Analysing recently developed causal machine learning methods

Erik de Knegt (486559)

Supervisor: dr. A. Pick
Second supervisor: dr. A.A. Naghi

Abstract

We exploit the strengths of recently developed causal machine learning methods. We analyse *Double Machine Learning*, *Generic Machine Learning* and the *Causal Forest* extensively. Researchers answering causal questions benefit from using these methods. We apply them onto a revisited study. Concerning the average treatment effect, the methods do not shed a new light on this study. However, for individual treatment effects and group average treatment effects they do.

Thesis Master Business Analytics and Quantitative Marketing (Econometrics and Operational Research)

February 2023
Erasmus University Rotterdam
Erasmus School of Economics

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Contents

1 Introduction **2**

2 Methodology **4**

2.1 General framework 4

2.2 Linear Probability Model 5

2.3 Logit and Probit model 6

2.4 Comparison Linear Probability Model with Logit/Probit Model 6

2.5 Double/Debiased Machine Learning 7

2.6 Generic Machine Learning 10

2.7 Causal Forest 12

3 Data description **14**

4 Application **16**

4.1 Traditional models 16

4.2 Double Machine Learning 17

4.3 Generic Machine Learning 17

4.4 Causal Forest 20

5 Conclusion **20**

References **22**

A Derivations **24**

A.1 Double Machine Learning 24

A.2 Generic Machine Learning 26

A.3 Causal Forest 27

B Variables **28**

C Additional tables **29**

C.1 Ordinary Least Squares 29

C.2 Probit model 30

C.3 Logit model 31

C.4 Generic Machine Learning 32

1 Introduction

Goal Our goal is to analyse three causal machine learning methods: *Double Machine Learning*, *Generic Machine Learning* and the *Causal Forest*. Often, traditional econometric methods are used for estimating treatment effects. Machine learning, however, is a field that consists of methods that are mostly used for prediction and for finding complicated patterns in data. Causal machine learning methods have been developed in recent years and combine these strengths. We analyse the mentioned methods and compare them with the Linear Probability Model and the Probit/Logit model.

Approach Before digging deeper into the methods, we create a general framework. When explaing the methods, we use the framework. In this way, we make clear how the methods relate to each other.

We explain the methods briefly. *Double/Debiased Machine Learning* (Chernozhukov, Chetverikov, et al., 2018) starts with doing two regressions. The first regression estimates the influence of the regressors on the dependent variable. The second estimates the influence of the regressors on the treatment assignment. The goal of the last regression is to deal with potential correlation between the treatment assignment and the regressors. *Double Machine Learning* is the method that enables us to partial out that correlation. The estimated treatment effects are accompanied by confidence intervals. According to its designers, the method is efficient and fast.

Chernozhukov, Demirer, Duflo, and Fernandez-Val (2018) came up with *Generic Machine Learning*. Its focus is to explore the heterogeneity in the treatment effects. The method consists of three stages. In the first stage, treatment effects are estimated individually. Then, the sample is separated into groups according to the treatment effect. This enables the user to get insight in the characteristics of individuals that respond similarly to the treatment. The last stage is to look for differences in the regressors among various groups.

The *Causal Forest* is a Random Forest that was adjusted by Wager and Athey (2018) to enable the estimation of treatment effects. Individuals are partitioned according to the values of the regressors, just as in a Random Forest. The modification is that each leaf represents a treatment effect. The partitioning is such that the treatment effect is the same within each leaf, but different across leaves.

We apply the methods on the study of Grinstein-Weiss et al. (2013) from the *American Economic Journal: Economic Policy*. We refer to this paper as the 'replication paper' regularly. The researchers analyse whether providing renters an Individual Development Account (IDA) has a long-term impact on homeownership. To give more context, an experiment was done in the period 2000 till 2003 among renters. Individuals in the treatment group were provided with an IDA, while, naturally, individuals in the control group were not. Grinstein-Weiss et al. (2008) [Note the different year!] analyse whether the renters with an IDA were more often the owner of a home at the end of the experiment. They concluded that this was indeed the case. In the study of our specific interest, they analyse whether the impact would also hold on the long term. The question was whether the individuals with an IDA owned a home more often than those who did not. Grinstein-Weiss et al. use a Linear Probability Model to give an answer to this problem. We analyse how their results relate to our results with causal machine learning methods. The treatment effects, accompanied by standard errors and confidence intervals, have our main interest.

Other studies on causal machine learning Several empirical studies have used these methods to find answers to causal questions. For example, Baiardi and Naghi (2020) use five papers in which traditional methods were used to answer a causal question. Like us, they use *Double Machine*

Learning, *Generic Machine Learning* and the *Causal Forest* to answer these questions. Their conclusion is that using causal machine learning methods adds value to traditional methods and should be used more often. We do a similar analysis, but our analysis of the methods differs in the sense that we build a general framework and provide a more extensive explanation.

Davis and Heller (2017) use a *Causal Forest* to predict treatment heterogeneity. They make use of data from a Randomized Control Trial to investigate whether youth summer job programs caused a decrease in violent-crime arrests. The *Causal Forest* helped them to identify two subgroups that respond differently to the treatment. Our application is similar, as we also identify subgroups that have a different treatment effect.

Deryugina, Heutel, Miller, Molitor, and Reif (2019) estimate the number of life-years that people loose due to pollution exposure. They use panel data for which each person-day observation is assigned to the treatment or control group, depending on the wind direction. First, they use OLS to estimate the average treatment effect using the complete data set. In addition to this, they estimate the average treatment effect within certain pre-defined groups. Deryugina et al. argue that this method may mask additional heterogeneity. Their solution is to use *Generic Machine Learning*. They take into consideration that this machine learning approach may reduce the interpretability, but it offers flexibility. We follow the same approach: Grinstein-Weiss et al. used OLS to estimate the average treatment effect. To search for treatment heterogeneity, they include interaction parameters (which interact with the treatment parameter). We apply *Generic Machine Learning* to find out the sources of heterogeneity without pre-defining subgroups.

Background on IDAs One of the most prominent issues that governments face is how to divide wealth in a country. In line with this, governments take measures to improve poor people's living situation. These measures can be direct or indirect. Direct measures include paying less taxes or the right for certain allowances. Indirect measures include education, consumption support and work incentives. Provision of an Individual Development Account is an indirect measure. Its intention is to expand wealth of the poor (M. Sherraden & Gilbert, 2016) by creating consciousness on how to spend their money. Various types of IDAs have been used by governments. Even accounts for children are in use (M. S. Sherraden & McBride, 2010).

The goal of indirect measures is to help people on the long term. Several studies have been done on the influence of IDAs, but they are all about the effects during the period that the IDA was active. Therefore, Grinstein-Weiss et al. (2013) investigate the influence of the IDA program six years later.

Added value Our research adds value to the existing literature in two ways. First, we provide an accessible explanation of the mentioned methods. The methods are all introduced recently. It is important that these new methods are evaluated and compared to traditional methods. By analysing the results, we put those new methods into the context of pre-existing methods. We contribute to the awareness of causal machine learning methods in this way. Second, our general framework shows the common ground of the different methods. Different notations and vocabulary are used in different papers. The similarity between different methods can therefore be hard to see. This helps the reader to put the methods into the context of other methods.

Notation remark As we use different models and try to fit them into one framework, we use some notation rules. For matrices, we use capital letters, such as X. For vectors, we use bold letter notation, such as $\boldsymbol{\theta}$ or $\boldsymbol{x}_i$. For elements within a matrix, we use small letter notation with subscript, such as x_{1i}. For scalars that indicate the size of a matrix or vector, we use capital letters. Samples and subsets are indicated by calligraphic fonts, such as $\mathcal{G}$.

Conclusion Our main goal is to analyse the mentioned methods. With our dataset, the average treatment effect hardly differs from OLS when using causal machine learning methods. Also, the standard errors are similar. *Generic Machine Learning* gives extra opportunities to find out sources of heterogeneity, compared to OLS.

2 Methodology

We set up a general framework. It fits traditional models as well as causal machine learning models. We translate all equations that are used in the original papers to our own notation. You can find these derivations in Appendix A. When seperately elaborating on the methods, we build on this general framework.

After that, we explain the Linear Probability Model, as Grinstein-Weiss et al. (2013) use this model to determine the treatment effect. As the dependent variable is binary, we also estimate the treatment effect with a Probit/Logit model. We expect the reader to be familiar with these traditional methods, so the explanations are brief.

Furthermore, we elaborate on the mentioned causal machine learning methods: *Double Machine Learning*, *Generic Machine Learning* and the *Causal Forest*. We explain how these methods work, what their characteristics are, and what their advantages and their disadvantages are. We relate these methods to the traditional methods.

2.1 General framework

Setup We define the general framework using the Rubin causal model (Rubin, 1974). Rubin defined the causal effect as the difference between the dependent variable when being treated and when being not treated. To do causal inference, let

$$\mathbf{y} = \boldsymbol{\theta} \circ \boldsymbol{d} + \Gamma(X) + \boldsymbol{u} \tag{1}$$

Let $\mathcal{S}$ be the sample with N individuals. K is the number of regressors. The vector $\mathbf{y}$ is the dependent variable. It is a binary vector with N elements: ones indicate that individuals had a home in 2010 and zeroes indicate that they did not. $\boldsymbol{d}$ is the treatment vector, with size N. This is a binary vector as well: ones are used when individuals participated in the program; zeroes for the others. We refer to the first group as the treatment group, while the second group is referred to as the control group. X is an $N \times K$ matrix with the regressors and $\boldsymbol{u}$ is a vector with the disturbance terms. It is N elements long. Each individual i is a triple $b_i = (y_i, d_i, \boldsymbol{x}_i)$.

$\Gamma(X)$ is an unknown specification that expresses the influence of X on $\boldsymbol{y}$. Define the function $\gamma(\boldsymbol{x}_i)$, such that

$$\Gamma(X) = \Gamma\left(\begin{bmatrix}\boldsymbol{x}_1\\ \dots\\ \boldsymbol{x}_N\end{bmatrix}\right) = \begin{pmatrix}\gamma(\boldsymbol{x}_1)\\ \dots\\ \gamma(\boldsymbol{x}_N)\end{pmatrix} \tag{2}$$

An example function is $\gamma(\boldsymbol{x}_i) = 2\log(x_{i1}) + 2.5\exp x_{i2} - 0.5^{x_{i3}}$. Note that we assume that the data is individually independently distributed. In this way, the relationship between the regressors and the dependent variable is the same for any individual.

We are particularly interested in $\boldsymbol{\theta}$. This vector contains the treatment effects of all N individuals. OLS, Probit, Logit and *Double Machine Learning* estimate the treatment effect homogeneously.

Therefore, $\boldsymbol{\theta}$ will contain one value N times for these methods. *Generic Machine Learning* and the *Causal Forest* estimate heterogeneous treatment effects, so $\boldsymbol{\theta}$ will contain various values for these methods. When we estimate the group average treatment effects, individuals belonging to the same group will have the same treatment effect. Mathematically spoken, the values in $\boldsymbol{\theta}$ that correspond to these individuals are the same. When we are talking about the average treatment effect, sometimes we use $\theta = \bar{\boldsymbol{\theta}}$ for reasons of ease.

To reduce the variance, we run an algorithm R times. Let $\hat{\boldsymbol{\theta}}_{(r)}$ be the estimate of $\boldsymbol{\theta}$ of the r-th run. In the end, let

$$\hat{\boldsymbol{\theta}} = \frac{1}{R}\sum_{r=1}^{R}\hat{\boldsymbol{\theta}}_{(r)} \tag{3}$$

Assumptions We do the following assumption when we use the algorithms:

1. We assume that the triples $b_i = (y_i, d_i, \boldsymbol{x}_i)$ are independently identically distributed.

For the causal machine learning models, the following assumptions hold as well:

2. We assume an additive treatment effect. we already used this in Equation 1.

3. We use the Stable Unit Treatment Value Assumption, as proposed by Angrist, Imbens, and Rubin (1996). This states that the (potential) value of the dependent variable of an individual should be unaffected by the assignment of treatments to other individuals.

4. We assume that all variables that affect $\mathbf{y}$ as well as $\boldsymbol{d}$ are known. In other words, assignment of the treatment is independent of $\mathbf{y}$, conditional on X.

If the model specification is correct and the assumptions hold, we can estimate the treatment effect and the relation between the regressors and the dependent variable. In practice, $\boldsymbol{d}$ and X correlate with each other. This is obvious for observational studies, but also in randomized studies, the randomization is never perfect. A solution could be to add many regressors, but we encounter two problems here. This makes the model overly complex and this increases the risk of overfitting. And even then there may be unobserved or unobservable regressors. Concluding, normal regression models do not enable us to do causal inference. Therefore, we need other methods, such as causal machine learning methods.

