“Using Causal Tree Algorithms with Difference in Difference methodology: a way to have Causal Inference in Machine Learning”

TESIS PARA OPTAR AL GRADO DE

MAGISTER EN ANÁLISIS ECONÓMICO

Alumno: Juan José Balsa Fernández

Profesor Guía: David Díaz S. y Diego Ronchetti

Santiago, Junio 2018

Abstract

The capacity of understand the real effects of a public policy intervention in the population has been for a long time one of the main focus of the economist around the world. At the same time, the development of different statistical methodologies have deeply helps them to complement the economic theory with the different types of data. One of the newest developments in this area is the Machine Learning algorithms for Causal inference, which gives them the possibility of using huge amounts of data, combined with computational tools for much more precise results. Nevertheless, these algorithms have not implemented one of the most used methodologies in the public evaluation, the Difference in Difference methodology. This document proposes an estimator that combines the Honest Causal Tree of Athey and Imbens (2016) with the Difference in Difference framework, giving us the opportunity to obtain heterogeneous treatment effect. Although the proposed estimator has higher levels of Bias, MSE, and Variance in comparison with the OLS, it is able to find significant results in cases where OLS do not, and instead of estimate an Average Treatment Effect, it is able to estimate a treatment effect for each individual.

JEL Classification: C14, C23
Key words: Machine Learning, Difference in Difference, Causal Inference, Causal Tree.

Acknowledgment

I am deeply grateful with Diego Ronchetti Ph.D. in Economics from the University of Lugano and assistant professor from the University of Groningen and with David Díaz Ph.D. in Business Intelligence from the University of Manchester and assistant professor of the University of Chile for all your corrections, advice, support and specially all your patience during this last months, without all your hours of dedication this thesis would never had reach to this final state. Also, I like to thank to Andres Sanfuentes M.A. in Economics from the University of Chicago and to Joaquin Mayorga MSc. in Economics from the University of Groningen for all the time spend it to read the several drafts of this thesis and giving valuable feedback and comments.

To my mother for showing me that for love even the greatest of difficulties can be overcome, to my family for teaching me to look at life with humility and passion and to my friends who showed me that the only limit to dreams is oneself.

Contents

1 Introduction

Statistical and econometric tools have advanced strongly in recent years thanks to the computational development and the large amounts of data that are available. In particular, the use of Machine Learning (ML) algorithms (such as: Neural Networks, Lasso Regression, Support Vector Machine, Decision Trees, etc.) has generated a leap in the predictive capacity of economic science.

ML involves the use of unsupervised or supervised algorithms for the mining of patterns and for the prediction of a certain objective. Normally supervised algorithms are used when we predict an outcome through training samples (Athey & Imbens, 2017). On the other hand, in words of Athey and Imbens the objective of unsupervised algorithms is to find patterns in the data such as similar group of items, like clustering images into groups. Therefore, both types of algorithms are not accompanied by causality, which limits their use as tools to explain the phenomena that surround us or to see the effectiveness of different government policies.

The literature related to ML and causal effects is still scarce and it has focused in the use of ML to generate Propensity Score, Matching or Synthetic Controls, in search of the heterogeneous effects of the programs. In these cases, the ML tools are usually used to define which group of variables are used for prediction and causality, or which subjects of the sample are used for suitable training the algorithm. However, the use of ML in more traditional methodologies such as Difference in Difference (DID) shown in the classic papers of Ashenfelter & Card (1985) and Card & Krueger (1993) has not yet been discussed in the literature.

This document proposes to investigate what requirements are needed in order to perform DID estimations, leveraging the advantages of using ML algorithms. The main motivation of this work, arises from the relatively few applications of ML in the economics literature, and particularly, in the DID heterogeneous effects estimation context, which is one of the best established methodologies in the public policy field. In order to do this we will adapt the Honest Tree algorithms (Athey and Imbens (2016)) recently proposed in the literature, to be applied in the DID context. We proposed the adaption to be named Causal-DID-Tree algorithm. This work also shows the gains of using the proposed methodology for the estimation of heterogeneous treatment effects in the DID context. To evaluate the gains, the proposed algorithm will be tested using simulated data and its performance will be compared against the classical OLS estimation of the DID methodology.

