, D.A. (2014). Common errors: How to (and not to) control for unobserved heterogeneity. *The Review of Financial Studies* 27(2): 617–661. doi:10.1093/rfs/hht047.

Greifer, N. (2020). *cobalt: Covariate Balance Tables and Plots*. R package version 4.2.3.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hoen, B., Wiser, R., Cappers, P., Thayer, M., and Sethi, G. (2011). Wind Energy Facilities and Residential Properties: The Effect of Proximity and View on Sales Prices. *Journal of Real Estate Research* 33(3): 279–316. doi:10.1080/10835547.2011.12091307.

Imai, K. and Ratkovic, M. (2013). Estimating treatment effect heterogeneity in randomized program evaluation. *Annals of Applied Statistics* 7(1): 443–470. doi:10.1214/12-AOAS593.

Imbens, G.W. and Wooldridge, J.M. (2009). Recent developments in the econometrics of program evaluation. *Journal of economic literature* 47(1): 5–86. doi:10.1257/jel.47.1.5.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*. Springer.

Jensen, C.U., Panduro, T.E., and Lundhede, T.H. (2014). The Vindication of Don Quixote: The Impact of Noise and Visual Pollution from Wind Turbines. *Land Economics* 90(4): 668–682. doi:10.3368/le.90.4.668.

Kampstra, P. (2008). Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of statistical software* 28(1): 1–9. doi:10.18637/jss.v028.c01.

Kim, J.H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis* 53(11): 3735–3745. doi:10.1016/j.csda.2009.04.009.

Kosfeld, R. and Werner, A. (2012). Deutsche Arbeitsmarktregionen – Neuabgrenzung nach den Kreisgebietsreformen 2007–2011. *Raumforschung und Raumordnung Spatial Research and Planning* 70(1): 49–64. doi:10.1007/s13147-011-0137-8.

Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software* 28(1): 1–26. doi:10.18637/jss.v028.i05.

Kuhn, M. (2014). Futility analysis in the cross-validation of machine learning models.

Kuhn, M. and Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. CRC Press.

Künzel, S.R., Sekhon, J.S., Bickel, P.J., and Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences of the United States of America* 116(10): 4156–4165. doi:10.1073/pnas.1804597116.

Lang, C., Opaluch, J.J., and Sfinarolakis, G. (2014). The windy city: Property value impacts of wind turbines in an urban setting. *Energy Economics* 44: 413–421. doi:10.1016/j.eneco.2014.05.010.

LeCun, Y.A., Bottou, L., Orr, G.B., and Müller, K.R. (2012). Efficient BackProp. In G. Montavon, G.B. Orr, and K.R. Müller (editors), *Neural Networks: Tricks of the Trade: Second Edition*, pages 9–48. Springer.

Lee, B.K., Lessler, J., and Stuart, E.A. (2010). Improving propensity score weighting using machine learning. *Statistics in Medicine* 29(3): 337–346. doi:10.1002/sim.3782.

McCaffrey, D.F., Ridgeway, G., and Morral, A.R. (2004). Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods* 9(4): 403–425. doi:10.1037/1082-989X.9.4.403.

Mullainathan, S. and Spiess, J. (2017). Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives* 31(2): 87–106. doi:10.1257/jep.31.2.87.

Nie, X. and Wager, S. (2017). Quasi-oracle estimation of heterogeneous treatment effects. arXiv:1712.04912 [stat.ML].

Nielsen, M.A. (2015). *Neural networks and deep learning*, volume 2018. Determination press.

Probst, P. and Boulesteix, A.L. (2017). To tune or not to tune the number of trees in random forest. *Journal of Machine Learning Research* 18(1): 6673–6690. doi:10.5555/3122009.3242038.

Probst, P., Boulesteix, A.L., and Bischl, B. (2019a). Tunability: Importance of Hyperparameters of Machine Learning Algorithms. *Journal of Machine Learning Research* 20(53): 1–32.

Probst, P., Wright, M.N., and Boulesteix, A.L. (2019b). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery* 9(3): e1301. doi:10.1002/widm.1301.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. Vienna: R Foundation for Statistical Computing.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE international conference on neural networks*, pages 586–591. doi:10.1109/ICNN.1993.298623.

Rubin, D.B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66(5): 688–701. doi:10.1037/h0037350.

RWI and microm (2019). Rwi-geo-grid: Socio-economic data on grid level (wave 8). version: 1. Essen: RWI – Leibniz Institute for Economic Research. Dataset. doi:10.7807/microm:V8.

Shanno, D.F. (1970). Conditioning of quasi-newton methods for function minimization. *Mathematics of computation* 24(111): 647–656. doi:10.2307/2004840.

Shen, H., Welch, W.J., and Hughes-Oliver, J.M. (2011). Efficient, adaptive cross-validation for tuning and comparing models, with application to drug discovery. *Annals of Applied Statistics* 5(4): 2668–2687. doi:10.1214/11-AOAS491.

Snoek, J., Larochelle, H., and Adams, R.P. (2012). Practical bayesian optimization of machine learning algorithms. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger (editors), *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc.

Stuart, E.A., Lee, B.K., and Leacy, F.P. (2013). Prognostic score–based balance measures can be a useful diagnostic for propensity score methods in comparative effectiveness research. *Journal of Clinical Epidemiology* 66(8, Supplement): S84–S90.e1. doi:10.1016/j.jclinepi.2013.01.013.

Sunak, Y. and Madlener, R. (2016). The impact of wind farm visibility on property values: A spatial difference-in-differences analysis. *Energy Economics* 55: 79–91. doi:10.1016/j.eneco.2015.12.025.

Sunak, Y. and Madlener, R. (2017). The impact of wind farms on property values: A locally weighted hedonic pricing model. *Papers in Regional Science* 96(2): 423–444. doi:10.1111/pirs.12197.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1): 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x.

Umweltbundesamt (2019). Treibhausgas-Emissionen in Deutschland. Link: https://www.umweltbundesamt.de/daten/klima/treibhausgas-emissionen-in-deutschland#emissionsentwicklung-1990-bis-2017. Accessed: 22.10.2020.

Umweltbundesamt (2020). Kohlendioxid-Emissionen. Link: https://www.umweltbundesamt.de/daten/klima/treibhausgas-emissionen-in-deutschland/kohlendioxid-emissionen#kohlendioxid-emissionen-im-vergleich-zu-anderen-treibhausgasen. Accessed: 22.10.2020.

Varian, H.R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives* 28(2): 3–28. doi:10.1257/jep.28.2.3.

Venables, W.N. and Ripley, B.D. (2002). *Modern Applied Statistics with S*. Springer.

Venables, W.N. and Ripley, B.D. (2013). *Modern applied statistics with S-PLUS*. Springer Science & Business Media.

Wager, S. and Athey, S. (2018). Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *Journal of the American Statistical Association* 113(523): 1228–1242. doi:10.1080/01621459.2017.1319839.

Wainer, J. and Cawley, G. (2017). Empirical evaluation of resampling procedures for optimising SVM hyperparameters. *The Journal of Machine Learning Research* 18(1): 475–509. doi:10.5555/3122009.3122024.

Wooldridge, J.M. (2016). *Introductory econometrics: A modern approach*. Nelson Education.