2.2 Linear Probability Model

Ordinary Least Squares is a method that investigates the linear relationship between regressors and dependent variable. We do the following assumptions on top of Assumption 1:

2. The dependent variable has a linear relationship with the regressors.

3. There is no multicollinearity between the regressors.

4. There is homoskedasticity.

5. Error terms are normally distributed.

Using the general framework in Equation 1, let $\Gamma(X) = X\beta$, where β is a $K \times 1$ vector with the coefficients corresponding to the regressors. It follows that we estimate the regression

$$\mathbf{y} = \boldsymbol{\theta} \circ \boldsymbol{d} + X\beta + \boldsymbol{u} \tag{4}$$

Vector $\boldsymbol{\theta}$ does not contain different values, so notation will be more understandable when we consider $\boldsymbol{d}$ as one of the regressors. Therefore let

$$\mathbf{y} = X^* \beta^* + \boldsymbol{u} \tag{5}$$

with $X^* = [X|\boldsymbol{d}]$ and $\beta^* = [\beta|\theta]'$. Because of the binary nature of $\boldsymbol{y}$, we should interpret its expected value as the probability that $y_i = 1$. This fact makes that we speak about the Linear Probability Model. The error terms are clearly heteroskedastic. Therefore, we use robust standard errors.

To estimate $\hat{\beta}$, we want to minimize the sum of the squared residuals in $\boldsymbol{u}$, which is equivalent to solving

$$\hat{\beta}^* = (X^{*\prime} X^*)^{-1} X^{*\prime} \mathbf{y} \tag{6}$$

Now, the treatment effect is $\hat{\beta}^*_{K+1}$. This coefficient has to be interpreted as the marginal effect of the treatment on the probability that $y_i = 1$.

2.3 Logit and Probit model

We use the general framework from Equation 1 again. As well as the Linear Probability Model, the Logit/Probit model treats the treatment effect just as the other regressors. It is estimated homogeneously, and we let $\Gamma(X) = X\beta$, such that we estimate the regression in Equation 5 again, but differently.

Now the problem with Linear Probability Model is that the dependent variable is binary. Therefore, in the Logit/Probit model we define

$$\tilde{\mathbf{y}} = \begin{cases} 1 & \text{if } \mathbf{y} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$= \begin{cases} 1 & \text{if } X^*\beta^* + \boldsymbol{u} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

We can state that the border must be at 0 without loss of generality, as we can add or subtract any fixed amount from the intercept. It follows that

$$\begin{aligned} \mathbb{P}(\tilde{\mathbf{y}} = 1|X^*) &= \mathbb{P}(\mathbf{y} > 0|X^*) && (9) \\ &= \mathbb{P}(X^*\beta^* + \boldsymbol{u} > 0|X^*) && (10) \\ &= \mathbb{P}(\boldsymbol{u} < X^*\beta^*|X^*) && (11) \\ &= \Phi(X^*\beta^*) && (12) \end{aligned}$$

where Φ is the logistic distribution in the Logit model and the normal distribution in the Probit model. We estimate this with Maximum Likelihood.

2.4 Comparison Linear Probability Model with Logit/Probit Model

The Linear Probability Model is an Ordinary Least Squares (OLS) model with a binary dependent variable. OLS estimates the relationship between the regressors and the dependent variable linearly. This is possible, but it has drawbacks. Assumption 2 states that there is a linear relationship between the regressors and the dependent variable and assumption 7 states that the error terms are normally distributed. Of course, these assumptions are not true for a binary regressor, as it is discrete. The result is that predictions can be below 0 or above 1, which is not applicable.

The advantage of the Linear Probability Model is especially in the interpretability. The coefficients in the Probit model represent the marginal effects. Therefore, they are not directly interpretable; their meaning depends on the constant and the values of the other coefficients. The only thing that can be directly interpretated is that a positive coefficient means that an increase in the regressor leads to an increase in the predicted probability.

Although coefficients in the Logit model are easier interpretable, they are not directly interpretable as well. They represent the expected change in log odds of having the dependent variable per individual change of the corresponding regressor.

The Probit model uses the normal distribution, which is an advantage with respect to the Logit model. If the data deviates from normal, however, Logit is more robust against this. Choosing between the LPM and Logit/Probit, Deke (2014) argue that researchers often prefer the LPM for treatment effects. We use all three models.

2.5 Double/Debiased Machine Learning

General Chernozhukov, Chetverikov, et al. (2018) proposed the *Double/Debiased Machine Learning* (DML) method. Its idea is as follows: we predict both the treatment assignment $\boldsymbol{d}$ and the outcome variable $\boldsymbol{y}$ from the regressors. Subsequently, we predict $\boldsymbol{y} - \hat{\boldsymbol{y}}$ from $\boldsymbol{d} - \hat{\boldsymbol{d}}$. This way, we partial out the effect of the regressors, so that we end up with a 'clean' treatment effect. Naturally, we can only trust the estimated treatment effect when all regressors that influence the treatment assignment are known. Later in this section, we work this out more mathematically. We show a visualisation of the procedure in Figure 1.

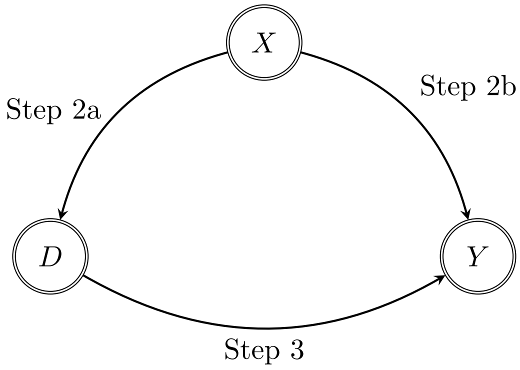

Figure 1: Graph that shows how Y, X and D are related to each other in *Double Machine Learning*. The step numbers correspond with the step numbers that follow later in this section.

Chernozhukov, Chetverikov, et al. (2018) argue that the algorithm is especially suitable in problems where many regressors are involved, while there is a small amount of data. The method performs well on estimating causal parameters. The procedure is $\sqrt{n}$-consistent, which means that the estimation error goes to zero at a rate of $\frac{1}{\sqrt{n}}$. We emphasize that DML differs from traditional estimation methods, such as OLS and IV estimation in the sense that $\gamma(\boldsymbol{x}_i)$ and $\mu(\boldsymbol{x}_i)$ can take on high-dimensional, non-linear functions.

Partially Linear Regression Model We extend the model of Equation 1. For reasons of completeness, we also repeat Equation 1 below.

$$\mathbf{y} = \boldsymbol{\theta} \circ \boldsymbol{d} + \Gamma(X) + \boldsymbol{u} \qquad \mathbb{E}[\boldsymbol{u}|\boldsymbol{d}, X] = 0 \tag{1 revisited}$$
$$\boldsymbol{d} = M(X) + \boldsymbol{v} \qquad \mathbb{E}[\boldsymbol{v}|X] = 0 \tag{13}$$

This makes up a Partially Linear Regression (PLR) model as in (Robinson, 1988). Vector $\boldsymbol{v}$ is N elements long and contains all the disturbance terms. The expression that relates the regressors and the treatment assignment is given by $M(X)$. $M(X)$ is constructed similarly to Γ, as we described in

Section 2.1: $M(X)$ is a vector of functions $\mu(\boldsymbol{x}_i)$. We stress that we are not interested in estimating Γ or M at first hand. The focus is on $\boldsymbol{\theta}$.

Note that the regressors do not correlate with the treatment assignment in a perfect situation. In that case, we could explain the variance in $\boldsymbol{d}$ only by the error terms, so that we would have $\mu(\boldsymbol{x}_i) = 0$ for any value of i.

For cross validation purposes, we use five folds. Furthermore, we choose $R = 100$. For each iteration r, we estimate the model using the following approach:

1. We split the data randomly into two subsets. One subset will be used in Steps 2 and 3, the other one in Step 4.

2. Note that we can rewrite the PLR model as follows:

$$\mathbf{y} = \mathbb{E}[\mathbf{y}|\boldsymbol{d}, X] + \boldsymbol{w} \tag{1 rewritten}$$

$$\boldsymbol{d} = \mathbb{E}[\boldsymbol{d}|X] + \boldsymbol{v} \tag{13 rewritten}$$

so that

$$\hat{\boldsymbol{w}} = \mathbf{y} - \mathbb{E}[\mathbf{y}|\boldsymbol{d}, X] \tag{14}$$

$$\hat{\boldsymbol{v}} = \boldsymbol{d} - \mathbb{E}[\boldsymbol{d}|X] \tag{15}$$

Now we do two estimations:

   (a) We estimate the dependence of the treatment assignment on the regressors. This is the estimation of Equation 1, which results into the residuals in Equation 14.

   (b) We estimate the dependence of the regressors on the dependent variable. This is the estimation of Equation 13, which results into the residuals in Equation 15.

   Any machine learning method can be used for estimations (a) and (b). We use the following methods: Lasso (Tibshirani, 1996), Ridge (Tibshirani, 1996), Elastic net (Zou & Hastie, 2005), Random Forest (Breiman, 2001), Gradient boosting (Friedman, 2001), Neural network (He, Zhang, Ren, & Sun, 2015) and Linear Regression for these estimations. For Lasso, Ridge and Elastic net we do grid search to find the best hyperparameters. Inspired by Syarif, Prugel-Bennett, and Wills (2016), we use 1,000 searches on a logarithmic scale from 0.01 to 10,000.

3. In the last step, we obtain the desired estimator $\hat{\theta}_{(r)}$. We partial out $\boldsymbol{d}$ by regressing $\hat{\boldsymbol{w}}$ on $\hat{\boldsymbol{v}}$. The result is the estimator

$$\hat{\theta}_{(r)} = (\hat{V}'\boldsymbol{d})^{-1}\hat{V}'(\mathbf{y} - \Gamma(X)) \tag{16}$$

   This step is where the "Debiased" comes from in the name of this method. Estimating the treatment effect directly would have given bias, but we removed the bias by above procedure.

Now all values in the vector $\hat{\boldsymbol{\theta}}$ are equal to $\frac{1}{R}\sum_{r=1}^{R}\hat{\theta}_{(r)}$.

Generalization and formalization The elaboration on the PLR model above is an application of the DML method, but the DML method itself is more general. To be more precise, in the PLR we assume linearity (which is in the name), but the DML is so powerful because it is not obliged that the treatment effect has additive impact. Therefore, we generalize the explanation above. We use a score function that is on its turn used in a moment condition. We call it $\psi(\boldsymbol{b}, \boldsymbol{\theta}, \Lambda)$, where $\Lambda = (\Gamma, M)$. ψ orthogonalized score function that is known before solving the moment equations. In the PLR model that we treated above, it holds that

$$\psi(\boldsymbol{b}, \boldsymbol{\theta}, \Lambda) = \{\mathbf{y} - \Gamma(X) - \boldsymbol{\theta} \circ (\boldsymbol{d} - M(X))\} \circ (\boldsymbol{d} - M(X)) \tag{17}$$

Notice that this is a multiplication of both error terms. Now the first condition is the identification condition

$$\mathbb{E}[\psi(\boldsymbol{b}, \boldsymbol{\theta}, \Lambda)] = 0 \tag{18}$$

What we are saying with this identification condition is that the regressors and errors need to be orthogonal to each other. In other words, they need to be independent. For the PLR model, the identification condition is as follows:

$$\mathbb{E}[\{\mathbf{y} - \Gamma(X) - \boldsymbol{\theta} \circ (\boldsymbol{d} - M(X))\}(\boldsymbol{d} - M(X))] = 0 \tag{19}$$

We also wish to satisfy the *Neyman orthogonality condition*, which is that we evaluate the derivative of the moment condition in Equation 18:

$$\partial_\Lambda \, \mathbb{E}[\psi(\boldsymbol{b}, \boldsymbol{\theta}, \Lambda)]\big|_{\Lambda=\Lambda_0} = 0 \tag{20}$$

where Λ_0 is the true value of Λ. This condition expresses that the score function and with that the estimate of $\boldsymbol{\theta}$ should be robust to minor changes of $\Gamma(X)$ and $M(X)$. Inserting the score function of the PLR model, the condition is as follows:

$$\partial_\Lambda \, \mathbb{E}[\{\mathbf{y} - \Gamma(X) - \boldsymbol{\theta} \circ (\boldsymbol{d} - M(X))\}(\boldsymbol{d} - M(X))]\big|_{\Lambda=\Lambda_0} = 0 \tag{21}$$

We insert the machine learning based estimators of the functions $\Gamma(X)$ and $M(X)$. Both approaches generate estimators of $\boldsymbol{\theta}$ by solving the empirical analogues of the moment conditions.