This document contributes to the literature in three key dimensions: Firstly, we investigate the current state of art in the causal ML literature, and summarize the main findings. Secondly, we propose an adaptation of the current causal ML algorithms to be applicable in the DID heterogeneous estimation context, and thirdly, we compare the proposed methodology against the classical approaches. To best of the author's knowledge this is the first implementation of a methodology that takes advantage of the ML algorithms to create an easy and direct way to estimate DID with heterogeneous effects.

The main results of our simulations indicate that in the presence of an exogenous shock and parallel trends between the treatment and control groups, the results delivered by the Causal-DID-Tree are unbiased and present a similar fitness to the one achieved by the traditional OLS estimation. Moreover, by construction, the Causal-DID-Tree algorithm is able to estimate heterogeneous treatment

effects for each individual in the sample, feat that is not possible with the OLS classical methodology, that only estimates average treatment effects.

The implications of being able to estimate the effects of a public policy at the individual level, are extremely valuable for both the academia and practitioner's world. In a wide variety of contexts, public policies do not have the same effect for the entire population, therefore in a world of scarce resources, the knowledge of the real effect of the policy for each group of individuals, or even better, at the individual, can improve the efficiency of public spending of resources and enhanced the wealth and well-being of the targeted population.

The document is organized as follows. Section 2 perform a literature review of ML algorithms applications and theoretical contributions in the economic context, and the current state of art of ML and causal inference. Section 3 discuss what causal inference is, what does the difference in difference methodology do and why OLS regression can be used for causal inference. Section 4 presents a description of decision trees algorithms, what are the generic changes that have been proposed to adapt decision trees to be used in the causal inference context, and discussed in detail the proposed Causal-DID-Tree algorithm methodology for heterogeneous effects estimation. Section 5 shows an experimental application of the proposed methodology, and compare it with OLS estimation. Several robustness checks are applied, and the performance of the algorithm is discussed and presented. Finally, Section 6 will concludes this document, highlighting the main contributions, results, implications, limitations and future work. [1]

2 Machine Learning and Causal inference in the literature

2.1 What is a Machine Learning Supervised Algorithm?

The focus of this document is to use Supervised Algorithms for causal inference problems, but first it is necessary to understand, "What a Supervised Algorithm is". In simple words, a supervised Machine Learning algorithm is a computer program that finds patterns in a dataset with the goal of making accurate predictions of a certain outcome variable, conditional to a set of features or input variables. To find the corresponding patterns, the humans need to present to the machine with examples of the problem that it needs to learn. The computer stars from random predictions, and adjust them comparing the current output of the model with the expected outcome until it reaches a solution that it is considered to be good enough.

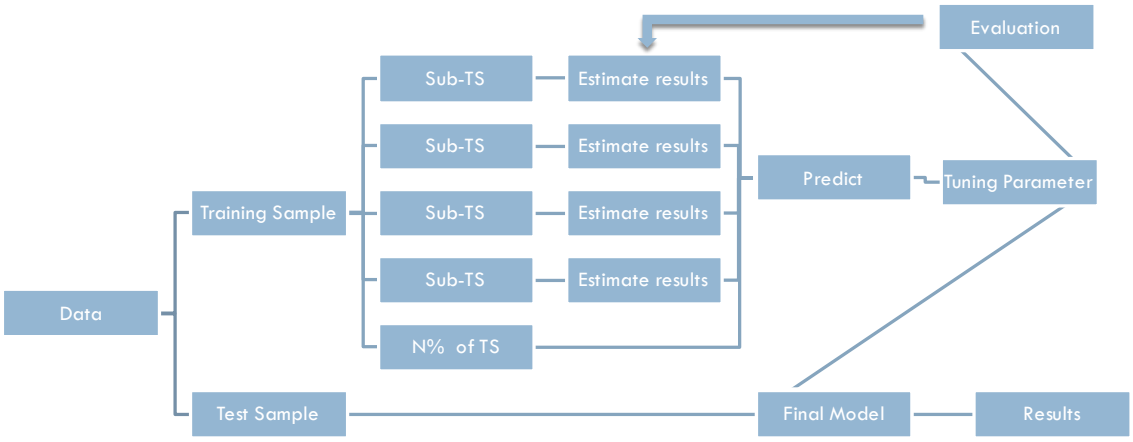
According to Athey & Imbens (2017) "*Supervised machine learning focuses primarily on prediction problems: given a dataset with data on an outcome Y_i, which can be discrete or continuous, and some predictors X_i, the goal is to estimate a model on a subset of the data, given the values of the predictors X_i. This subset is called the training sample, and it is used for predicting outcomes in the remaining data, which is called the test sample.*" As mentioned, the main goal of supervised machine learning algorithms is to perform prediction for an outcome, given a set of predictors or covariates. In contrast, in causal inference, the objective is to test whether a certain treatment have an effect on the population that receives it, and to quantify this effect if its exist.