Wright, M.N. and Ziegler, A. (2017). ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *Journal of Statistical Software* 77(1): 1–17. doi:10.18637/jss.v077.i01.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* 67(2): 301–320. doi:10.1111/j.1467-9868.2005.00503.x.

Zou, H. and Hastie, T. (2020). *elasticnet: Elastic-Net for Sparse Estimation and Sparse PCA*. R package version 1.3.

Appendix

A.1 Proof of BLP Estimation Strategy*

Recall the weighted linear regression equation from Section 2.2 given by:

$$Y = \alpha' X_1 + \beta_1\big(D - p(Z)\big) + \beta_2\big(D - p(Z)\big)\big(S(Z) - \mathbf{E}[S(Z)]\big) + \epsilon, \tag{A1.1}$$

$$\text{with weights } w(Z) = \frac{1}{p(Z)\big(1 - p(Z)\big)}.$$

The minimization problem from this weighted regression yields the following normal equations:

$$\mathbf{E}\big[w(Z)(Y - \alpha' X_1 - \beta' X_2)X_2\big] = 0 \tag{A1.2}$$

Define:

$$X_1 := X_1(Z), \text{ e.g. } \big[1, B(Z)\big]' \tag{A1.3}$$

$$X_2 := \big[\{D - p(Z)\}, \{(D - p(Z))(S - \mathbf{E}[S])\}\big]' \tag{A1.4}$$

$$S := S(Z) \tag{A1.5}$$

Since $Y = b_0(Z) + s_0(Z)D + U$, Y can be replaced in Equation (A1.2):

$$\mathbf{E}\big[w(Z)\big(b_0(Z) + s_0(Z)D + U - \alpha' X_1 - \beta' X_2\big)X_2\big] = 0 \tag{A1.6}$$

Using the linearity property of expectations, this equation can be decomposed into the following terms:

$$\mathbf{E}\big[w(Z)b_0(Z)X_2\big] \tag{A1.7a}$$

$$\mathbf{E}\big[w(Z)s_0(Z)DX_2\big] \tag{A1.7b}$$

$$\mathbf{E}\big[w(Z)UX_2\big] \tag{A1.7c}$$

$$\mathbf{E}\big[w(Z)\alpha' X_1 X_2\big] \tag{A1.7d}$$

$$\mathbf{E}\big[w(Z)\beta' X_2 X_2\big] \tag{A1.7e}$$

By the law of iterated expectations (LIE), Equation (A1.7a) can also be written as:

$$\mathbf{E}\big[w(Z)b_0(Z)X_2\big] = \mathbf{E}\big[w(Z)b_0(Z)\mathbf{E}[X_2 \mid Z]\big] \tag{A1.8}$$

* The following proof is taken from the project studies thesis as submitted on April 04, 2020.

Utilizing that the treatment variable D is a binary variable with $D \in \{0, 1\}$ and that the propensity scores $p(Z)$ are the probability of treatment conditional on the covariates, i.e. $\mathbf{E}[D \mid Z] = Pr(D = 1 \mid Z) = p(Z)$, the expectation of both elements of X_2 is zero:

$$\begin{aligned}\mathbf{E}\big[D - p(Z) \mid Z\big] &= \mathbf{E}\big[D \mid Z\big] - \mathbf{E}\big[p(Z) \mid Z\big] && \text{(A1.9)}\\ &= p(Z) - p(Z) && \text{(A1.10)}\\ &= 0 && \text{(A1.11)}\end{aligned}$$

Therefore Equation (A1.7a) and Equation (A1.7d) are equal to zero:

$$\mathbf{E}\big[w(Z)b_0(Z)X_2\big] = \mathbf{E}\big[w(Z)b_0(Z)\underbrace{\mathbf{E}[X_2 \mid Z]}_{=0}\big] = 0 \qquad \text{(A1.12)}$$

$$\mathbf{E}\big[w(Z)\alpha' X_1 X_2\big] = \mathbf{E}\big[w(Z)\alpha' X_1 \underbrace{\mathbf{E}[X_2 \mid Z]}_{=0}\big] = 0 \qquad \text{(A1.13)}$$

Since U is the residual from a regression of Y on Z, D, its expectation conditional on Z, D is zero by construction and therefore Equation (A1.7c) is zero as well:

$$\mathbf{E}\big[w(Z)UX_2\big] = \mathbf{E}\big[w(Z)\underbrace{\mathbf{E}[U \mid Z, D]}_{=0}X_2\big] = 0 \qquad \text{(A1.14)}$$

Thus, the normal equations from Equation (A1.6) simplify to:

$$\mathbf{E}\big[w(Z)\big(s_0(Z)D - \beta' X_2\big)X_2\big] = 0 \qquad \text{(A1.15)}$$

Further, the following relation will be used:

$$\begin{aligned}\mathbf{E}\big[\big(D - p(Z)\big)\big(D - p(Z)\big) \mid Z\big] &= \mathbf{E}\big[(D - p(Z))^2 \mid Z\big] && \text{(A1.16)}\\ &= \mathbf{Var}\big[D - p(Z) \mid Z\big] - \underbrace{\mathbf{E}\big[D - p(Z) \mid Z\big]}_{=0} && \text{(A1.17)}\end{aligned}$$

Since $p(Z) \mid Z$ is a constant, $\mathbf{Var}[D - p(Z) \mid Z]$ simplifies to $\mathbf{Var}[D \mid Z]$. Utilizing that $D \mid Z \sim \mathrm{B}\big(1, p(Z)\big)$, its variance is given by $\mathbf{Var}[D \mid Z] = p(Z)(1 - p(Z))$, which corresponds to the inverse of the weights $w(Z)$ as defined in Section 2.2.

$$\begin{aligned}&= \mathbf{Var}\big[D \mid Z\big] && \text{(A1.18)}\\ &= p(Z)\big(1 - p(Z)\big) && \text{(A1.19)}\\ &= w^{-1}(Z) && \text{(A1.20)}\end{aligned}$$

This relation is used to show that both components of X_2 are orthogonal under the weights $w(Z)$. As a reminder, $X_2 := [\{D - p(Z)\}, \{(D - p(Z))(S - \mathbf{E}[S])\}]'$. Using the results from Equations (A1.16) to (A1.20), orthogonality can be shown:

$$\mathbf{E}\big[w(Z)(D - p(Z))(D - p(Z))(S - \mathbf{E}[S])\big] \tag{A1.21}$$
$$= \mathbf{E}\big[w(Z)w^{-1}(Z)(S - \mathbf{E}[S])\big] \tag{A1.22}$$
$$= \mathbf{E}\big[S - \mathbf{E}[S]\big] \tag{A1.23}$$
$$= 0 \tag{A1.24}$$

Utilizing orthogonality of components of X_2 then leads to the following system of equations:

$$\mathbf{E}\big[w(Z)\{s_0(Z)D - \beta_1(D - p(Z))\}(D - p(Z))\big] = 0 \tag{A1.25}$$
$$\mathbf{E}\big[w(Z)\{s_0(Z)D - \beta_2(D - p(Z))(S - \mathbf{E}[S])\}(D - p(Z))(S - \mathbf{E}[S])\big] = 0 \tag{A1.26}$$