Fully Interactive Model The PLR model can also be modified such that no linearity is assumed:

$$\mathbf{y} = \Gamma(\boldsymbol{d}, X) + \boldsymbol{u} \qquad \mathbb{E}[\boldsymbol{u}|\boldsymbol{d}, X] = 0 \tag{22}$$
$$\boldsymbol{d} = M(X) + \boldsymbol{v} \qquad \mathbb{E}[\boldsymbol{v}|X] = 0 \tag{13 revisited}$$

We call this the fully interactive model, as it enables $\boldsymbol{d}$ to interact completely with X. Classification models are the only models that we can use for estimating the second equation. The average treatment effect is given by $\mathbb{E}[\gamma(1, \boldsymbol{x}_i) - \gamma(0, \boldsymbol{x}_i)]$. Robins and Rotnitzky (1995) propose the score function

$$\begin{aligned}\psi(\boldsymbol{b}, \boldsymbol{\theta}, \Lambda) = [\Gamma(1, X) - \Gamma(0, X)] + \boldsymbol{d}(\mathbf{y} - \Gamma(1, X)) \oslash M(X) \\ - (\mathbf{1} - \boldsymbol{d})(\mathbf{y} - \Gamma(0, X)) \oslash [\mathbf{1} - M(X)] - \boldsymbol{\theta}\end{aligned} \tag{23}$$

One the one hand, a richer interaction between $\boldsymbol{d}$ and X is possible. On the other hand, this model is less explainable.

2.6 Generic Machine Learning

General *Generic Machine Learning* enables us to estimate treatment effects heterogeneously. The program may have had different impacts on distinct groups. Therefore, exploring heterogeneity adds valuable information. Chernozhukov, Demirer, et al. (2018) designed the method, which can handle a large number of regressors. The idea is as follows: we make rough estimations of the individual treatment effect: the 'proxy estimators'. Any machine learning method can be used for this. That is why the method is called *generic*. These estimations are potentially biased and inconsistent, but Chernozhukov, Demirer, et al. argue that they are usable for exploring treatment heterogeneity. Therefore, we use the estimations to estimate a baseline conditional average treatment effect and a heterogeneity parameter. After that, we classify the observations according to the treatment effect. This enables us to investigate differences between groups of observations that responded differently to the treatment.

Approach Often, R is chosen large, for example 100. Therefore, we also choose $R = 100$. For each of the R iterations, we follow these steps:

1. We split the data into two samples, a main sample $\mathcal{M}$ and an auxiliary sample $\mathcal{A}$. This is based on a 50/50 split.

2. We use sample $\mathcal{A}$ to calculate a 'proxy estimator' of the individual treatment effects. For this purpose, we fit two machine learning models that fit $\mathbf{y}$ on X. In the first model, we use a sample that only includes individuals in the treatment group. We denote these estimates by $\mathbb{E}[y_i(1)|\boldsymbol{x}_i]$. In the second model, we use a sample that only includes individuals in the control group. We denote these estimates by $\mathbb{E}[y_i(0)|\boldsymbol{x}_i]$. Just to be clear, treatment assignment is not involved as a regressor in these two models. The individual treatment effects are equal to the difference between the predictions of the two models. Formally written, let
$$\theta_i^I = \mathbb{E}[y_i(1)|\boldsymbol{x}_i] - \mathbb{E}[y_i(0)|\boldsymbol{x}_i] \tag{24}$$
We use different methods to determine which one gives the best fit. More concretely, we use Lasso, Tree, Random Forest, Support Vector Machine, Gradient Boosting, and Linear Regression. For Lasso we do grid search to find the best hyperparameters. Again, we use 1,000 searches on a logarithmic scale from 0.01 to 10,000. In the case of Random Forest, we use 100 trees. The proxy estimator is biased. Nevertheless, Chernozhukov, Chetverikov, et al. show that we can use it for valid inference on key features of the treatment effect, such as the presence of heterogeneity.

3. In this step, we make use of sample $\mathcal{M}$. We make inference on the key features of the treatment effect. To this end, we compute three estimates of interest:

   - To analyse whether there is treatment heterogeneity, we estimate the best linear predictor of the conditional average treatment effect. First, we define
   $$\boldsymbol{t} = \begin{pmatrix} t_1 \\ \dots \\ t_N \end{pmatrix} = \begin{pmatrix} d_1 \\ \dots \\ d_N \end{pmatrix} - \begin{pmatrix} \mathbb{P}(d_1 = 1|\boldsymbol{x}_1) \\ \dots \\ \mathbb{P}(d_N = 1|\boldsymbol{x}_N) \end{pmatrix} \tag{25}$$
   Vector $\boldsymbol{t}$ consists of residualised treatment assignments. We also define
   $$\boldsymbol{s} = \begin{pmatrix} s_1 \\ \dots \\ s_N \end{pmatrix} = \begin{pmatrix} \theta_1^I - \bar{\theta}^I \\ \dots \\ \theta_N^I - \bar{\theta}^I \end{pmatrix} \tag{26}$$

which is the vector with normalised treatment effects. Lastly, we define

$$\boldsymbol{r} = \begin{pmatrix} r_1 \\ \dots \\ r_N \end{pmatrix} = \begin{pmatrix} \mathbb{E}[y_1(0)|\boldsymbol{x}_i] \\ \dots \\ \mathbb{E}[y_N(0)|\boldsymbol{x}_i] \end{pmatrix} \tag{27}$$

which we calculated in Step 2 already. Now the following regression

$$\mathbf{y} = \hat{\alpha}_1 \mathbf{e} + \hat{\alpha}_2 \boldsymbol{r} + \beta_1 \boldsymbol{t} + \beta_2 \boldsymbol{t} \circ \boldsymbol{s} + \hat{\epsilon} \tag{28}$$

For estimation, we use Weighted Least Squares with weights that are given by

$$w(\boldsymbol{x}_i) = \frac{1}{\mathbb{P}(d_i = 1|\boldsymbol{x}_i)\{1 - \mathbb{P}(d_i = 1|\boldsymbol{x}_i)\}} \tag{29}$$

such that $\mathbb{E}[w(\boldsymbol{x}_i)\epsilon_i \boldsymbol{x}_i] = 0$ holds. α_1 is a constant and the term $\alpha_2 \boldsymbol{r}$ is included to improve precision. The interaction $\boldsymbol{t} \circ \boldsymbol{s}$ is orthogonal to $\boldsymbol{t}$ when we estimate using the weights $w(\boldsymbol{x}_i)$. $\boldsymbol{r}$ is included to improve efficiency. The estimator $\hat{\beta}_1$ is an unbiased estimator of the average treatment effect. We can see β_2 as a heterogeneity parameter. To be more specific, it quantifies how well the proxy predictor estimated treatment heterogeneity. Therefore, if there is no heterogeneity, $\beta_2 = 0$.

Subsequently, the individual treatment effects are given by

$$\hat{\boldsymbol{\theta}} = \beta_1 + \beta_2 \boldsymbol{s} \tag{30}$$

such that the best linear predictor of the conditional average treatment effect is equal to $\bar{\boldsymbol{\theta}}$.

- The sorted group average treatment effects. We partition the data into J subsets of equal size. J is often chosen equal to 4 or 5. We choose $J = 5$. We sort the subsets, such that $\mathcal{G}_1$ is the least affected group and $\mathcal{G}_K$ is the most affected group. Now matrix G has dimensions $N \times J$ and each element indicates whether the corresponding individual belongs the corresponding group.

  The target in this step is to find the expected values of the conditional average treatment effects for the groups $j = 1, \dots, J$, which are in the vector

$$\hat{\boldsymbol{\theta}}_{\mathcal{G}} = \begin{pmatrix} \hat{\theta}_{\mathcal{G}_1} \\ \dots \\ \hat{\theta}_{\mathcal{G}_J} \end{pmatrix} = \begin{pmatrix} \mathbb{E}[\theta|\mathcal{G}_1] \\ \vdots \\ \mathbb{E}[\theta|\mathcal{G}_J] \end{pmatrix} \tag{31}$$

  To obtain these, we estimate

$$\mathbf{y} = \hat{\alpha}_1 \mathbf{e} + \hat{\alpha}_2 \boldsymbol{r} + \boldsymbol{t} \circ G \hat{\boldsymbol{\theta}}_{\mathcal{G}} + \nu \tag{32}$$

  with the weights from (29).

- Classification analysis. The group average treatment effects indicate whether there is heterogeneity or not. By performing a classification analysis, we analyse the source of the heterogeneity. We calculate the average value of a regressor k in each group j:

$$\delta_{jk} = \mathbb{E}[\boldsymbol{x}_k|\mathcal{G}_j] = \frac{1}{\#(j)} \sum_{i=1}^{\#(j)} x_{ik} \tag{33}$$

In the results, we refer to this value as 'δ_j for regressor k'. With $\#(j)$, we refer to the number of individuals that belong to group j. Differences between individuals in the least and most affected group are investigated: we calculate $\delta_{5k} - \delta_{1k}$ for all regressors. These values are an indicator for how heterogeneous the treatment is in a specific regressor.

4. We calculate the performance measures

$$\lambda = \hat{\beta}_2^2 \mathrm{Var}(\boldsymbol{\theta}) \tag{34}$$

and

$$\bar{\lambda} = \frac{1}{J} \sum_{j=1}^{J} \hat{\theta}_{\mathcal{G}_j}^2 \tag{35}$$

for each used machine learning method. A high λ means a large variance in the treatment effects, such that a big part of the variance can be explained by the treatment. Therefore, we choose the method with the highest λ. This performance measure is especially important when estimating the best linear predictors. Concerning $\bar{\lambda}$, we choose the highest one as well. This measure is important for estimating the sorted group average treatment effects and the classification analysis.

For all the machine learning methods, we calculate the medians of λ and $\bar{\lambda}$ over the splits to determine which method we should choose.

2.7 Causal Forest

General The *Causal Forest* is especially suitable to estimate causal effects in a Randomized Control Trial with a large amount of data. Its intention is to estimate the individual treatment effects. The treatment assignment should be binary. As this holds for our treatment, we can use the *Causal Forest*.

Setup Before explaining the *Causal Forest*, we dig deeper into the Causal Tree, designed by Athey and Imbens (2016). This is a modified version of the regression tree by Breiman (2001). The algorithm estimates the treatment effect partition-wise. In other words, across the subsets in the partition, the treatment effect varies, but within each subset it is the same. To this end, we take 80% of the sample set and separate into two samples, namely a training sample $\mathcal{S}_T$ with size N_T and an estimation sample $\mathcal{S}_E$ with size N_E. We use $\mathcal{S}_T$ to make a partition of the data and $\mathcal{S}_E$ to estimate the treatment effect within each subset. This idea is called honest splitting. It prevents overfitting.

Analogously to the model we defined in Section 2.1, D_T is the $N_T \times 1$ vector with the treatment assignments of the individuals in $\mathcal{S}_T$ and θ_T is the corresponding $N_T \times 1$ vector with treatment effects. Athey and Imbens (2016) admit that cutting the sample into a training and testing sample results into a loss in precision, but they argue that the benefit in reducing the bias offsets at least part of this cost.

Splitting into partitions We partition the data by iteratively splitting into subsets. This iterative process starts with the full sample $\mathcal{S}_T$. In each iteration, we investigate for one subset whether we should split it up into two subsets or not. The splitting is based on the values of the regressors. There is one requirement: each subset should contain treated and non-treated individuals. We could not calculate the treatment effect elsewise. Therefore, each subset must contain at least twenty-five individuals, of which at least 10% must be in the treatment group and 10% in the control group.

We zoom in on one iteration: a partition with a certain number of subsets has already been made. For one of these subsets $\mathcal{G}_j$, we consider whether and how it should be further split up. To this end, we introduce $\mathbb{P}$, the set of partitions of $\mathcal{G}_j$. Remember that this set includes the $\mathcal{G}_j$ itself and all partitions that split $\mathcal{G}_j$ into two subsets. For each partition in $\mathbb{P}$, we first estimate the treatment effects within the separate subsets. After that, we estimate the Expected Mean Squared Error (EMSE), the criterion to decide which split we choose. In the next two paragraphs, we elaborate on these two steps.