[1] The document also have two other sections (7 and 8) that provide support from the results and the computational code used.

Athey & Imbens (2017) emphasize that a key distinction between prediction and causal inference comes from the fact that supervised machine learning methods typically rely on data-driven model selection, but in contrast, econometric applications rely on economic theory to define what the model specification should be. Supervised machine learning algorithms, then, take a more heuristic approach, and most commonly through cross-validation, find the best specification for the task at hand. As mentioned, often the main focus on ML is on prediction performance without regard to the implications for inference.

Figure I shows how a Supervised algorithm usually works. First, it splits the sample into a training sample and a test sample. Then, for the training sample, the algorithm will create subsamples and leave a n percent apart. After, the algorithm will start to estimate results from the subsamples and use them to predict the outcomes of the n percent that it was left aside. Later on, the tuning parameter is chosen, based on the one that minimizes the loss function, that is normally defined as the sum of the squared residuals in the cross-validation samples. The final model performance is assessed by calculating the mean-squared error of model predictions (that is, the sum of squared residuals) on the held-out test sample, which was not used at all for model estimation or tuning. Again, Athey & Imbens (2017) express that the predictions using this methodology are not typically unbiased and estimators may not be asymptotically normal and centered around the estimate. Then in Section 4 we will explain in more detail how the specific algorithm used in this document works.

Figure I: Supervised Algorithm Example



2.2 What has the literature done so far?

In recent years, the amount of general literature related to ML has increased dramatically. One of the most common topics is the use of algorithms in controlling large numbers of covariates. McCaffrey et al. (2004) use Lasso and Random Forest regressions looking to estimate a propensity

score. In contrast, Wyss et al. (2014) utilize simulations and empirical tests to compare Covariate-balancing propensity scores with logistic regression, boosted classification and Regression Trees. These two examples are critized by Athey & Imbens (2017), who suggest that such methods do not necessarily emphasize the covariates that correlate to both the outcomes and the treatment indicator.

Athey et al. (2017) suggest better ways for working with ML under the presence of large numbers of pre-treatment variables. For example, the Approximate Residual Balancing Estimator (ARBE) is proposed by Athey, Imbens, and Wager (2016) and uses elastic net (or LASSO) to estimate conditional outcome expectations. This is then put through an approximate balancing approach to further remove bias, which can come from remaining imbalances in the pre-treatment variables. Moreover, Belloni et al. (2013) propose the double selection estimator, which uses LASSO as a covariate selection method. First, the authors select pretreatment variables that are essentials in explaining the outcome, then they combine the two sets of pre-treatment variables. Also, Van der Laan and Rubin (2006) propose a closely-related Machine Learning Estimator (MLE) and Chernozhukov et al. (2016) in the context of much more general estimation problem, propose a closely related Double Machine Learning Estimator (DMLE) that also incorporates sample splitting to further improve the convergence rates and its robustness.