Solving Equation (A1.25) for β_1 yields:

$$\beta_1 = \frac{\mathbf{E}\big[w(Z)s_0(Z)D(D - p(Z))\big]}{\mathbf{E}\big[w(Z)(D - p(Z))(D - p(Z))\big]} \tag{A1.27}$$

Using the LIE in the denominator and applying the just derived relationship leads to:

$$\beta_1 = \frac{\mathbf{E}\big[w(Z)s_0(Z)D(D - p(Z))\big]}{\mathbf{E}\big[w(Z)\mathbf{E}\big[(D - p(Z))(D - p(Z)) \mid Z\big]\big]} \tag{A1.28}$$
$$= \frac{\mathbf{E}\big[w(Z)s_0(Z)D(D - p(Z))\big]}{\mathbf{E}\big[w(Z)w^{-1}(Z)\big]} \tag{A1.29}$$
$$= \mathbf{E}\big[w(Z)s_0(Z)D(D - p(Z))\big] \tag{A1.30}$$

Relying on the LIE again, this equation can be rewritten as:

$$= \mathbf{E}\big[w(Z)s_0(Z)\mathbf{E}[D(D - p(Z)) \mid Z]\big] \tag{A1.31}$$

To simplify the inner expectation term, the term $p(Z)(D - p(Z))$ is simultaneously added and subtracted. In a next step, the equation is rearranged using the LIE, such that the relationship from Equations (A1.16) to (A1.20) can be used again. Addi-

tionally, $p(Z)$ can be pulled out of the expectation since it is constant conditional on Z, such that Equation (A1.9) can be applied.

$$\begin{aligned}
\mathbf{E}\big[D(D-p(Z)) \mid Z\big] &= \mathbf{E}\big[D(D-p(Z)) + \overbrace{\big(p(Z)-p(Z)\big)(D-p(Z))}^{=0} \mid Z\big] && \text{(A1.32)}\\
&= \mathbf{E}\big[(D-p(Z))(D-p(Z)) + p(Z)(D-p(Z)) \mid Z\big] && \text{(A1.33)}\\
&= \mathbf{E}\big[(D-p(Z))^2 \mid Z\big] + \mathbf{E}\big[p(Z)(D-p(Z)) \mid Z\big] && \text{(A1.34)}\\
&= w^{-1}(Z) + p(Z)\underbrace{\mathbf{E}\big[D-p(Z) \mid Z\big]}_{=0} && \text{(A1.35)}\\
& && \text{(A1.36)}
\end{aligned}$$

Inserting these results into the equation for β_1 yields the desired result:

$$\begin{aligned}
\beta_1 &= \mathbf{E}\big[w(Z)w^{-1}(Z)s_0(Z)\big] && \text{(A1.37)}\\
\beta_1 &= \mathbf{E}\big[s_0(Z)\big] && \text{(A1.38)}
\end{aligned}$$

Solving Equation (A1.26) for β_2 gives:

$$\beta_2 = \frac{\mathbf{E}\big[w(Z)s_0(Z)D(D-p(Z))(S-\mathbf{E}[S])\big]}{\mathbf{E}\big[w(Z)(D-p(Z))(S-\mathbf{E}[S])(D-p(Z))(S-\mathbf{E}[S])\big]} \qquad \text{(A1.39)}$$

Relying on the LIE again and using the results and relationships derived above, the equation can be simplified to obtain:

$$\begin{aligned}
\beta_2 &= \frac{\mathbf{E}\big[w(Z)w^{-1}(Z)s_0(Z)(S-\mathbf{E}[S])\big]}{\mathbf{E}\big[w(Z)(D-p(Z))^2(S-\mathbf{E}[S])^2\big]} && \text{(A1.40)}\\
&= \frac{\mathbf{E}\big[s_0(Z)(S-\mathbf{E}[S])\big]}{\mathbf{E}\big[w(Z)w^{-1}(Z)(S-\mathbf{E}[S])^2\big]} && \text{(A1.41)}\\
&= \frac{\mathbf{E}\big[s_0(Z)(S-\mathbf{E}[S])\big]}{\mathbf{E}\big[(S-\mathbf{E}[S])^2\big]} && \text{(A1.42)}
\end{aligned}$$

Focusing on the numerator first, the term can again be expanded by simultaneously adding and subtracting $\mathbf{E}[s_0(Z)(S-\mathbf{E}[S])]$. The equation can then be rearranged using the linearity property of expectations:

$$\begin{aligned}
&\mathbf{E}\big[(s_0(Z)(S-\mathbf{E}[S])) + \overbrace{\mathbf{E}[\big(s_0(Z)-s_0(Z)\big)(S-\mathbf{E}[S])]}^{=0}\big] && \text{(A1.43)}\\
&= \mathbf{E}\big[\{s_0(Z)-\mathbf{E}[s_0(Z)]\}\{S-\mathbf{E}[S]\} + \mathbf{E}[s_0(Z)(S-\mathbf{E}[S])]\big] && \text{(A1.44)}\\
&= \mathbf{E}\big[\{s_0(Z)-\mathbf{E}[s_0(Z)]\}\{S-\mathbf{E}[S]\}\big] + \mathbf{E}\big[s_0(Z)\big]\underbrace{\mathbf{E}\big[S-\mathbf{E}[S]\big]}_{=0} && \text{(A1.45)}
\end{aligned}$$

The numerator therefore corresponds to the covariance between $s_0(Z)$ and S:

$$\mathbf{E}\big[s_0(Z)(S - \mathbf{E}[S])\big] = \mathbf{E}\big[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}\big] \tag{A1.46}$$

$$= \mathbf{Cov}(s_0(Z), S) \tag{A1.47}$$

The denominator $\mathbf{E}\big[(S - \mathbf{E}[S])^2\big]$ corresponds to the second central moment of S, thus:

$$\mathbf{E}\big[(S - \mathbf{E}[S])^2\big] = \mathbf{Var}(S) \tag{A1.48}$$

Hence, inserting the results from Equations (A1.47) and (A1.48) into Equation (A1.41), β_2 is given by:

$$\beta_2 = \frac{\mathbf{Cov}(s_0(Z), S)}{\mathbf{Var}(S)} \tag{A1.49}$$

The proof proceeds by showing that the just derived parameters β_1 and β_2 are solutions to the optimality criterion of linear prediction of $s_0(Z)$ using $S(Z)$ given by:

$$\mathbf{E}\big[s_0(Z) - \beta_1 - \beta_2(S - \mathbf{E}[S])\big] = 0 \tag{A1.50a}$$

$$\mathbf{E}\big[\{s_0(Z) - \beta_1 - \beta_2(S - \mathbf{E}[S])\}\{S - \mathbf{E}[S]\}\big] = 0 \tag{A1.50b}$$

Plugging in β_1 and β_2 into Equation (A1.50a) yields:

$$\mathbf{E}\left[s_0(Z) - \mathbf{E}[s_0(Z)] - \frac{\overbrace{\mathbf{E}[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}]}^{=\mathbf{Cov}(s_0(Z),S)}}{\underbrace{\mathbf{E}[(S - \mathbf{E}[S])^2]}_{=\mathbf{Var}(S)}}(S - \mathbf{E}[S])\right]$$