Estimation of the treatment effects The individuals in $\mathcal{S}_E$ are assigned to the leaves. The treatment effect of the observations in subset $\mathcal{G}_j$ is given by

$$\theta_{\mathcal{G}j} = \frac{1}{\#(i \in \mathcal{G}_j : d_i = 1)} \sum_{i \in \mathcal{G}_j : d_i = 1} y_i - \frac{1}{\#(i \in \mathcal{G}_j : d_i = 0)} \sum_{i \in \mathcal{G}_j : d_i = 0} y_i \tag{36}$$

We define matrix G, with dimensions $N \times J$, where each element in the matrix indicates whether the corresponding observation belongs the corresponding group. Plus, we define

$$\boldsymbol{\theta}_{\mathcal{G}} = \begin{pmatrix} \theta_{\mathcal{G}_1} \\ \dots \\ \theta_{\mathcal{G}_J} \end{pmatrix} \tag{37}$$

Now, we see that

$$\boldsymbol{\theta} = G \boldsymbol{\theta}_{\mathcal{G}} \tag{38}$$

Computation of the goodness of fit In decision trees, the Mean Squared Error is used to estimate the goodness of fit of the tree. We can not do that in the case of treatment effects. To clarify this, normally the Mean Squared Error is calculated by summing up the squared differences of predictions and test values. However, we do have a prediction of the treatment effect in each leaf, but we do not have the 'real treatment effect'. Athey and Imbens use the Expected Mean Squared Error. They propose it as follows:

$$EMSE = -\frac{1}{N_T} \boldsymbol{d}_T' \boldsymbol{\theta}_T \boldsymbol{d}_T + \frac{1}{N_T + N_E} \sum_{l \in \Pi} \left(\frac{S^2_{l,treat}}{p} + \frac{S^2_{l,control}}{1-p} \right). \tag{39}$$

Here Π is the set with leaves (based on the training set). $S^2_{l,treat}$ and $S^2_{l,control}$ are the within variances in leave l in the treatment and control group respectively. Note that from Equation 39, we can conclude that the first term rewards high heterogeneity across the leaves and the second term penalizes splits that lead to small leaves.

For each subset in the partition, we estimate the EMSE. The split that maximizes Equation 39 is chosen. No split is made when all splits result into a lower EMSE than when no split is made. The tree is fully grown when no more splits can be made.

Construction of the Causal Forest Trees are easy to interpret, but also easy to misinterpret, because of its high variance. Therefore, a *Causal Forest* is created by growing R trees. We use $R = 1000$. Each time a different training and estimation sample is used. As the number of trees grows, the estimates will be asymptotically normally centered around the true treatment effect. Then, as Equation 3 already showed, the vector with treatment effect is given by

$$\hat{\boldsymbol{\theta}} = \frac{1}{R} \sum_{r=1}^{R} \hat{\boldsymbol{\theta}}_{(r)} \tag{3 revisited}$$

3 Data description

We use data that was collected for the study of Grinstein-Weiss et al. (2013). The central question of the paper is whether providing people with an Individual Development Account (IDA) has a long-term impact on homeownership.

Experimental design The program was administered by the Community Action Program of Tulsa County. All participants rented a house at the moment of start. Everyone was exactly three years in the program. from 2000 till 2003. A Randomized Control Trial was conducted: there was a treatment group with participants that got an IDA and a control group with individuals that did not get an IDA.

At the end of the experiment and six years later, in 2009, it was inventoried which individuals owned a home. (Grinstein-Weiss et al., 2008) examine whether the program had a short-term effect. To be more specific, they estimate whether homeownership had increased during the three years of the program. (Grinstein-Weiss et al., 2013) however investigate whether there is a noticeable influence six years later.

Individual Development Accounts The IDAs were saving accounts for people with a low income. Money that was spent from the saving account was matched with a 1:1 ratio for home repair, small-business investments, post-secondary education and retirement savings. For example, when some money was used on a small-business investment, the spent amount of money was supplemented by the same amount of money by means of the program. Furthermore, there was a match with a 2:1 ratio for home purchase. So, money spent on a home was supplemented with twice the same amount of money. The largest amount that could be matched was $750 per year. As people could match their savings for up to three years, participants could match up to $6,750 if they matched $750 on the purchase of a home in each of the three years. 66% of the participants never made a matched withdrawal from their accounts. Participants had to agree that meanwhile they did not participate in other similar programs.

The goal of providing people an IDA is to stimulate users to save their money or to spend it on useful goods. Before-hand and during the program, participants had to complete a number of trainings on money management, debt reduction and other topics of that kind. Participants needed to have an income below 150% of the federal poverty guideline.

Descriptive statistics At the start of the period in which the individuals got the IDA, there were 863 participants. At the end of the experiment in 2003, 642 participants were left over. Of 652 participants we know whether they owned a home in 2009, which was estimated at this moment. This is because 105 participants in 2003 did not respond in 2009 and 115 participants which did not respond in 2003 did respond in 2009. Table 1 shows the sample sizes specifically for the treatment and control group. Obviously, we can only use the data of the 652 individuals that participated in

2009. When removing individuals that we missed data of, we are left with 604 to run the models on.

	2000 (Start experiment)	*2003 (End experiment)*	*2009 (Count moment)*
Treatment	434	318 (73,3%)	320 (73,7%)
Control	429	324 (75,5%)	332 (77,4%)
Total	863	642 (74,4%)	652 (75,6%)

Table 1: Number of participants in treatment and control group at different moments. The percentages are with respect to the start of the experiment.

Randomization Estimations of treatment effects are only credible if the treatment has the same influence on the treated as on the non-treated. There, the assignment to treatment and control group was fully random. Table 2 shows statistics split out based on the treatment assignment. We highlight the most striking things: treatment group members had more assets, but they also had more debts. This could imply that participants in this group would buy things slightly faster. The difference is not extreme, however. Furthermore, participants in the treatment group seem to be more of the 'white, well-educated men', but also here differences are small. The statistics are like those measured in 2003, at the end of the experiment (Grinstein-Weiss et al., 2013).

	Treatment group	*Control group*
Age	34,238	34,330
Male	0,195	0,160
Married	0,262	0,206
Race (is Caucasian)	0,393	0,428
Education		
High school graduate or less	0,317	0,326
Some college	0,412	0,425
College degree or more	0,271	0,243
Number of adults in household	0,446	0,431
Have children at home	0,822	0,742
Income	1423	1283
Have health insurance	0,597	0,536
Own a business	0,044	0,042
Own other property	0,027	0,023
Own car	0,809	0,810
Live in unsubsidized housing	0,684	0,650
Total assets	5555	4891
Total debts	8912	8479
Satisfied with health	0,862	0,863
Satisfied with financial situation	0,641	0,601

Table 2: Descriptive statistics split out for treatment and control group in 2009.

When applying the methods, we also include regressors to control for unseen non-randomness. They include the following: whether one lives in an unsubsidized home, age, income, total value of assets, total value of debts, education, gender, race, whether married or not, whether one has children, whether one has a bank account, whether one has a health insurance, whether one has a car, a business or another property, whether one has retirement savings, whether one receives welfare payment, whether one is satisfied with his health and with his financial situation, the number of adult in their household. Overall, there are 28 regressors. After dummying the ordinal regressors, we are left with 35 regressors to build the models on.

4 Application

4.1 Traditional models

We create three traditional models: an OLS model, a Probit model and a Logit model. For implementation, we use the package `statsmodels` in `Python`. Table 4 in the replication paper shows the treatment effect that Grinstein-Weiss et al. obtain with OLS. We replicate this and use it as a reference.

	Coefficient	*P-value*	*95% confidence interval*
OLS	0.006 (0.040)	0.872	[-0.072, 0.085]
Probit	0.023 (0.111)	0.833	[-0.194, 0.241]
Logit	0.036 (0.182)	0.845	[-0.321, 0.392]

Table 3: Treatment effects for OLS, Probit and Logit.
None of the coefficients is significant.

	OLS	*Probit*	*Logit*
const	insignificant	-1.639 (0.409)	-2.665 (0.676)
unsubsidized	0.135** (0.047)	0.377 (0.131)	0.621 (0.216)
bin_age_u17	-0.099* (0.042)	-0.276 (0.118)	-0.452 (0.194)
own_bank_u17	0.139* (0.055)	0.410 (0.160)	0.675 (0.267)
own_scale2_u17	0.031** (0.010)	0.089 (0.027)	0.143 (0.045)
bin_sat_heal_u17	0.194** (0.059)	0.577 (0.174)	0.949 (0.294)
tri_ed_u17_2.0	0.117* (0.058)	0.346 (0.160)	0.547 (0.262)

Table 4: Significant regressors for OLS, Probit and Logit.
* is significant at 5% level and ** is significant at 1% level.

Table 3 presents the treatment effects for OLS as well as the Logit and the Probit model. None of the models gives a treatment effect that differs significantly from zero. It seems that participating in the program did not cause more people to own a home on the long-term. The Probit and Logit model estimators will be more consistent due to the binary nature of the dependent variable, but their standard errors are slightly larger than those of OLS. Table 4 shows the regressors that are significant in the three models. The same regressors are significant, as well at the 5% level as at the 1% level for the three different models. We show the regression results of the Probit and Logit model in Tables C.2 and C.3. Extensive results can be found in the Appendix, in Table C.1.

4.2 Double Machine Learning

Table 5 presents the results of the DML models. We obtain them by making use of the `Python` package `doubleml`. We make use of six and three different machine learning models in the Partially Linear Regression Model and the Fully Interactive Model respectively.

	Coefficient	*P-value*	*95% confidence interval*
Partially Linear Regression Model			
Lasso	0.005 (0.039)	0.893	[-0.071, 0.082]
Ridge	0.006 (0.040)	0.885	[-0.071, 0.083]
Elastic net	0.009 (0.039)	0.815	[-0.068, 0.086]
Random forest	0.004 (0.041)	0.914	[-0.075, 0.084]
Gradient boosting	0.010 (0.043)	0.821	[-0.074, 0.094]
Neural network	0.010 (0.060)	0.861	[-0.107, 0.128]
Linear regression	0.003 (0.042)	0.936	[-0.079, 0.086]
Fully Interactive Model			
Random forest	0.012 (0.042)	0.775	[-0.070, 0.094]
Gradient boosting	0.095 (0.083)	0.252	[-0.068, 0.258]
Neural network	0.094 (0.126)	0.457	[-0.153, 0.340]

Table 5: Results of the DML models with the use of different machine learning algorithms. None of the models give a statistically significant treatment effect.

The coefficients in the PLR model are similar with OLS. All of them are insignificant. Plus, the standard errors are of the same order of magnitude. However, the coefficients in the Fully Interactive (FI) model are slightly higher. Nonetheless, their standard errors are slightly larger as well, so the coefficients are insignificant as well. The question that follows is why the treatment effects are larger in the FI model than in the PLR model. In the FI model, non-linear relations between D and X are allowed. When including these relations leads to a higher treatment effect, this means that part of the treatment effect depends on the regressors.

4.3 Generic Machine Learning

For *Generic Machine Learning*, we run the model with the use of six different machine learning models. We use the `R`-package `GenericML` for implementation. The resulting performance measures are shown in Table 6. For λ, the Support Vector Machine turns out to have the highest value, so when the best linear predictor of the conditional average treatment effect must be estimated, this algorithm turns out to have the best performance. For $\bar{\lambda}$, the Linear Regression has the highest value. Therefore, when it comes to estimating the group average treatment effect, this is the best algorithm to use.

	λ	$\bar{\lambda}$
Lasso	0.0011	0.0125
Tree	0.0012	0.0131
Random Forest	0.0009	0.0130
Support Vector Machine	0.0013	0.0131
Gradient Boosting	0.0012	0.0126
Linear Regression	0.0006	0.0132

Table 6: Performance measures when using different machine learning algorithms

The coefficients using the Support Vector Machine are shown in Table 7. In contrast to what we saw in the case of DML, a heterogeneity parameter is involved here. Both coefficients turn out to be highly insignificant.

	Coefficient	*P-value*	*95% confidence interval*
Baseline conditional average	0.016 (0.065)	0.780	[-0.113, 0.143]
Heterogeneity	-0.008 (0.440)	0.983	[-0.869, 0.872]

Table 7: Best linear predictor coefficients using Support Vector Machine in *Generic Machine Learning*

We dive deeper into the heterogeneity. Table 8 shows the average treatment effect groupwise. The first group is least affected (even negatively), the fifth group has the largest treatment effect. We see differences, but even treatment effects in the least and most affected groups are insignificant. Obviously, the standard errors are larger as the ones in earlier models because the subgroups contain less individuals.