Another group of researchers has focused on finding weights that can balance covariates or functions of the covariates, in order to imitate randomized experiment data, once it has been re-weighted. Some examples of these experiments are in Athey et al. (2017). Similar approaches has also been developed by Graham, Pinto, and Egel (2016), Zubizarreta (2015) and Imai and Ratkovic (2014). Another good example of this approach is Hainmueller (2012) proposition of the Entropy Balancing Method (EBM). The authors methodology relies on a maximum entropy re-weighting scheme, which calibrates unit weights in order to satisfy a potentially large set of pre-specified balance conditions that incorporate information about known sample moments in the treatment and control group.

Moving to the development of some of these algorithms, in the paper of Athey and Tibshirani (2017) the authors generalize the Random Forest method of Breiman (2001) and use that as an alternative way to estimate non-parametric quantile regression, conditional average partial effect estimation, and heterogeneous treatment effect estimation via instrumental variables. Moreover, Wager and Athey (2017) develop a Causal Generalized Random Forest algorithm that is able to use the Random Forest methodology to find heterogeneous treatment effects and obtain a causal inference from the results. They additionally discuss a practical method for constructing asymptotic confidence intervals for the true treatment effect that are centered at the causal forest estimates.

In the case of the literature related to ML and Causal inferences, this is mostly scarce and in some cases, the properties of the estimators are not very well defined. Nevertheless, Imai and Ratkovic (2013), motivated by two well-known randomized evaluation studies in the social sciences, use a method that adapts Support Vector Machine classifier to estimate the effects of both treatments and attributes. The key in the authors methodology -that allows them to found treatment effects-, is the use of different penalty terms for the two types of features. In the same line, Tian et al. (2014) and Weisberg and Pontes (2015) focus mostly in medical trials, developing methods similar to Lasso for causal inference in a sparse high-dimensional linear setting. Moveover, Beygelzimer and Langford (2009) develop an algorithm for optimal treatment policy estimation, using off-the-shelf

loss minimization methods.

Furthermore, Cicala (2017) and Burlig et al. (2017) using ML algorithms estimated energy use counterfactuals that then are used in a DID methodology with OLS regression. In addition, Athey and Imbens (2016) creates the Causal Honest Tree algorithm that is a modification of the Regression Tree methods but focuses in optimize for the goodness of fit in treatment effects. Lastly, in the case of Bayesian nonparametric methods Taddy et al. (2016) use them with Dirichlet priors to flexibly estimate the data-generating process, and then project the estimates of heterogeneous treatment effects and their measurement in relation to observable covariates.

In summary, although the Machine Learning literature is quite extensive and has been focused on generating new methodologies, the literature that links Machine Learning and Causal Inference remain scarce. This literature has been focused mainly on Propensity Score and obtaining heterogeneous effects for cross-sectional data in public policy evaluations. However, the properties of these estimators and whether or not they converge to their real values continue to be studied. Finally, it is interesting to note that so far in the literature there hasn't been an article that directly relates the use of Machine Learning algorithms to the Difference in Difference methodology.

3 Causal Inference and OLS

3.1 What is Causal Inference?

The term "Causal Inference" although it may sound a bit abstract is present everyday in quite surprising ways. For example, "my neck hurts, because I slept badly" or "today I worked all day, I'm exhausted", what we are doing in these examples is giving a causal connotation to a bad night's sleep or to work. Although it sounds simple, causal inference is the effect that is attributed to an action on a unit (object, person, etc) and many academics spend their entire lives looking for or trying to understand those effects. In a more formal way, what is sought is to obtain the variation of the variable Y_i, when a change in the variable X_i occurs and everything else remains constant, therefore, ceteris paribus.