Using the linearity of expectation and the fact that $\mathbf{E}[\mathbf{E}[s_0(Z)]] = \mathbf{E}[s_0(Z)]$, the first two terms cancel out. Additionally, since both the numerator and denominator are expectations, the term preceding $(S - \mathbf{E}[S])$ can be pulled out of the outer expectation resulting in:

$$\frac{\mathbf{E}[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}]}{\mathbf{E}[(S - \mathbf{E}[S])^2]}\underbrace{\mathbf{E}\big[S - \mathbf{E}[S]\big]}_{=0} \tag{A1.51}$$

Plugging in β_1 and β_2 into Equation (A1.50b) yields:

$$\mathbf{E}\left[\left\{s_0(Z) - \mathbf{E}[s_0(Z)] - \frac{\mathbf{E}[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}]}{\mathbf{E}[(S - \mathbf{E}[S])^2]}(S - \mathbf{E}[S])\right\}\left\{S - \mathbf{E}[S]\right\}\right]$$

Pulling $\mathbf{E}[s_0(Z)]$ out of the expectation again and using that $\mathbf{E}[S - \mathbf{E}[S]] = 0$, the second term inside the first parentheses multiplied by $(S - \mathbf{E}[S])$ is zero. The fraction can be pulled out of the expectation as well leading to:

$$\begin{aligned}&\mathbf{E}\big[s_0(Z)(S - \mathbf{E}[S])\big] - \frac{\mathbf{E}[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}]}{\mathbf{E}[(S - \mathbf{E}[S])^2}\mathbf{E}\big[(S - \mathbf{E}[S])^2\big] \\ &= \mathbf{E}\big[s_0(Z)(S - \mathbf{E}[S])\big] - \mathbf{E}\big[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}\big] \qquad \text{(A1.52)}\end{aligned}$$

In Equation (A1.46) it was derived that both terms are equal:

$$\mathbf{E}\big[s_0(Z)(S - \mathbf{E}[S])\big] = \mathbf{E}\big[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}\big] \qquad \text{(A1.53)}$$

Thus,

$$\mathbf{E}\big[s_0(Z)(S - \mathbf{E}[S])\big] - \mathbf{E}\big[\{s_0(Z) - \mathbf{E}[s_0(Z)]\}\{S - \mathbf{E}[S]\}\big] = 0 \qquad \text{(A1.54)}$$

Therefore, the parameters $\beta_1 = \mathbf{E}[s_0(Z)]$ and $\beta_2 = \mathbf{Cov}(s_0(Z), S))/\mathbf{Var}(S)$ solve the optimality criteria in the problem of optimal linear prediction of $s_0(Z)$ using $S(Z)$ as stated in Section 2.2.

A.2 Proof of GATES Estimation Strategy*

Recall the weighted linear regression equation from Section 2.3 given by:

$$Y = \alpha' X_1 + \sum_{k=1}^{K} \gamma_k \big(D - p(Z)\big)\mathbf{1}(G_k) + \nu.$$

$$\text{with weights } w(Z) = \frac{1}{p(Z)(1 - p(Z))}.$$

The minimization problem from this weighted regression yields the following normal equations:

$$\mathbf{E}\big[w(Z)\big(Y - \alpha' X_1 - \gamma' X_2\big)X_2\big] = 0 \qquad \text{(A2.1)}$$

Define:

$$X_1 := X_1(Z), \text{ e.g. } \big[1, B(Z)\big]' \qquad \text{(A2.2)}$$

$$X_2 := \big[\big(D - p(Z)\big)\mathbf{1}(G_k)_{k=1}^K\big]' \qquad \text{(A2.3)}$$

* The following proof is taken from the project studies thesis as submitted on April 04, 2020.

Y is again replaced with $b_0(Z) + s_0(Z)D + U$ leading to:

$$\mathbf{E}\big[w(Z)\big(b_0(Z) + s_0(Z) + U - \alpha' X_1 - \gamma' X_2\big)X_2\big] = 0 \tag{A2.4}$$

Following the same strategies as described in the proof for the BLP in Appendix A.1, it is used that:

$$\mathbf{E}\big[w(Z)b_0(Z)\underbrace{\mathbf{E}[X_2 \mid Z]}_{=0}\big] = 0 \tag{A2.5a}$$

$$\mathbf{E}\big[w(Z)\alpha' X_1\underbrace{\mathbf{E}[X_2 \mid Z]}_{=0}\big] = 0 \tag{A2.5b}$$

$$\mathbf{E}\big[w(Z)X_2\underbrace{\mathbf{E}[U \mid Z, D]}_{=0}\big] = 0 \tag{A2.5c}$$

Thus, the normal equations from Equation (A2.4) simplify to:

$$\mathbf{E}\big[w(Z)\big(s_0(Z)D - \gamma' X_2\big)X_2\big] = 0 \tag{A2.6}$$

Since $\mathbf{E}[(D - p(Z))(D - p(Z)) \mid Z] = 0$, the components of X_2 are orthogonal in this case as well. Thus, the above equation can be solved for γ resulting in:

$$\gamma_k = \frac{\mathbf{E}\big[w(Z)s_0(Z)D(D - p(Z))\mathbf{1}(G_k)_{k=1}^K\big]}{\mathbf{E}\big[w(Z)(D - p(Z))^2\mathbf{1}(G_k)_{k=1}^K\big]} \tag{A2.7}$$

In the numerator, it is exploited that $\mathbf{E}[D(D-p(Z))] = w^{-1}(Z)$ as derived in Equations (A1.32) to (A1.35). In the denominator the equality from Equations (A1.16) to (A1.20) is used. Since $\mathbf{1}(G_k)_{k=1}^K$ is one for observations belonging to group k only and zero otherwise, it can be omitted by conditioning on the groups G_k. Doing so yields the expression for γ_k as stated in Section 2.3:

$$\gamma_k = \frac{\mathbf{E}\big[w(Z)w^{-1}(Z)s_0(Z) \mid G_k\big]}{\mathbf{E}\big[w(Z)w^{-1}(Z) \mid G_k\big]} \tag{A2.8}$$

$$\gamma_k = \mathbf{E}\big[s_0(Z) \mid G_k\big] \tag{A2.9}$$

A.3 Supplementary Tables

Table A1: Variables included in RWI-GEO-GRID

Category	Variable
Basic	Number of houses
	Number of private households*
	Number of commercial buildings*
	Number residential buildings*
Mobility	Car density: Ratio between number of cars and number of households*
	Share Audi of total makes of car
	Share BMW of total makes of car
	Share Fiat of total makes of car
	Share Ford of total makes of car
	Share Mazda of total makes of car
	Share Mercedes of total makes of car
	Share Nissan of total makes of car
	Share Opel of total makes of car
	Share Peugeot of total makes of car
	Share Renault of total makes of car
	Share other Asian cars of total makes of car
	Share other makes of car of total makes of car
	Share Toyota of total makes of car
	Share VW of total makes of car
	Share convertibles of total cars
	Share SUVs of total cars
	Share small cars of total cars
	Share estate cars of total cars
	Share mini cars of total cars
	Share medium-sized cars of total cars
	Share upper medium-sized cars of total cars
	Share luxury cars of total cars
	Share other segments of total cars
	Share lower medium-sized cars of total cars
	Share utility cars of total cars
	Share vans of total cars

Table A1: Variables included in RWI-GEO-GRID (continued)