	Coefficient	*P-value*	*95% confidence interval*
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_1}}$	-0.016 (0,144)	0.897	[-0.299, 0.266]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_2}}$	-0.004 (0,145)	0.974	[-0.288, 0.280]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_3}}$	0.018 (0,145)	0.868	[-0.266, 0.302]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_4}}$	0.035 (0,144)	0.772	[-0.248, 0.318]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_5}}$	0.054 (0,144)	0.664	[-0.228, 0.336]

Table 8: Sorted group average treatment effects obtained by *Generic Machine Learning* when made use of Linear Regression

Although in general there is no treatment heterogeneity, it may be there in specific regressors. We investigate this. It turns out to be there in several regressors, namely income, race, marital status, number of adults in the household, experience with difficulties with income, satisfaction with health, involvement in the community, liabilities and followed education. In the Appendix, in Tables C.11 till C.22, we show the results for all these regressors.

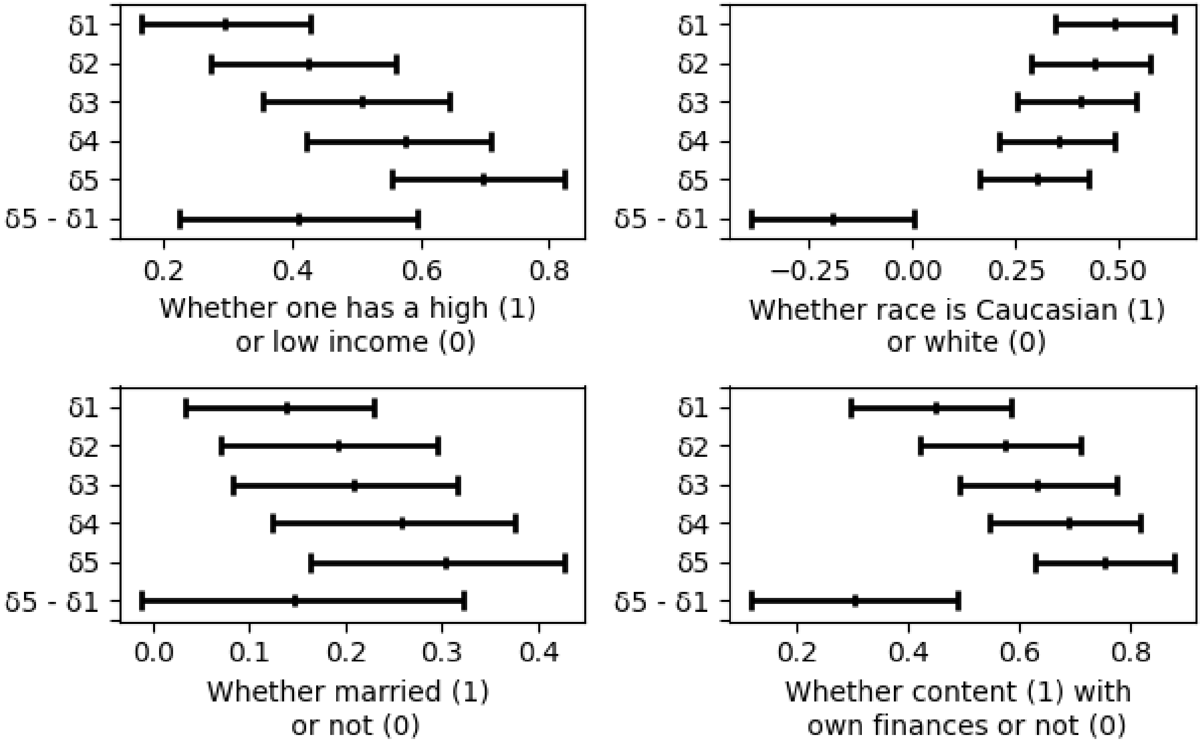

Figure 2: Classification analysis of four regressors. δ_1 indicates the coefficient for the 20% of the data that was least affected by the treatment. So is δ_5 for the most affected group. $\delta_1 - \delta_5$ indicates the heterogeneity that is caused by the concerning regressor.

The most striking regressors that cause heterogeneity are presented in Figure 2. There are differences in the treatment effect for married, white people with a higher income, that were content with their financial situation. For race and marital status, these differences are insignificant. However, heterogeneity is proved to be significant for income and satisfaction with the financial situation.

We investigate how Grinstein-Weiss et al. coped with heterogeneity in the treatment effects and what they found. Table 9 shows the results of an OLS regression in which interaction parameters were added. In the literature this is quite common, besides running regressions on a couple of subgroups and comparing them to each other (Słoczyński, 2022). This allows the treatment to differ by subsample. Only those individuals which had a higher income turn out to be long-term influenced by the program. Satisfaction about the financial situation is not included as an interaction term in this regression.

	Coefficient	*P-value*
Treatment	-0.106 (0.186)	0.568
Treatment × **higher income**	0.167* (0.084)	0.046
Treatment × **female**	0.062 (0.118)	0.597
Treatment × **Caucasian**	-0.101 (0.086)	0.244
Treatment × **married**	0.112 (0.108)	0.299
Treatment × **some college**	-0.014 (0.096)	0.887
Treatment × **college graduate**	0.120 (0.106)	0.256
Treatment × **children**	-0.050 (0.102)	0.624
Treatment × **cohort**	0.068 (0.092)	0.462
Treatment × **banked**	-0.041 (0.105)	0.697
Treatment × **welfare**	0.045 (0.089)	0.615
Treatment × **car**	0.009 (0.105)	0.933
Treatment × **insurance**	0.007 (0.082)	0.931

Table 9: Interaction effects in the replication paper

4.4 Causal Forest

For implementing the *Causal Forest*, we use the `Python`-package `econml`. The individual treatment effects are the most important result. Figure 3 shows the estimated individual treatment effects for all individuals ordered. It shows that the treatment is heterogeneous. Approximately half of the individuals has a positive treatment effect and half of them has a negative treatment effect. Each subset contains at least twenty-five individuals. The higher this value, the more the individual treatment effects approach the average treatment effect. Table 10 shows that the average treatment effect is estimated to be 0.003, which is a similar value as obtained with OLS. The treatment effect is not significantly different from zero, so an effect of the treatment can not be identified. This means that again we a clue that the program with the IDAs did not have a long-term effect.

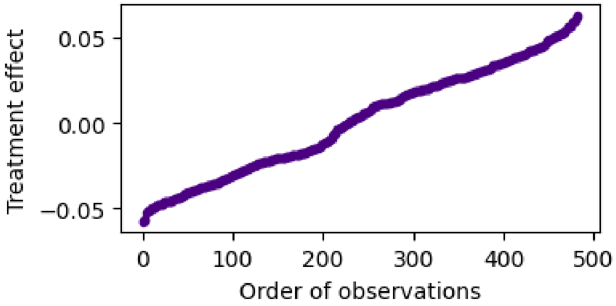

Figure 3: Sorted individual treatment effects, obtained by the Causal Forest

Coefficient	0.003 (0.014)
P-value	0.805
95% confidence interval	[-0.023, 0.031]

Table 10: Average treatment effect, obtained by the *Causal Forest*

5 Conclusion

Our goal was to analyse three causal machine learning methods: *Double Machine Learning*, *Generic Machine Learning* and the *Causal Forest*. We have written them into one econometric form. *Double Machine Learning* is useful when estimating the average treatment effect and *Generic Machine*

Learning when investigating the heterogeneity. The *Causal Forest* can be used particularly to estimate the individual treatment effects.

We applied these methods onto an earlier study in which OLS was used to answer a causal question. The causal machine learning methods often give similar results when it comes to the average treatment effect. This did not add value to traditional methods.

Added value came more when regarding the heterogeneity in the treatment effect. Especially GML gave insight in its forms and sources. (Grinstein-Weiss et al., 2013) found sources of heterogeneity by adding interaction terms concerning subgroups in the OLS regression. *Causal Forest* and especially GML give insight in the heterogeneity ex-post, which is more credible.

Using more methods to answer the same question always adds value, because this combines the strengths of more methods. Therefore, it would be good when the investigated methods would be used more often, and we advise researchers to use them. Moreover, the advantage of causal machine learning in any case is that no assumptions have to be done. This makes them widely useable. Especially when it comes to tracing the source of heterogeneity, *Generic Machine Learning* can give useful insights. We must admit, however, that we did not find very striking things with the modern methods, which were not found by OLS. Overall, we advise anyone to use the methods to answer causal questions, but they must not expect that they always gain new insights.

Concerning the causal question, the evidence that Grinstein-Weiss et al. found can now be even called stronger: the program with the IDAs did not have long-term impact. We created three traditional models and three machine learning models. All of them point to this direction.

Further research could especially be done to find out what the effect is of binary nature of the dependent variable in the machine learning methods. The standard errors may not be robust in this case. Plus, similar studies should be done to find out advantages and disadvantages of the methods in other contexts.

References

Angrist, J. D., Imbens, G. W., & Rubin, D. B. (1996). Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, *91*(434), 444–455.

Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353–7360.

Baiardi, A., & Naghi, A. (2020). The value added of machine learning to causal inference: Evidence from revisited studies.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., & Newey, W. (2017). Double/debiased/neyman machine learning of treatment effects. *American Economic Review*, *107*(5), 261–65.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., & Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, *21*(1), C1–C68.

Chernozhukov, V., Demirer, M., Duflo, E., & Fernandez-Val, I. (2018). *Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in india* (Tech. Rep.). National Bureau of Economic Research.

Davis, J., & Heller, S. B. (2017). Using causal forests to predict treatment heterogeneity: An application to summer jobs. *American Economic Review*, *107*(5), 546–50.

Deke, J. (2014). *Using the linear probability model to estimate impacts on binary outcomes in randomized controlled trials* (Tech. Rep.). Mathematica Policy Research.

Deryugina, T., Heutel, G., Miller, N. H., Molitor, D., & Reif, J. (2019). The mortality and medical costs of air pollution: Evidence from changes in wind direction. *American Economic Review*, *109*(12), 4178–4219.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Grinstein-Weiss, M., Lee, J.-S., Greeson, J. K., Han, C.-K., Yeo, Y. H., & Irish, K. (2008). Fostering low-income homeownership through individual development accounts: A longitudinal, randomized experiment. *Housing Policy Debate*, *19*(4), 711–739.

Grinstein-Weiss, M., Sherraden, M., Gale, W. G., Rohe, W. M., Schreiner, M., & Key, C. (2013). Long-term impacts of individual development accounts on homeownership among baseline renters: Follow-up evidence from a randomized experiment. *American Economic Journal: Economic Policy*, *5*(1), 122–45.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the ieee international conference on computer vision* (pp. 1026–1034).

Robins, J. M., & Rotnitzky, A. (1995). Semiparametric efficiency in multivariate regression models with missing data. *Journal of the American Statistical Association*, *90*(429), 122–129.

Robinson, P. M. (1988). Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, 931–954.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, *66*(5), 688.

Sherraden, M., & Gilbert, N. (2016). *Assets and the poor: New american welfare policy*. Routledge.

Sherraden, M. S., & McBride, A. M. (2010). *Striving to save: Creating policies for financial security of low-income families*. University of Michigan Press.

Słoczyński, T. (2022). Interpreting ols estimands when treatment effects are heterogeneous: Smaller groups get larger weights. *Review of Economics and Statistics*, *104*(3), 501–509.

Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). Svm parameter optimization using grid search and genetic algorithm to improve classification performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *14*(4), 1502–1509.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288.

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242.

Zou, H., & 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.

A Derivations

Below we give an overview of the formulas that we take from other papers and how we translated them into a form that suites this thesis. Sometimes notation in other papers is different from our notation. To keep things clear, sometimes we use a slightly different notation than in the original paper, but we never change the form.

A.1 Double Machine Learning

The Partially Linear Model We have rewritten the Partially Linear Regression (PLR) model. Chernozhukov, Chetverikov, et al. (2018) write it as follows:

$$y_i = \theta_0 d_i + g_0(\boldsymbol{x}_i) + u_i \qquad \mathbb{E}[u_i|d_i, \boldsymbol{x}_i] = 0 \tag{40}$$

$$d_i = m_0(\boldsymbol{x}_i) + v_i \qquad \mathbb{E}[v_i|\boldsymbol{x}_i] = 0 \tag{41}$$

Now the matrix notation, as can be seen in Section 2.5, is as follows:

$$\begin{aligned}\boldsymbol{y} = \begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix} &= \begin{pmatrix} \theta_0 d_1 \\ \dots \\ \theta_0 d_N \end{pmatrix} + \begin{pmatrix} g_0(\boldsymbol{x}_1) \\ \dots \\ g_0(\boldsymbol{x}_N) \end{pmatrix} + \begin{pmatrix} u_1 \\ \dots \\ u_N \end{pmatrix} \\ &\equiv \boldsymbol{\theta} \circ \boldsymbol{d} + \Gamma(X) + \boldsymbol{u} \qquad \mathbb{E}[\boldsymbol{u}|\boldsymbol{d}, X] = 0 \qquad \text{(1 revisited)} \\ \boldsymbol{d} = \begin{pmatrix} d_1 \\ \dots \\ d_N \end{pmatrix} &= \begin{pmatrix} m_0(\boldsymbol{x}_1) \\ \dots \\ m_0(\boldsymbol{x}_N) \end{pmatrix} + \begin{pmatrix} v_1 \\ \dots \\ v_N \end{pmatrix} \\ &\equiv M(X) + V \qquad \mathbb{E}[\boldsymbol{v}|X] = 0 \qquad \text{(13 revisited)}\end{aligned}$$

Note that in the DML algorithm, we estimate the average treatment effects, so $\boldsymbol{\theta}$ is a vector with only one different value.