The classic document of causal inference was written by Rubin (1974), however he says that the original concept is from Fisher. Rubin proposes that if we have an individual i, and a treatment T_i that takes value 1 when the subject is treated and 0 when not, the changes in the results of this individual Y_i will be the causal effect of the treatment T_i, i.e.

$$Treatment = E[Y_i|T_i = 1] - E[Y_i|T_i = 0] \tag{1}$$

Nevertheless, trying to obtain that effect in reality is impossible because it will be involved in what Holland (1986) called "the fundamental problem of causal inference", which refers to the fact that a subject cannot receive and not receive treatment at the same time (the absence of counterfactual). So, when we are evaluating we must compare subjects who received different levels of treatment, which under certain conditions can cause bias. But, thanks to one of the most important theorems of statistic, the Central Limit Theorem (CLT), that established that if we add independent random variables or observations their properties will tend to a normal distribution. Moreover, if there is a sufficiently big random sample of the population, the normality property will apply and it will

be possible to obtain the Average Treatment Effects (ATE), taking two different groups (T and C), one affected by the treatment and the other not:

$$ATE = E[Y_i|T_i = 1] - E[Y_i|T_i = 0] \tag{2}$$

After all, the reality is not that simple and in the words of Cameron & Trivedi (2005) *"random assignment of treatment is generally not feasible in economics, estimation of ATE-type parameters must be based on observational data generated under nonrandom treatment assignment. Then the consistent estimation of ATE will be threatened by several complications that include, for example, possible correlation between the outcomes and treatment, omitted variables, and endogeneity of the treatment variable".* A way to solve the problems highlighted by Cameron & Trivedi (2005), is the Difference in Difference methodology originally used by Ashenfelter & Card (1984) and Card & Krueger (1993).

The Difference in Difference methodology is useful when we have Panel data (i.e. different units in different periods of time) or there are two cross-section data sets of two different periods of time and in sort point of time (in example, between the two data set) a shock affect to one group of the population and not to the other. Cameron & Trivedi (2005) explain that there are two underlying assumptions in Difference in Difference (DID). First, it is assumed that a common time trend exists between groups, i.e. the time effects are common across treated and untreated individuals. The common trends assumption is needed if either panel or cross-section data is used. Second, if cross-section data is used then the composition of the treated and untreated groups is assumed to be stable before and after the change -with panel data difference eliminates the fixed effects-.

$$DID = (E[Y_{it+1}|T = 1] - E[Y_{it+1}|T = 0]) - (E[Y_{it}|T = 1] - E[Y_{it}|T = 0]) \tag{3}$$

DID methodology is supposed to mitigate the time trend factors and groups invariant characteristics. Some authors say that this helps with the selection bias and with others economic exogenous factor which could occur. Nevertheless, the DID methodology is not absent of critics. First, the demonstration of the parallel trends or the idea that the shock was really exogenous for both groups is more theoretical, rather than a mathematical thing to do. Second, this methodology could also suffer from reverse causality or omitted variable bias, this mean that the effect that is being obtained is not the real causal effect of the treatment.

Another form of causality in the literature is developed by Granger (1969) and is called Granger's Causality. This methodology is mainly used in the time series data, and in Holland (1986) it is explained in this way *"a variable cause another in a Granger's way if this variable statistically predict the other one, this is, if the variable is prior than other one"*. A more mathematical explanation from Holland is that if X_i, W_i, Z_i denote three variables defined on a population, then X_i and W_i are conditionally independent given Z_i if

$$Pr(W_i = w|X_i = x, Z_i = z) = Pr(W_i = w|Z_i = z) \tag{4}$$

So, in Granger's time-series setting, the value of W_i is determined at some point s, and the values of X_i and Z_i, are determined at or prior to some other point r<s. It will be said that X_i is not a Granger cause of W_i (relative to the information of Z_i) if X_i and W_i are conditionally independent given Z_i. So Holland continues his explanation, saying that X_i is a Granger cause of W_i if different values of X_i lead to a predictive distributions of W_i given both X_i and the information in Z_i, that

is, if X_i helps predict W_i even when Z_i is taken into consideration. Therefore, Granger's Causality helps you to know which variable precedes the other from a statistical point of view, however, this way of observing causality is not exempt from the problem of reverse causality, or from an economically spurious relationship, so there is not a pure or perfect method of causality.

In conclusion, Causal Inference is not a simple topic, and in many cases it has philosophical extensions for what is understood as a cause or an effect. However, for the study of social sciences such as economics, sociology, psychology or others, in order to find causal effects, it is necessary to be able to overcome the "Fundamental problem of causal inference" through one of the several ways that have been developed over the years, and have a theoretical framework on how variables interact and what mechanisms there are between them. In this way statistics only confirms and quantifies what theoretically should be causal.