Category	Variable
Building Development	Share 1-2 family houses in homogeneous built-up road section*
	Share 1-2 family houses in non-homogeneous built-up road section*
	Share 3-5 family houses*
	Share 6-9 family houses*
	Share block of flats with 10-19 households*
	Share high-rise buildings with 20 households or more*
	Share predominantly for commercial purposes used houses*
Household	Share households with foreign head of household*
	Share single households
	Share couple households
	Share families with children*
	Share children of total people*
	Unemployment rate*
	Share ethnic background Germany
	Share ethnic background Italy
	Share ethnic background Turkey
	Share ethnic background Greece
	Share ethnic background Spain and Portugal
	Share ethnic background Balkans
	Share ethnic background Eastern Europe
	Share ethnic background Africa south of the Sahara
	Share ethnic background non-European Islamic states
	Share ethnic background Southern/ East/ Southeast Asia
	Share ethnic background others
	Share ethnic background late emigrants former Soviet Union
	Lowest default probability in payment
	Quite low default probability in payment
	Quite below-average default probability in payment
	Below-average default probability in payment
	Slightly below-average default probability in payment

Table A1: Variables included in RWI-GEO-GRID (continued)

Category	Variable
	Average probability
	Slightly above-average default probability in payment
	Above-average default probability in payment
	Highest default probability in payment
	Inhabitants (absolute)*
	purchase power (total)*
Population	Share male inhabitants 0-3 years*
	Share male inhabitants 3-6 years*
	Share male inhabitants 6-10 years*
	Share male inhabitants 10-15 years*
	Share male inhabitants 15-18 years*
	Share male inhabitants 18-20 years*
	Share male inhabitants 20-25 years*
	Share male inhabitants 25-30 years*
	Share male inhabitants 30-35 years*
	Share male inhabitants 35-40 years*
	Share male inhabitants 40-45 years*
	Share male inhabitants 45-50 years*
	Share male inhabitants 50-55 years*
	Share male inhabitants 55-60 years*
	Share male inhabitants 60-65 years*
	Share male inhabitants 65-70 years*
	Share male inhabitants over 75 years*
	Share female inhabitants 0-3 years*
	Share female inhabitants 3-6 years*
	Share female inhabitants 6-10 years*
	Share female inhabitants 10-15 years*
	Share female inhabitants 15-18 years*
	Share female inhabitants 18-20 years*
	Share female inhabitants 20-25 years*
	Share female inhabitants 25-30 years*
	Share female inhabitants 30-35 years*
	Share female inhabitants 35-40 years*
	Share female inhabitants 40-45 years*
	Share female inhabitants 45-50 years*

Table A1: Variables included in RWI-GEO-GRID (continued)

Category	Variable
	Share female inhabitants 50-55 years*
	Share female inhabitants 55-60 years*
	Share female inhabitants 60-65 years*
	Share female inhabitants 65-70 years*
	Share female inhabitants over 75 years*

Notes. The table is provided by the FDZ Ruhr together with the data. Asterisks after variables denote variables which are used in the empirical analysis during this thesis. The population variables are summarized as follows: share of inhabitants aged 0-25, share of inhabitants aged 25-40, share of inhabitants aged 40-65, and share of inhabitants aged 65 and older. For complete variable definitions and descriptions see Breidenbach and Eilers (2018).

Table A2: Variables included in RWI-GEO-RED

Category	Variable
Identifier	Object identifier
	Unique object identifier
Time period	Beginning of ad, year*
	Beginning of ad, month*
	Ending of ad, year
	Ending of ad, month
Object features	Facilities of object
	Number of bathrooms*
	Dummy: Protected historic building*
	Dummy: Usable as holiday home
	Available from
	Dummy: Guest toilet in object
	House type*
	Dummy: Basement in object*
	Dummy: Garage/parking space available
	Number of rooms*
	Number of floors*
	Construction phase

Table A2: Variables included in RWI-GEO-GRID (continued)

Category	Variable
	Dummy: Granny flat in object
	Type of real estate*
	Dummy: Rented when sold
	Rental income per month in EUR
	Dummy: Wheelchair-accessible, no steps
	Number of bedrooms
	Plot area*
	Usable floor space
	Living area*
Energy and structure information	Year of construction*
	Type of energy performance certificates
	Energy efficiency rating
	Energy consumption per year and square meter
	Dummy: Warm water consumption included in energy consumption
	Type of heating*
	Year of last modernization of object
	Condition of object*
Price information	Brokerage at contract conclusion
	Asking price in EUR*
	Security deposit
	Price of parking space in EUR
Regional information	Federal state
	Local labor market (Kosfeld and Werner, 2012)
	1-sqm raster cell following INSPIRE*
	Municipality Identifier (AGS, 2015)*
	District identifier (AGS, 2015)
	Address: postcode
Meta-information of advertisement	Number of clicks on customer profile
	Number of clicks on contact button
	Number of clicks on customer URL
	Number of clicks on share button
	Number of hits of ad

Table A2: Variables included in RWI-GEO-GRID (continued)

Category	Variable
	Days of availability of ad
	Date of data retrieval
Generated technical variables	Classification of object identifiers used more than once
	Spell counter within object identifier

Notes. The table is provided by the FDZ Ruhr together with the data. Asterisks after variables denote variables which are used in the empirical analysis during this thesis. For complete variable definitions and descriptions see Boelmann and Schaffner (2019).

Table A3: Distribution measures for the full and restricted sample

Variable	Group	Mean	Percentiles				
			5 %	25 %	50 %	75 %	95 %
Ad year	all	2012.1	2007	2009	2012	2015	2018
	compl	2013.9	2009	2012	2014	2016	2018
Asking price	all	296,304	70,000	159,000	229,299	332,000	695,000
	compl	339,112	98,000	190,000	270,000	389,000	780,000
Year of construction	all	1972.4	1900	1956	1982	2007	2016
	compl	1976.2	1900	1960	1984	2009	2017
Living area	all	180	91	121	146	195	352
	compl	174	95	125	150	200	323
Lot area	all	744	188	396	600	860	1,906
	compl	707	175	352	576	847	1,740
# Floors	all	2.1	1.0	2.0	2.0	3.0	3.0
	compl	2.2	1.0	2.0	2.0	3.0	3.0
# Rooms	all	6.0	3.0	4.0	5.0	7.0	11.0
	compl	6.1	4	4.5.0	5.0	7.0	11.0
# Bathrooms	all	1.8	1.0	1.0	2.0	2.0	3.0
	compl	1.9	1.0	1.0	2.0	2.0	4.0
Protected (0/1)	all	0.01	0	0	0	0	0
	compl	0.01	0	0	0	0	0

Table A3: Distribution measures for the full and restricted sample (continued)