DML estimator In Equation 16, the DML estimator is given. In Chernozhukov, Chetverikov, et al. (2018), this estimator was given by

$$\hat{\theta}_0 = \Big(\frac{1}{N}\sum_{i \in I} \hat{v}_i d_i\Big)^{-1} \frac{1}{N}\sum_{i \in I} \hat{v}_i (y_i - \hat{g}_0(\boldsymbol{x}_i)) \tag{42}$$

We do not use the subscript naught. Rewritten to matrix from, we write

$$\begin{aligned}\hat{\theta} &= \left[\begin{pmatrix} v_1 & \dots & v_N \end{pmatrix} \begin{pmatrix} d_1 \\ \dots \\ d_N \end{pmatrix}\right]^{-1} \begin{pmatrix} v_1 & \dots & v_N \end{pmatrix} \left[\begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix} - \begin{pmatrix} g_0(\boldsymbol{x}_1) \\ \dots \\ g_0(\boldsymbol{x}_N) \end{pmatrix}\right] \\ &= (\hat{V}'\boldsymbol{d})^{-1}\hat{V}'(\boldsymbol{y} - \Gamma(X)) \qquad \text{(16 revisited)}\end{aligned}$$

Generalization and formalization According to the explanation at page 8 and 9 of Chernozhukov, Chetverikov, et al. (2018), the score function corresponding to the PLR model is

$$\phi(b_i; \theta_0, g) = (y_i - \theta_0(d_i - m_0(\boldsymbol{x}_i)) - g(\boldsymbol{x}_i))(d_i - m_0(\boldsymbol{x}_i)) \tag{43}$$

Remember that b_i is the triple $(y_i, \boldsymbol{x}_i, d_i)$. This score function is used to estimate the moment condition

$$\frac{1}{N}\sum_{i\in I}\phi(b_i;\theta_0,\hat{g}_0)=0 \tag{44}$$

We write Equation 43 to matrix notation, which gives the following:

$$\begin{aligned}
\psi(\boldsymbol{b},\boldsymbol{\theta},\Lambda) &= \begin{pmatrix}\phi(b_1;\theta_0,g)\\ \dots\\ \phi(b_N;\theta_0,g)\end{pmatrix}\\
&= \left(\begin{pmatrix}y_1\\ \dots\\ y_N\end{pmatrix} - \theta_0\left(\begin{pmatrix}d_1\\ \dots\\ d_N\end{pmatrix} - \begin{pmatrix}m_0(\boldsymbol{x}_1)\\ \dots\\ m_0(\boldsymbol{x}_N)\end{pmatrix}\right) - \begin{pmatrix}g(\boldsymbol{x}_1)\\ \dots\\ g(\boldsymbol{x}_N)\end{pmatrix}\right)\circ\left(\begin{pmatrix}d_1\\ \dots\\ d_N\end{pmatrix} - \begin{pmatrix}m_0(\boldsymbol{x}_1)\\ \dots\\ m_0(\boldsymbol{x}_N)\end{pmatrix}\right)\\
&= \{\boldsymbol{y}-\Gamma(X)-\boldsymbol{\theta}\circ(\boldsymbol{d}-M(X))\}\circ(\boldsymbol{d}-M(X))
\end{aligned} \tag{17 revisited}$$

with the equivalent moment condition of Equation 44

$$\mathbb{E}[\psi(\boldsymbol{b},\boldsymbol{\theta},\Lambda)]=0 \tag{18 revisited}$$

The Neyman orthogonality condition is given in Equation 1.8 in Chernozhukov, Chetverikov, et al. (2018) and it is as follows:

$$\partial_\eta\,\mathbb{E}\,\phi(b_i;\theta_0,\eta_0)[\eta-\eta_0]=0 \tag{45}$$

where $\eta=(m,g)$. We can write Equation 45 as

$$\partial_\eta\,\mathbb{E}\,\phi(b_i;\theta_0,\eta_0)_{\eta=\eta_0}=0 \tag{46}$$

Now the step to the matrix notation is just small:

$$\partial_\Lambda\,\mathbb{E}[\psi(\boldsymbol{b},\boldsymbol{\theta},\Lambda)]\big|_{\Lambda=\Lambda_0}=0 \tag{20 revisited}$$

Fully interactive model We use the Fully Interactive Model as well to estimate the average treatment effect, as given in Equation 23. This comes from Equation 5 in Chernozhukov et al. (2017):

$$\phi(b_i,\theta,\eta)=g(1,\boldsymbol{x}_i)-g(0,\boldsymbol{x}_i)+\frac{d_i(y_i-\gamma(1,\boldsymbol{x}_i))}{\mu(\boldsymbol{x}_i)}-\frac{(1-d_i)(y_i-\gamma(0,\boldsymbol{x}_i))}{1-\mu(\boldsymbol{x}_i)}-\theta \tag{47}$$

The matrix notation is as follows:

$$\begin{aligned}
\psi(\boldsymbol{b},\boldsymbol{\theta},\Lambda) &= \begin{pmatrix}\phi(b_1;\theta_0,g)\\ \dots\\ \phi(b_N;\theta_0,g)\end{pmatrix}\\
&= \begin{pmatrix}\gamma(1,\boldsymbol{x}_1)\\ \dots\\ \gamma(1,\boldsymbol{x}_N)\end{pmatrix} - \begin{pmatrix}\gamma(0,\boldsymbol{x}_1)\\ \dots\\ \gamma(0,\boldsymbol{x}_N)\end{pmatrix} + \begin{pmatrix}d_1\\ \dots\\ d_N\end{pmatrix}\begin{pmatrix}(y_1-\gamma(1,\boldsymbol{x}_1))/\mu(\boldsymbol{x}_1)\\ \dots\\ (y_N-\gamma(1,\boldsymbol{x}_N))/\mu(\boldsymbol{x}_N)\end{pmatrix}\\
&\quad - \begin{pmatrix}1-d_1\\ \dots\\ 1-d_N\end{pmatrix}\begin{pmatrix}(y_1-\gamma(0,\boldsymbol{x}_1))/(1-\mu(\boldsymbol{x}_1))\\ \dots\\ (y_N-\gamma(0,\boldsymbol{x}_N))/(1-\mu(\boldsymbol{x}_N))\end{pmatrix} - \boldsymbol{\theta}\\
&= [\Gamma(1,X)-\Gamma(0,X)]+\boldsymbol{d}(\boldsymbol{y}-\Gamma(1,X))\oslash M(X)\\
&\quad -(1-\boldsymbol{d})(\boldsymbol{y}-\Gamma(0,X))\oslash[1-M(X)]-\boldsymbol{\theta}
\end{aligned} \tag{23 revisited}$$

A.2 Generic Machine Learning

Baseline definitions Chernozhukov, Demirer, et al. (2018) define the baseline conditional average

$$b_0(\boldsymbol{x}_i) = \mathbb{E}(y_i | d_i = 0, \boldsymbol{x}_i) \tag{48}$$

in Equation 2.1 in their paper and the conditional average treatment effect

$$s_0(\boldsymbol{x}_i) = \mathbb{E}(y_i | d_i = 1, \boldsymbol{x}_i) - \mathbb{E}(y_i | d_i = 0, \boldsymbol{x}_i) \tag{49}$$

in Equation 2.7 in their paper.

Best linear predictors At page 11 of the article of Chernozhukov, Demirer, et al. (2018), we find that the best linear predictor of the conditional average treatment effect is estimated by the following regression:

$$y_i = \hat{\alpha}' R_i + \hat{\beta}_1 (d_i - p(\boldsymbol{x}_i)) + \hat{\beta}_2 (d_i - p(\boldsymbol{x}_i))(s(\boldsymbol{x}_i) - \mathbb{E}_{N,M}\, \hat{s}_0(\boldsymbol{x}_i)) + \hat{\epsilon}_i \qquad \mathbb{E}_{N,M}[w(\boldsymbol{x}_i)\hat{\epsilon}_i R_i] = 0 \tag{50}$$

where $p(\boldsymbol{x}_i) = \mathbb{P}(d_i = 1 | \boldsymbol{x}_i)$. $\hat{s}_0(\boldsymbol{x}_i)$ is the proxy estimator of the treatment effect, which we called θ_i^I in Equation 24. Chernozhukov, Demirer, et al. (2018) propose to use

$$R_i = \begin{pmatrix} 1 \\ \mathbb{E}(y_i | d_i = 0, \boldsymbol{x}_i) \end{pmatrix} \tag{51}$$

Therefore, we rewrite Equation 50:

$$y_i = \hat{\alpha}_1 + \hat{\alpha}_2\, \mathbb{E}(y_i | d_i = 0, \boldsymbol{x}_i) + \hat{\beta}_1 (d_i - p(\boldsymbol{x}_i)) + \hat{\beta}_2 (d_i - p(\boldsymbol{x}_i))(s(\boldsymbol{x}_i) - \mathbb{E}_{N,M}\, \hat{s}_0(\boldsymbol{x}_i)) + \hat{\epsilon}_i \tag{50 rewritten}$$

It follows that

$$\begin{aligned}
\boldsymbol{y} &= \begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix} \\
&= \hat{\alpha}_1 \mathbf{e} + \hat{\alpha}_2 \begin{pmatrix} \mathbb{E}(y_1 | d_1 = 0, \boldsymbol{x}_1) \\ \dots \\ \mathbb{E}(y_N | d_N = 0, \boldsymbol{x}_N) \end{pmatrix} + \hat{\beta}_1 \begin{pmatrix} d_1 - \mathbb{P}(d_1 = 1 | \boldsymbol{x}_1) \\ \dots \\ d_N - \mathbb{P}(d_N = 1 | \boldsymbol{x}_N) \end{pmatrix} \\
&\quad + \hat{\beta}_2 \begin{pmatrix} d_1 - \mathbb{P}(d_1 = 1 | \boldsymbol{x}_1) \\ \dots \\ d_N - \mathbb{P}(d_N = 1 | \boldsymbol{x}_N) \end{pmatrix} \circ \begin{pmatrix} \theta_1 - \bar{\theta} \\ \dots \\ \theta_N - \bar{\theta} \end{pmatrix} + \hat{\epsilon} \\
&= \hat{\alpha}_1 \mathbf{e} + \hat{\alpha}_2 \boldsymbol{r} + \beta_1 \boldsymbol{t} + \beta_2 \boldsymbol{t} \circ \boldsymbol{s} + \hat{\epsilon}
\end{aligned} \tag{28 revisited}$$

Sorted group average treatment effects The regression in Equation 32 comes from Equation 3.3 by Chernozhukov et al. (2017):

$$y_i = \alpha' R_i + \sum_{k=1}^{K} \gamma_k (d_i - p(\boldsymbol{x}_i)) \cdot I(G_{ik}) + \nu_i \tag{52}$$

$$= \alpha' R_i + (d_i - p(\boldsymbol{x}_i)) \sum_{k=1}^{K} \gamma_k I(G_{ik}) + \nu_i \tag{53}$$

$I(G_{ik})$ is an indicator function that equals 1 if an individual i is in $\mathcal{G}_k$. Let us put this into matrix form:

$$\begin{aligned}
\boldsymbol{y} &= \begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix} \\
&= \hat{\alpha}_1 \mathbf{e} + \hat{\alpha}_2 \begin{pmatrix} \mathbb{E}(y_1|d_1 = 0, \boldsymbol{x}_1) \\ \dots \\ \mathbb{E}(y_N|d_N = 0, \boldsymbol{x}_N) \end{pmatrix} + \begin{pmatrix} d_1 - p(\boldsymbol{x}_1) \\ \dots \\ d_N - p(\boldsymbol{x}_N) \end{pmatrix} \circ \begin{pmatrix} \sum_{k=1}^{K} \gamma_k I(G_{1k}) \\ \dots \\ \sum_{k=1}^{K} \gamma_k I(G_{Nk}) \end{pmatrix} + \begin{pmatrix} \nu_1 \\ \dots \\ \nu_N \end{pmatrix} \\
&= \hat{\alpha}_1 \mathbf{e} + \hat{\alpha}_2 \boldsymbol{r} + \boldsymbol{t} \circ G \boldsymbol{\theta}_{\mathcal{G}} + \nu \qquad \text{(32 revisited)}
\end{aligned}$$

A.3 Causal Forest

Athey and Imbens (2016) propose to use the Expected Mean Squared Error in their Section 3.2 ”Modifying the Honest Approach” as follows:

$$EMSE = -\frac{1}{N_T} \sum_{i \in \mathcal{S}_T} \hat{\theta}_i^2 + \frac{1}{N_T + N_E} \sum_{l \in \Pi} \left(\frac{S_{l,treat}^2}{p} + \frac{S_{l,control}^2}{1-p} \right) \tag{54}$$

As $\sum_{i \in \mathcal{S}_T} \hat{\theta}_i^2 = \boldsymbol{\theta}_T' \boldsymbol{\theta}_T$, matrix notation is as follows:

$$EMSE = -\frac{1}{N_T} \boldsymbol{\theta}_T' \boldsymbol{\theta}_T + \frac{1}{N_T + N_E} \sum_{l \in \Pi} \left(\frac{S_{l,treat}^2}{p} + \frac{S_{l,control}^2}{1-p} \right) \qquad \text{(39 revisited)}$$

B Variables

Variable name	**Meaning**
own_home_u42	*Whether he possessed a home in 2009 (dependent variable)*
treat	*Whether he belonged to the treatment group*
const	*Constant*
unsubsidized	*Whether he lived in an unsubsidized home*
bin_age_u17	*Age*
hiinc	*Whether he has an income higher than the 50th percentile*
female_u17	*Whether he is female*
race_cau_u17	*Whether his race is Caucasian*
married_u17	*Whether he is married*
own_bank_u17	*Whether he had an own bank account*
bin_cohort	*Late survey cohort at baseline*
ins_heal_u17	*Whether he had a health insurance*
hh_adult_u17	*The number of adults in his household*
bin_child_u17	*Whether he had children*
own_bus_u17	*Whether he owned a business*
own_prop_u17	*Whether he owned a rental property or another real estate*
own_ira_u17	*Whether he had a retirement account*
src_welf_u17	*Whether he received welfare payment*
own_car_u17	*Whether he had a car*
own_scale2_u17	*Ownership of household goods*
str_scale2_u17	*Measure of the difficulty he experiences to live with his income*
gv_scale2_u17	*How he gives help in the community on a scale 0 - 1*
gt_scale2_u17	*How he gives help in the community on a scale 0 - 1*
bin_sat_heal_u17	*Whether he was satisfied with his health*
bin_sat_fin2_u17	*Whether he was satisfied with his financial situation*
ci_scale_u17	*How involved he is in the community on a scale 0 - 1*
cat_ass_tot_0.0	*Assets: When his assets are worth less than $1,421*
cat_ass_tot_1.0	*Assets: When his assets are worth between $1,422 and $2,842*
cat_ass_tot_2.0	*Assets: When his assets are worth between $2,843 and $4,263*
cat_ass_tot_3.0	*Assets: When his assets are worth $4,264 and up*
cat_ass_tot_missing	*Assets: Value missing*
cat_lib_tot_0.0	*Liabilities: When his liabilities are worth less than $1,421*
cat_lib_tot_1.0	*Liabilities: When his liabilities are worth between $1,422 and $2,842*
cat_lib_tot_2.0	*Liabilities: When his liabilities are worth between $2,843 and $4,263*
cat_lib_tot_3.0	*Liabilities: When his liabilities are worth $4,264 and up*
cat_lib_tot_missing	*Liabilities: Value missing*
tri_ed_u17_0.0	*Education: When he has a high school graduate or less*
tri_ed_u17_1.0	*Education: When he followed some college*
tri_ed_u17_2.0	*Education: When he has some college degree or more*

Table B.1: Variables that are used in the models

C Additional tables

* is significant at 5% level, ** is significant at 1% level and *** is significant at 0.1% level.

C.1 Ordinary Least Squares

Dependent variable:	own_home_u42	**F-statistic:**	2.475
Number of observations:	604	**Loglikelihood:**	-391.65
R-squared:	0.129	**AIC:**	853.3
Adjusted R-squared:	0.077	**BIC:**	1007.0

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
const	-0.0608	0.143	-0.424	0.672	-0.343	0.221
treat	0.0065	0.040	0.161	0.872	-0.072	0.085
unsubsidized	0.1351**	0.047	2.864	0.004	0.042	0.228
bin_age_u17	-0.0985*	0.042	-2.328	0.020	-0.182	-0.015
hiinc	-0.0306	0.043	-0.702	0.483	-0.116	0.055
female_u17	0.0691	0.062	1.123	0.262	-0.052	0.190
race_cau_u17	0.0174	0.044	0.399	0.690	-0.068	0.103
married_u17	0.0639	0.058	1.104	0.270	-0.050	0.178
own_bank_u17	0.1386*	0.055	2.502	0.013	0.030	0.247
bin_cohort	-0.0112	0.046	-0.246	0.806	-0.101	0.078
ins_heal_u17	-0.0055	0.042	-0.130	0.897	-0.089	0.078
hh_adult_u17	-0.0138	0.034	-0.411	0.681	-0.080	0.052
bin_child_u17	0.0046	0.053	0.088	0.930	-0.099	0.108
own_bus_u17	-0.0161	0.101	-0.160	0.873	-0.214	0.182
own_prop_u17	-0.0648	0.128	-0.507	0.613	-0.316	0.187
src_welf_u17	0.0073	0.047	0.156	0.876	-0.085	0.099
own_car_u17	-0.0457	0.062	-0.739	0.460	-0.167	0.076
own_scale2_u17	0.0312**	0.010	3.155	0.002	0.012	0.051
str_scale2_u17	0.0620	0.091	0.680	0.497	-0.117	0.241
gv_scale2_u17	-0.0542	0.123	-0.440	0.660	-0.296	0.188
bin_sat_heal_u17	0.1944**	0.059	3.268	0.001	0.078	0.311
bin_sat_fin2_u17	-0.0514	0.048	-1.072	0.284	-0.146	0.043
ci_scale_u17	0.1012	0.100	1.013	0.312	-0.095	0.297
cat_ass_tot_1.0	0.1240	0.070	1.774	0.077	-0.013	0.261
cat_ass_tot_2.0	0.0752	0.074	1.018	0.309	-0.070	0.220
cat_ass_tot_3.0	0.1176	0.064	1.848	0.065	-0.007	0.243
cat_ass_tot_missing	-0.0342	0.079	-0.435	0.664	-0.189	0.120
cat_lib_tot_1.0	-0.0057	0.078	-0.073	0.942	-0.158	0.147
cat_lib_tot_2.0	-0.0054	0.083	-0.065	0.948	-0.169	0.158
cat_lib_tot_3.0	-0.0488	0.055	-0.892	0.373	-0.156	0.059
cat_lib_tot_missing	-0.0504	0.066	-0.762	0.446	-0.180	0.080
tri_ed_u17_1.0	-0.0021	0.048	-0.044	0.965	-0.096	0.092
tri_ed_u17_2.0	0.1174*	0.058	2.025	0.043	0.004	0.231

Table C.1: OLS Regression Results

C.2 Probit model

Dependent variable:	own_home_u42	**Pseudo R-squared:**	0.1017
Number of observations:	604	**Loglikelihood:**	-371.57

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
const	-1.6390**	0.409	-4.004	0.000	-2.441	-0.837
treat	0.0234	0.111	0.211	0.833	-0.194	0.241
unsubsidized	0.3771**	0.131	2.881	0.004	0.121	0.634
bin_age_u17	-0.2756*	0.118	-2.336	0.019	-0.507	-0.044
hiinc	-0.0786	0.120	-0.657	0.511	-0.313	0.156
female_u17	0.2023	0.172	1.176	0.239	-0.135	0.539
race_cau_u17	0.0504	0.121	0.417	0.677	-0.187	0.287
married_u17	0.1868	0.161	1.163	0.245	-0.128	0.502
own_bank_u17	0.4102**	0.160	2.571	0.010	0.097	0.723
bin_cohort	-0.0323	0.126	-0.257	0.797	-0.278	0.214
ins_heal_u17	-0.0081	0.118	-0.069	0.945	-0.239	0.222
hh_adult_u17	-0.0442	0.093	-0.475	0.634	-0.226	0.138
bin_child_u17	0.0186	0.147	0.127	0.899	-0.269	0.307
own_bus_u17	-0.0417	0.272	-0.153	0.878	-0.576	0.492
own_prop_u17	-0.1949	0.354	-0.551	0.582	-0.889	0.499
own_ira_u17	-0.0057	0.203	-0.028	0.977	-0.403	0.391
src_welf_u17	0.0200	0.130	0.153	0.878	-0.235	0.275
own_car_u17	-0.1385	0.173	-0.801	0.423	-0.477	0.200
own_scale2_u17	0.0885**	0.027	3.252	0.001	0.035	0.142
str_scale2_u17	0.1910	0.251	0.760	0.447	-0.302	0.684
gv_scale2_u17	-0.1620	0.343	-0.472	0.637	-0.835	0.511
gt_scale2_u17	0.0038	0.329	0.012	0.991	-0.641	0.648
bin_sat_heal_u17	0.5770**	0.174	3.315	0.001	0.236	0.918
bin_sat_fin2_u17	-0.1397	0.131	-1.064	0.287	-0.397	0.118
ci_scale_u17	0.2730	0.279	0.978	0.328	-0.274	0.820
cat_ass_tot_1.0	0.3575	0.194	1.840	0.066	-0.023	0.738
cat_ass_tot_2.0	0.2047	0.208	0.986	0.324	-0.202	0.612
cat_ass_tot_3.0	0.3332	0.177	1.883	0.060	-0.014	0.680
cat_ass_tot_missing	-0.0839	0.221	-0.379	0.704	-0.517	0.349
cat_lib_tot_1.0	-0.0218	0.210	-0.104	0.917	-0.434	0.390
cat_lib_tot_2.0	-0.0298	0.232	-0.128	0.898	-0.485	0.425
cat_lib_tot_3.0	-0.1527	0.151	-1.009	0.313	-0.450	0.144
cat_lib_tot_missing	-0.1609	0.187	-0.858	0.391	-0.528	0.206
tri_ed_u17_1.0	0.0063	0.133	0.047	0.962	-0.254	0.266
tri_ed_u17_2.0	0.3462*	0.160	2.163	0.031	0.033	0.660

Table C.2: Probit Regression Results

C.3 Logit model

Dependent variable:	own_home_u42		**Pseudo R-squared:**			0.1009
Number of observations:	604		**Loglikelihood:**			-371.88
	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
const	-2.6649**	0.676	-3.940	0.000	-3.991	-1.339
treat	0.0355	0.182	0.195	0.845	-0.321	0.393
unsubsidized	0.6214**	0.216	2.876	0.004	0.198	1.045
bin_age_u17	-0.4524*	0.194	-2.332	0.020	-0.833	-0.072
hiinc	-0.1415	0.196	-0.720	0.471	-0.526	0.243
female_u17	0.3170	0.280	1.133	0.257	-0.231	0.865
race_cau_u17	0.0818	0.198	0.413	0.679	-0.306	0.470
married_u17	0.2980	0.264	1.129	0.259	-0.219	0.815
own_bank_u17	0.6745*	0.267	2.527	0.011	0.151	1.198
bin_cohort	-0.0573	0.206	-0.278	0.781	-0.461	0.346
ins_heal_u17	-0.0229	0.193	-0.119	0.906	-0.401	0.356
hh_adult_u17	-0.0732	0.152	-0.483	0.629	-0.371	0.224
bin_child_u17	0.0267	0.240	0.111	0.912	-0.444	0.497
own_bus_u17	-0.0725	0.447	-0.162	0.871	-0.948	0.803
own_prop_u17	-0.3023	0.590	-0.512	0.608	-1.459	0.854
own_ira_u17	0.0052	0.332	0.016	0.988	-0.646	0.657
src_welf_u17	0.0427	0.214	0.200	0.842	-0.376	0.462
own_car_u17	-0.2300	0.284	-0.810	0.418	-0.786	0.326
own_scale2_u17	0.1426**	0.045	3.180	0.001	0.055	0.230
str_scale2_u17	0.3067	0.413	0.743	0.458	-0.503	1.116
gv_scale2_u17	-0.2352	0.564	-0.417	0.677	-1.342	0.871
gt_scale2_u17	-0.0129	0.540	-0.024	0.981	-1.071	1.045
bin_sat_heal_u17	0.9493**	0.294	3.226	0.001	0.373	1.526
bin_sat_fin2_u17	-0.2311	0.216	-1.069	0.285	-0.655	0.193
ci_scale_u17	0.4619	0.456	1.013	0.311	-0.432	1.356
cat_ass_tot_1.0	0.5872	0.318	1.848	0.065	-0.036	1.210
cat_ass_tot_2.0	0.3443	0.339	1.016	0.310	-0.320	1.009
cat_ass_tot_3.0	0.5465	0.291	1.878	0.060	-0.024	1.117
cat_ass_tot_missing	-0.1350	0.366	-0.369	0.712	-0.852	0.583
cat_lib_tot_1.0	-0.0418	0.348	-0.120	0.904	-0.723	0.639
cat_lib_tot_2.0	-0.0323	0.377	-0.086	0.932	-0.772	0.707
cat_lib_tot_3.0	-0.2394	0.248	-0.964	0.335	-0.726	0.247
cat_lib_tot_missing	-0.2598	0.308	-0.842	0.400	-0.864	0.345
tri_ed_u17_1.0	-0.0025	0.219	-0.012	0.991	-0.433	0.428
tri_ed_u17_2.0	0.5427*	0.262	2.075	0.038	0.030	1.055