Variable	Group	Mean	Percentiles				
			5 %	25 %	50 %	75 %	95 %
Basement (0/1)	all	0.4	0	0	0	1	1
	compl	0.6	0	0	1	1	1
Property condition	all	4.5	1.0	1.0	5.0	7.0	8.0
	compl	4.9	1.0	3.0	6.0	7.0	8.0
# Private households	all	874	48	215	556	1,232	2,623
	compl	896	57	239	592	1,264	2,627
# Industry buildings	all	119	5	25	66	145	395
	compl	118	6	29	70	149	377
# Housing buildings	all	399	35	145	329	595	982
	compl	422	41	162	353	624	1,015
Car density	all	1.1	0.7	0.9	1.1	1.3	1.5
	compl	1.1	0.7	1.0	1.1	1.3	1.5
Purch. power per capita	all	21,593	16,087	18,860	21,024	23,606	28,940
	compl	22,932	17,435	20,142	22,318	24,987	30,407
Foreign HH head (%)	all	7.0	0.2	2.6	5.5	9.8	18.8
	compl	7.6	0.4	3.0	6.2	10.6	19.5
Unemployment rate (%)	all	5.7	0.8	3.0	4.9	7.7	13.4
	compl	5.2	0.7	2.7	4.4	6.9	12.3
# Inhabitants	all	1,735	100	457	1,146	2,472	51,111
	compl	1785	118	501	1,215	2,559	5,170
Children (%)	all	0.3	0.2	0.2	0.3	0.3	0.4
	compl	0.3	0.2	0.2	0.3	0.3	0.4
Families (%)	all	33.5	4.5	20.6	32.5	44.7	66.5
	compl	32.6	3.3	18.4	31.2	44.5	68
Ind. used buildings (%)	all	2.4	0.0	0.9	1.7	2.9	6.3
	compl	2.3	0.0	0.9	1.7	2.9	6.3
Det. houses (hom.) (%)	all	27.4	1.0	9.0	21.1	40.0	76.7
	compl	27.1	1.3	9.2	20.8	39.3	75.3

Table A3: Distribution measures for the full and restricted sample (continued)

Variable	Group	Mean	Percentiles				
			5 %	25 %	50 %	75 %	95 %
Det. houses	all	28.5	4.5	19.8	28.6	36.8	50.4
(het.) (%)	compl	28.7	5.7	20.1	28.8	36.9	50
3-5 family	all	20.9	7.3	15.8	21.1	26.2	33.8
homes (%)	compl	20.9	4.5	15.8	21.0	26.2	3.38
6-9 family	all	11.7	0.0	3.5	9.9	17.9	30.6
homes (%)	compl	11.7	0.0	3.8	10.0	17.9	30.5
Housing	all	6.6	0.0	0.0	3.8	10.1	23.4
blocks (%)	compl	6.6	0.0	0.0	4.1	10.0	22.7
Skyscrapers	all	2.7	0.0	0.0	0.0	3.0	13.7
(%)	compl	2.8	0.0	0.0	0.0	3.3	14.4
Age: 0-25 (%)	all	24.6	19.1	22.9	24.8	26.6	29.3
	compl	24.4	19.5	22.9	24.6	26.1	28.6
Age: 25-40 (%)	all	17.1	13.2	15.4	16.9	18.6	22.1
	compl	16.9	13.0	15.1	16.6	18.2	22.1
Age: 40-65 (%)	all	37.3	32.8	35.3	37.1	39.1	42.3
	compl	37.6	33.0	35.7	37.5	39.3	42.4
Age: 65+ (%)	all	20.9	15.1	18.3	20.6	23.2	27.8
	compl	21.2	15.4	18.6	20.9	23.4	27.7

Notes. "all" denotes the full, unrestricted sample, which also contains observations with missing data for some of the variables, while "compl" denotes the restricted sample, which contains complete cases only, i.e. observations without missing values in any of the variables. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.

Table A4: Comparison of number of wind turbines and installed capacity as of 31.12.2018, taken from the project studies thesis as submitted on April 04, 2020

Federal state	Number of wind turbines			Installed capacity in MW		
	Own data	DWG	MaStR	Own data	DWG	MaStR
Baden-Württemberg	714	725	753	1,519	1,529	1,587
Bavaria	1,120	1,161	1,217	2,493	2,515	2,521
Berlin	-	4	10	-	12	12
Brandenburg	3,788	3,821	3,808	7,001	7,081	7,087
Bremen	89	91	92	199	198	201
Hamburg	68	65	71	121	128	122
Hessen	1,100	1,159	964	2,145	2,201	2,147
Lower Saxony	6,174	6,305	6,110	10,473	11,165	11,006
Mecklenburg-Vorpommern	1,718	1,920	1,821	3,165	3,366	3,257
NRW	3,613	3,726	3,404	5,788	5,773	5,760
Rhineland-Palatinate	1,663	1,748	1,671	3,527	3,589	3,547
Saarland	198	207	205	-	476	497
Saxony	875	899	956	1,213	1,227	1,268
Saxony-Anhalt	2,862	2,862	2,862	5,038	5,139	5,125
Schleswig-Holstein	2,950	2,959	3,301	6,543	6,536	6,773
Thuringia	833	859	909	1,554	157	1,643
Total	27,765	28,511	28,154	50,779	51,092	52,553

DWG: Deutsche WindGuard, data from Deutsche WindGuard (2019).

MaStR: Core Energy Market Data Register, data from Bundesnetzagentur (2020a).

Table A5: Data sources: data on wind turbines, taken from the project studies thesis as submitted on April 04, 2020

Federal state	Publicly available	Link	Provided by
Baden Württemberg	yes	https://bit.ly/3aewaRs	Landesanstalt für Umwelt Baden-Württemberg
Bavaria	yes	https://bit.ly/2U7WKG2	Bayerisches Landesamt für Umwelt
Berlin	no	-	-
Brandenburg	yes	https://bit.ly/33BsCpT	Landesamt für Umwelt Brandenburg
Bremen	yes	https://bit.ly/2UtDBxq	Senatorin für Klimaschutz, Umwelt, Mobilität, Stadtentwicklung und Wohnungsbau
Hamburg	yes	https://bit.ly/2J5NgFd	-
Hessen	yes	https://bit.ly/3bgXRJf	Hessisches Landesamt für Naturschutz, Umwelt und Geologie
Lower Saxony	yes	https://bit.ly/3doeNiK	Niedersächsisches Ministerium für Ernährung, Landwirtschaft und Verbraucherschutz
Mecklenburg-Vorpommern	yes	https://bit.ly/33Hptoj	Landesamt für Umwelt, Naturschutz und Geologie
NRW	no	-	Landesamt für Natur, Umwelt und Verbraucherschutz NRW
Rhineland-Palatine	no	-	Energieagentur Rheinland-Pfalz/Energieatlas
Saarland	yes	https://bit.ly/2QAdzXW	Landesamt für Umwelt- und Arbeitsschutz
Saxony	yes	https://bit.ly/33JgLpF	Sächsisches Landesamt für Umwelt, Landwirtschaft und Geologie
Saxony-Anhalt	no	-	Landesenergieagentur Sachsen-Anhalt GmbH
Schleswig-Holstein	yes	https://bit.ly/2XXfhap	Landesamt für Landwirtschaft, Umwelt und ländliche Räume
Thuringia	yes	https://bit.ly/2Uan96b	Thüringer Landesverwaltungsamt