Table C.3: Logit Regression Results

C.4 Generic Machine Learning

	Coefficient	*P-value*	*95% confidence interval*
Lasso			
Baseline conditional average	0.0155	0.779	[-0.111, 0.142]
Heterogeneity	-0.0112	0.977	[-0.981, 0.926]
Tree			
Baseline conditional average	0.0159	0.780	[-0.113, 0.144]
Heterogeneity	-0.0043	0.976	[-0.315, 0.326]
Random forest			
Baseline conditional average	0.0167	0.758	[-0.109, 0.144]
Heterogeneity	0.1289	0.687	[-0.643, 0.973]
Support vector machine			
Baseline conditional average	0.0218	0.697	[-0.105, 0.148]
Heterogeneity	0.1145	0.572	[-0.352, 0.571]
Gradient boosting			
Baseline conditional average	0.0156	0.780	[-0.113, 0.143]
Heterogeneity	-0.0081	0.983	[-0.869, 0.872]
Linear regression			
Baseline conditional average	0.0206	0.715	[-0.107, 0.147]
Heterogeneity	0.0012	0.982	[-0.350, 0.341]

Table C.4: The best linear predictors of the baseline conditional average (β_1) and the heterogeneity parameter (β_2).

	Coefficient	*P-value*	*95% confidence interval*
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_1}}$	-0.008 (0.141)	0.949	[-0.284, 0.274]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_2}}$	-0.006 (0.147)	0.961	[-0.295, 0.275]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_3}}$	-0.001 (0.146)	0.936	[-0.286, 0.296]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_4}}$	0.024 (0.145)	0.844	[-0.260, 0.303]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_5}}$	0.025 (0.145)	0.844	[-0.259, 0.308]

Table C.5: Sorted group average treatment effects using Lasso

	Coefficient	*P-value*	*95% confidence interval*
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_1}}$	-0,004 (0.151)	0.977	[0.293, 0.284]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_2}}$	-0,004 (0.149)	0.977	[0.288, 0.283]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_3}}$	-0,001 (0.145)	0.993	[0.283, 0.284]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_4}}$	0,003 (0.133)	0.905	[0.264, 0.287]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_5}}$	0,017 (0.133)	0.890	[0.278, 0.306]

Table C.6: Sorted group average treatment effects using a Tree

	Coefficient	*P-value*	*95% confidence interval*
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_1}}$	-0.032 (0.145)	0.800	[0.315, 0.251]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_2}}$	-0.014 (0.147)	0.915	[0.298, 0.273]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_3}}$	0.023 (0.141)	0.856	[0.264, 0.304]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_4}}$	0.030 (0.145)	0.801	[0.257, 0.308]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_5}}$	0.037 (0.146)	0.743	[0.248, 0.322]

Table C.7: Sorted group average treatment effects using a Random Forest

	Coefficient	*P-value*	*95% confidence interval*
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_1}}$	-0.010 (0,144)	0.939	[-0.292, 0.273]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_2}}$	0.011 (0.145)	0.924	[-0.272, 0.295]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_3}}$	0.015 (0,144)	0.900	[-0.268, 0.298]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_4}}$	0.017 (0,146)	0.828	[-0.269, 0.303]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_5}}$	0.042 (0,144)	0.737	[-0.241, 0.326]

Table C.8: Sorted group average treatment effects using a Support Vector Machine

	Coefficient	*P-value*	*95% confidence interval*
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_1}}$	0.007 (0,141)	0.868	[-0.270, 0.280]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_2}}$	0.012 (0,147)	0.909	[-0.275, 0.299]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_3}}$	0.018 (0,15)	0.853	[-0.277, 0.314]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_4}}$	0.022 (0,145)	0.860	[-0.262, 0.307]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_5}}$	0.022 (0,148)	0.865	[-0.269, 0.309]

Table C.9: Sorted group average treatment effects using Gradient Boosting

	Coefficient	*P-value*	*95% confidence interval*
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_1}}$	-0.016 (0,144)	0.897	[-0.299, 0.266]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_2}}$	-0.004 (0,145)	0.974	[-0.288, 0.280]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_3}}$	0.018 (0,145)	0.868	[-0.266, 0.302]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_4}}$	0.035 (0,144)	0.772	[-0.248, 0.318]
$\hat{\boldsymbol{\theta}}_{\boldsymbol{\mathcal{G}_5}}$	0.054 (0,144)	0.664	[-0.228, 0.336]

Table C.10: Sorted group average treatment effects using Linear Regression

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0.295***	0.000	[0.163, 0.427]
$\boldsymbol{\delta_2}$	0.425***	0.000	[0.273, 0.561]
$\boldsymbol{\delta_3}$	0.508***	0.000	[0.354, 0.646]
$\boldsymbol{\delta_4}$	0.575***	0.000	[0.422, 0.711]
$\boldsymbol{\delta_5}$	0.697***	0.000	[0.555, 0.823]
$\boldsymbol{\delta_5 - \delta_1}$	0.410***	0.000	[0.223, 0.595]

Table C.11: Classification analysis for the regressor *hiinc*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,492***	0.000	[0.347, 0.636]
$\boldsymbol{\delta_2}$	0,442***	0.000	[0.289, 0.578]
$\boldsymbol{\delta_3}$	0,408***	0.000	[0.257, 0.543]
$\boldsymbol{\delta_4}$	0,358***	0.000	[0.211, 0.489]
$\boldsymbol{\delta_5}$	0,303***	0.000	[0.163, 0.427]
$\boldsymbol{\delta_5 - \delta_1}$	-0,197*	0.027	[-0.393, 0.003]

Table C.12: Classification analysis for the regressor *race_cau_u17*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,139**	0.001	[0.033, 0.229]
$\boldsymbol{\delta_2}$	0,192***	0.000	[0.070, 0.296]
$\boldsymbol{\delta_3}$	0,208***	0.000	[0.083, 0.317]
$\boldsymbol{\delta_4}$	0,258***	0.000	[0.124, 0.376]
$\boldsymbol{\delta_5}$	0,303***	0.000	[0.163, 0.427]
$\boldsymbol{\delta_5 - \delta_1}$	0,148*	0.035	[-0.013, 0.323]

Table C.13: Classification analysis for the regressor *married_u17*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,189***	0.000	[0.069, 0.292]
$\boldsymbol{\delta_2}$	0,272***	0.000	[0.110, 0.357]
$\boldsymbol{\delta_3}$	0,275***	0.000	[0.138, 0.396]
$\boldsymbol{\delta_4}$	0,291***	0.000	[0.152, 0.415]
$\boldsymbol{\delta_5}$	0,353***	0.000	[0.207, 0.482]
$\boldsymbol{\delta_5 - \delta_1}$	0,164*	0.029	[-0.009, 0.344]

Table C.14: Classification analysis for the regressor *bin_cohort*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,254***	0.000	[0.115, 0.410]
$\boldsymbol{\delta_2}$	0,342***	0.000	[0.168, 0.508]
$\boldsymbol{\delta_3}$	0,383***	0.000	[0.214, 0.569]
$\boldsymbol{\delta_4}$	0,475***	0.000	[0.281, 0.672]
$\boldsymbol{\delta_5}$	0,648***	0.000	[0.430, 0.871]
$\boldsymbol{\delta_5 - \delta_1}$	0,385***	0.001	[0.124, 0.654]

Table C.15: Classification analysis for the regressor *hh_adult_u17*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,615***	0.000	[0.535, 0.689]
$\boldsymbol{\delta_2}$	0,564***	0.000	[0.485, 0.642]
$\boldsymbol{\delta_3}$	0,534***	0.000	[0.455, 0.614]
$\boldsymbol{\delta_4}$	0,506***	0.000	[0.429, 0.582]
$\boldsymbol{\delta_5}$	0,495***	0.000	[0.418, 0.574]
$\boldsymbol{\delta_5 - \delta_1}$	-0,129**	0.006	[-0.234, -0.024]

Table C.16: Classification analysis for the regressor *str_scale2_u17*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,926***	0.000	[0.839, 1,000]
$\boldsymbol{\delta_2}$	0,892***	0.000	[0.790, 0.977]
$\boldsymbol{\delta_3}$	0,875***	0.000	[0.767, 0.966]
$\boldsymbol{\delta_4}$	0,833***	0.000	[0.725, 0.942]
$\boldsymbol{\delta_5}$	0,762***	0.000	[0.629, 0.879]
$\boldsymbol{\delta_5 - \delta_1}$	-0,1803**	0.009	[-0.317, -0.025]

Table C.17: Classification analysis for the regressor *bin_sat_heal_u17*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,451***	0.000	[0.299, 0.586]
$\boldsymbol{\delta_2}$	0,575***	0.000	[0.422, 0.711]
$\boldsymbol{\delta_3}$	0,633***	0.000	[0.493, 0.774]
$\boldsymbol{\delta_4}$	0,692***	0.000	[0.548, 0.819]
$\boldsymbol{\delta_5}$	0,754***	0.000	[0.629, 0.879]
$\boldsymbol{\delta_5 - \delta_1}$	0,3033***	0.000	[0.119, 0.491]

Table C.18: Classification analysis for the regressor *bin_sat_fin2_u17*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,350***	0.000	[0.287, 0.413]
$\boldsymbol{\delta_2}$	0,367***	0.000	[0.301, 0.434]
$\boldsymbol{\delta_3}$	0,389***	0.000	[0.325, 0.456]
$\boldsymbol{\delta_4}$	0,402***	0.000	[0.335, 0.468]
$\boldsymbol{\delta_5}$	0,439***	0.000	[0.371, 0.512]
$\boldsymbol{\delta_5 - \delta_1}$	0,097*	0.020	[0.001, 0.194]

Table C.19: Classification analysis for the regressor *ci_scale_u17*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,350***	0.000	[0.287, 0.413]
$\boldsymbol{\delta_2}$	0,367***	0.000	[0.301, 0.434]
$\boldsymbol{\delta_3}$	0,389***	0.000	[0.325, 0.456]
$\boldsymbol{\delta_4}$	0,402***	0.000	[0.335, 0.468]
$\boldsymbol{\delta_5}$	0,439***	0.000	[0.371, 0.512]
$\boldsymbol{\delta_5 - \delta_1}$	0,097*	0.020	[0.001, 0.194]

Table C.20: Classification analysis for the regressor *cat_lib_tot_1*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,074*	0.021	[0.000, 0.137]
$\boldsymbol{\delta_2}$	0,108**	0.005	[0.012, 0.188]
$\boldsymbol{\delta_3}$	0,133**	0.001	[0.034, 0.233]
$\boldsymbol{\delta_4}$	0,158***	0.001	[0.046, 0.254]
$\boldsymbol{\delta_5}$	0,238***	0.000	[0.108, 0.351]
$\boldsymbol{\delta_5 - \delta_1}$	0,148*	0.021	[0.004, 0.310]

Table C.21: Classification analysis for the regressor *cat_lib_tot_4*

	Coefficient	*P-value*	*95% confidence interval*
$\boldsymbol{\delta_1}$	0,172***	0.000	[0.057, 0.271]
$\boldsymbol{\delta_2}$	0,208***	0.000	[0.083, 0.317]
$\boldsymbol{\delta_3}$	0,242***	0.000	[0.110, 0.357]
$\boldsymbol{\delta_4}$	0,258***	0.000	[0.124, 0.376]
$\boldsymbol{\delta_5}$	0,344***	0.000	[0.207, 0.482]
$\boldsymbol{\delta_5 - \delta_1}$	0,164*	0.026	[-0.007, 0.344]

Table C.22: Classification analysis for the regressor *tri_ed_u17_2*