Table A6: Full summary statistics

Variable	Means		
	Full sample	Treatment	Control
Asking price	341,360	270,276	382,578
Year of advertisement	2013.5	2013.7	2013.4
Year of construction	1976.4	1976.1	1976.6
Living area	174.1	168.3	177.4
Lot area	683.9	730.8	656.7
Number of floors	2.2	2.1	2.3
Number of rooms	6.1	6	6.2
Number of bathrooms	1.9	1.9	1.9
Protected building (0/1)	0.01	0.01	0.02
Basement (0/1)	0.62	0.57	0.66
Property condition	4.9	4.9	4.9
Number of private households*	926.5	781.8	1,010.5
Number of industrial buildings*	127.1	100.2	142.8
Number of living buildings*	433.7	407.9	448.6
Car density*	1.1	1.2	1.1
Purchasing power per capita*	22,962	21,871	23,594
Foreign household head (%)*	7.5	6.2	8.3
Unemployment rate (%)*	5.2	5.7	4.9
Number of inhabitants*	1,848	1,583	2,002
Children (%)*	0.29	0.29	0.28
Families (%)*	32.6	33.8	31.9
Industrially used buildings (%)*	2.4	2.3	2.4
Detached houses (hom.) (%)*	25.3	29.9	22.7
Detached houses (het.) (%)*	29.1	28.9	29.2
3-5 Family houses (%)*	21.2	20.6	21.5
6-9 Family houses (%)*	12.1	10.9	12.8
Housing blocks (%)*	6.9	5.4	7.8
Skyscrapers (%)*	3	2.1	3.5
Age: 0–25 (%)*	24.4	24.4	24.5
Age: 25–40 (%)*	16.9	16.3	17.2
Age: 40–65 (%)*	37.6	38.2	37.3
Age: 65+ (%)*	21	21	21
Heating: Cogeneration (0/1)	0.00	0.00	0.00
Heating: Electric (0/1)	0.00	0.00	0.00

Table A6: Full summary statistics (continued)

Variable	Means		
	Full Sample	Treatment	Control
Heating: Self-contained central (0/1)	0.03	0.03	0.03
Heating: District (0/1)	0.01	0.01	0.01
Heating: Floor (0/1)	0.06	0.06	0.05
Heating: Gas (0/1)	0.07	0.10	0.06
Heating: Wood pellet (0/1)	0.00	0.00	0.00
Heating: Night storage (0/1)	0.00	0.00	0.00
Heating: Stove (0/1)	0.03	0.02	0.03
Heating: Oil (0/1)	0.02	0.03	0.02
Heating: Solar (0/1)	0.00	0.00	0.00
Heating: Thermal heat pump (0/1)	0.03	0.03	0.03
Heating: Central (0/1)	0.74	0.72	0.76
Farmhouse (0/1)	0.01	0.01	0.01
Bungalow (0/1)	0.04	0.04	0.03
Semidetached house (0/1)	0.15	0.14	0.16
Single-family house (0/1)	0.48	0.52	0.46
Multi-family house (0/1)	0.11	0.10	0.12
Terraced house (0/1)	0.13	0.12	0.14
Villa (0/1)	0.03	0.02	0.03
Category: Other (0/1)	0.05	0.05	0.04
Schleswig-Holstein (0/1)	0.05	0.05	0.04
Hamburg (0/1)	0.01	0.01	0.01
Lower Saxony (0/1)	0.05	0.10	0.02
Bremen (0/1)	0.01	0.01	0.00
NRW (0/1)	0.30	0.46	0.20
Hessen (0/1)	0.11	0.06	0.14
Rhineland-Palatine (0/1)	0.10	0.11	0.09
Baden-Württemberg (0/1)	0.14	0.04	0.20
Bavaria (0/1)	0.13	0.04	0.18
Saarland (0/1)	0.01	0.01	0.02
Brandenburg (0/1)	0.04	0.04	0.04
Mecklenburg-Vorpommern (0/1)	0.01	0.02	0.01
Saxony (0/1)	0.03	0.03	0.03
Saxony-Anhalt (0/1)	0.02	0.03	0.01
Thuringia (0/1)	0.01	0.01	0.01

Notes. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.

Table A7: Covariate balance before and after PS weighting – all runs

	Criterion	Unadjusted	Logit	XGBoost	Random Forest
1	ASMD	0.1246	0.0236	0.0641	0.0945
	KS	0.0884	0.0234	0.0559	0.0682
	# Unbalanced	29	1	18	25
2	ASMD	0.1243	0.0244	0.0557	0.0936
	KS	0.0879	0.0242	0.0504	0.0671
	# Unbalanced	29	1	17	25
3	ASMD	0.1236	0.0265	0.0612	0.0936
	KS	0.0876	0.0244	0.0553	0.0669
	# Unbalanced	29	1	17	25
4	ASMD	0.1247	0.0254	0.0617	0.0942
	KS	0.0885	0.0261	0.0530	0.0679
	# Unbalanced	29	1	17	25
5	ASMD	0.1243	0.0248	0.0614	0.0935
	KS	0.0880	0.0240	0.0514	0.0671
	# Unbalanced	29	1	18	25
6	ASMD	0.1243	0.0252	0.0601	0.0936
	KS	0.0881	0.0244	0.0526	0.0668
	# Unbalanced	29	1	17	25
7	ASMD	0.1246	0.0225	0.0626	0.0946
	KS	0.0884	0.0229	0.0558	0.0685
	# Unbalanced	29	1	18	25
8	ASMD	0.1243	0.0245	0.0560	0.0930
	KS	0.0881	0.0238	0.0478	0.0666
	# Unbalanced	30	1	16	25
9	ASMD	0.1244	0.0236	0.0609	0.0938
	KS	0.0876	0.0229	0.0525	0.0666
	# Unbalanced	29	1	18	25
10	ASMD	0.1236	0.0274	0.0612	0.0927
	KS	0.0873	0.0259	0.0525	0.0660
	# Unbalanced	29	1	18	24

Notes: ASMD is the average of standardized absolute mean differences over all covariates. KS denotes the Kolmogorov-Smirnov statistic. # Unbalanced refers to the number of variables which exhibit a standardized absolute mean difference of more than 0.1. Smaller values indicate better balance after propensity score weighting for all three criteria. The grey cells mark the best results for the respective criterion and run of the experiment.

Table A8: Full GATES results

Elastic Net									
Group 1	2	3	4	5	6	7	8	9	Group 10
-20,419	-9,059	-8,275	-6,261	-5,407	-4,174	-2,601	-1,423	-905	-3,760
(-24,379 ; -16,521)	(-13,145 ; -4,902)	(-12,559 ; -4,083)	(-10,466 ; -2,084)	(-9,612 ; -1,084)	(-8,402 ; 124)	(-6,822 ; 1,755)	(-5,521 ; 2,698)	(-5,256 ; 3,435)	(-7,950 ; 494)
[0.000]	[0.000]	[0.000]	[0.006]	[0.029]	[0.115]	[0.454]	[0.774]	[0.756]	[0.163]
Random Forest									
Group 1	2	3	4	5	6	7	8	9	Group 10
-13,028	-4,313	-1,520	192	806	2,040	2,539	3,727	5,424	5,124
(-16,571 ; -9,552)	(-7,496 ; -1,243)	(-4,585 ; 1,738)	(-2,987 ; 3,348)	(-2,388 ; 3,868)	(-1,221 ; 5,178)	(-759 ; 5,743)	(286 ; 6,983)	(2,042 ; 8,704)	(1,491 ; 8,717)
[0.000]	[0.013]	[0.644]	[1.000]	[1.000]	[0.457]	[0.265]	[0.069]	[0.004]	[0.010]
XGBoost									
Group 1	2	3	4	5	6	7	8	9	Group 10
-12,956	-5,618	-3,013	-1,698	-1,048	-615	-34	181	382	-1,434
(-16,431 ; -9,446)	(-8,942 ; -2,339)	(-6,299 ; 329)	(-4,985 ; 1,461)	(-4,341 ; 2,141)	(-3,895 ; 2,648)	(-3,246 ; 3,163)	(-3,068 ; 3,555)	(-2,856 ; 3,723)	(-4,721 ; 1,885)
[0.000]	[0.002]	[0.152]	[0.574]	[0.830]	[1.000]	[1.000]	[1.000]	[1.000]	[0.167]

Notes. The reported results are medians over 100 splits. The first row for each algorithm shows the point estimates. 90 % CIs are in parentheses below estimated coefficients, and adjusted p-values are in square brackets in the last row of each panel.

Table A9: Full CLAN results

Variable	Elastic Net		Random Forest		XGBoost	
	Most	Least	Most	Least	Most	Least
Asking price	612,362	289,705	624,126	453,416	577,599	494,968
Year of construction	1968.6	1969.9	1974.8	1975.2	1969.4	1968.3
Protected building (0/1)	0.04	0.01	0.03	0.02	0.03	0.03
Purch. power / capita*	24,232	22,654	24,929	24,709	24,626	24,392
Unemployment rate (%)*	5.6	5.2	4.7	4.9	4.8	4.9
# Inhabitants*	2,473	1,896	2,212	2,253	2,121	2,137
Skyscrapers (%)*	4.1	3.7	3.9	4.3	3.9	4.0
Housing blocks (%)*	8.6	9.1	8.4	9.1	8.3	8.6
Age: 0-25 (%)*	24.2	24.4	24.2	24.2	24.1	24.1
Age: 25-40 (%)*	17.6	17.2	17.4	17.6	17.4	17.5
Age: 40-65 (%)*	37.1	37.1	37.1	37.0	37.1	37.0
Age: 65+ (%)*	21.1	21.3	21.4	21.2	21.5	21.4
Farmhouse (0/1)	0.01	0.01	0.01	0.01	0.01	0.02
Bungalow (0/1)	0.02	0.03	0.03	0.03	0.03	0.03
Semidetached house (0/1)	0.04	0.24	0.09	0.15	0.09	0.12
Single-family house (0/1)	0.42	0.27	0.46	0.42	0.44	0.41
Multi-family house (0/1)	0.22	0.07	0.16	0.14	0.17	0.16
Terraced house (0/1)	0.03	0.30	0.08	0.13	0.08	0.10
Villa (0/1)	0.19	0.002	0.11	0.07	0.10	0.09
Category: Other (0/1)	0.07	0.05	0.07	0.05	0.08	0.07
Property condition	5.2	5.1	4.7	4.7	5.0	5.0

Notes. The reported results are medians over 100 splits. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.

Table A10: Full GATES results – log specification

Elastic Net									
Group 1	2	3	4	5	6	7	8	9	Group 10
-0.020	-0.016	-0.011	-0.011	-0.009	-0.005	-0.002	0.000	0.004	0.000
(-0.027 ; -0.013)	(-0.024 ; -0.009)	(-0.018 ; -0.004)	(-0.018 ; -0.004)	(-0.017 ; -0.002)	(-0.013 ; 0.002)	(-0.010 ; 0.005)	(-0.008 ; 0.008)	(-0.004 ; 0.011)	(-0.007 ; 0.008)
[0.000]	[0.000]	[0.007]	[0.005]	[0.035]	[0.251]	[0.547]	[0.695]	[0.501]	[0.363]
Random Forest									
Group 1	2	3	4	5	6	7	8	9	Group 10
-0.012	-0.008	-0.004	-0.001	0.001	0.002	0.004	0.006	0.009	0.019
(-0.018 ; -0.006)	(-0.014 ; -0.002)	(-0.009 ; 0.002)	(-0.007 ; 0.005)	(-0.005 ; 0.007)	(-0.004 ; 0.008)	(-0.003 ; 0.010)	(0.001 ; 0.012)	(0.003 ; 0.015)	(0.013 ; 0.025)
[0.000]	[0.018]	[0.358]	[0.798]	[0.715]	[0.548]	[0.493]	[0.119]	[0.006]	[0.000]
XGBoost									
Group 1	2	3	4	5	6	7	8	9	Group 10
-0.015	-0.012	-0.009	-0.008	-0.006	-0.006	-0.005	-0.005	-0.005	-0.003
(-0.022 ; -0.009)	(-0.018 ; -0.005)	(-0.016 ; -0.003)	(-0.015 ; -0.001)	(-0.013 ; 0.001)	(-0.013 ; 0.001)	(-0.011 ; 0.002)	(-0.012 ; 0.001)	(-0.011 ; 0.002)	(-0.010 ; 0.003)
[0.000]	[0.001]	[0.013]	[0.035]	[0.125]	[0.153]	[0.258]	[0.209]	[0.301]	[0.376]

Notes. The reported results are medians over 100 splits. The first row for each algorithm shows the point estimates. 90 % CIs are in parentheses below estimated coefficients, and adjusted p-values are in square brackets in the last row of each panel.

Table A11: Full CLAN results – log specification

Variable	Elastic Net		Random Forest		XGBoost	
	Most	Least	Most	Least	Most	Least
Asking price	12.6	12.8	12.7	12.5	12.7	12.5
Year of construction	1944.1	1983.5	1970.6	1974.7	1961.1	1967.8
Protected building	0.04	0.02	0.03	0.02	0.03	0.03
Purch. power / capita*	23,797	22,606	24,427	23,029	23,632	22,986
Unemployment rate*	5.7	5.1	5.3	5.3	5.3	5.3
# Inhabitants*	2,596	1,693	2,036	1,868	2,276	1,923
Skyscrapers*	3.5	3.6	3.3	3.1	3.7	3.1
Housing blocks*	8.6	7.7	7.8	7.3	8.3	7.4
Age: 0-25 (%)*	24.2	24.5	24.1	24.3	24.1	24.4
Age: 25-40 (%)*	17.7	17.0	17.1	17.0	17.4	17.1
Age: 40-65 (%)*	37.0	37.5	37.5	37.7	37.1	37.5
Age: 65+ (%)*	21.2	21.1	21.3	21.1	21.4	21.0
Farmhouse	0.01	0.03	0.01	0.01	0.01	0.03
Bungalow	0.02	0.04	0.03	0.03	0.03	0.03
Semidetached house	0.07	0.18	0.11	0.13	0.10	0.15
Single-family house	0.39	0.45	0.49	0.50	0.45	0.41
Multi-family house	0.25	0.10	0.14	0.12	0.15	0.15
Terraced house	0.13	0.07	0.11	0.11	0.12	0.13
Category: Other	0.07	0.06	0.06	0.05	0.07	0.05
Villa	0.06	0.05	0.06	0.05	0.07	0.05
Property condition	6.2	4.2	4.9	4.6	5.5	5.0

Notes. The reported results are medians over 100 splits. (0/1) after variables denote dummy variables, asterisks indicate grid-level variables.