3.2 When and Why OLS is Causal?

Following the guidance of Holland (1986), there are certain assumptions that a process must meet to bypass the fundamental problem of causality and therefore, that process estimates or treatments effects would be causal. But how does it works with OLS? To answer this question, we must go back to the equation (1), which shows us the true effect of the treatment. As previously mentioned, this effect is impossible to find, but if the assumption of independence is fulfilled, that is, that the units are randomly assigned between treatment and control, we will arrive at the equation (2). Then, the treatment effect can be found if the assumption of constant effect is used on the equation (2), that is, that the effect of the treatment is the same for each of the sample units:

$$Treatment = E[Y_i|T_i = 1] - E[Y_i|T_i = 0] \tag{5}$$

Which can be rewritten as:

$$E[Y_i|T_i = 1] = Treatment + E[Y_i|T_i = 0] \tag{6}$$

Finally, this could be written in a linear equation form:

$$Y_i = \alpha + \beta * Treatment_i + e_i \tag{7}$$

Where Y_i is the outcome value of unit i, α is the constant, $Treatment_i$ is a dummy variable that takes value 1 if the unit was treated or not, β is the treatment effect and e_i is the estimation error that is distributed $e_i \sim (0, \sigma^2)$. Now, the way to estimate this effect, is usually by minimizing a loss function $min \sum_i L(e_i)$, which, for OLS is the sum of squared errors:

$$min \sum_i L(e_i) = min \sum_i (e_i)^2 = min \sum_i (Y_i - \widehat{Y})^2 = min \sum_i (Y_i - g(x_i, \beta))^2$$

Where, $g(x_i, \beta)$ is any function, which represents the estimated value of Y_i. This estimated value of Y_i, under OLS is linear, therefore:

$$min \sum_i (Y_i - g(x_i, \beta))^2 = min \sum_i (Y_i - E[Y|X])^2 = min \sum_i (Y_i - X'\widehat{\beta})^2$$

That after the minimization and expressing the above in a matrix way[2]:

$$\widehat{\beta} = (X'X)^{-1}X'Y = (X'X)^{-1}X'X\beta + (X'X)^{-1}X'e \tag{8}$$

This brings us again the independence assumption, where in the case of OLS, it must be fulfilled that the independent variables (X) are not correlated with the error e ($E[e|X] = 0$), that is, both vectors are orthogonal ($X \perp e$). Applied forms to see this, is that there are no relevant variables omitted, that the treatment is really exogenous to the assignment of the group of treatment and control, among others. Therefore, if the previous assumption is fulfilled we have that the estimator fulfills with:

$$\widehat{\beta} = \beta \tag{9}$$

As can be seen in equation (9), when $E[e|X] = 0$, the term on the right of equation (8) becomes 0 for a sample with infinite size, and by simple algebra the term on the left becomes the real value of β, thus the OLS estimator is unbiased. In addition to these results, if the OLS estimator complies to have at least heteroscedastic errors conditioned in the regressors, the model is well specified and the vector of regressors x_i is possibly stochastic with second finite moment ($M_{xx} = lim_{i\to\inf} N^{-1}X'X$ exist), then the estimator will asymptotically converge to the actual value of the treatment, which, if the theory supports it, is the causal population effect of the treatment.

4 Causality in Machine Learning

So far we have described what causality is, how to identify if a process is causal and why one of the most used methodologies of econometric (OLS) is causal - under certain assumptions-. Now we must begin to analyze the Machine Learning algorithms and their relation to causality. For the purposes of this document, the Honest Causal Tree algorithm has been chosen, which is part of the Decision and Regression Tree algorithms family. In the following subsections, we will explain how decision and regression trees works, what modifications should be made to obtain causal results and how the proposed methodology is able to combine Honest Causal Trees with DID.

4.1 What is a Decision Tree algorithm? How its work?

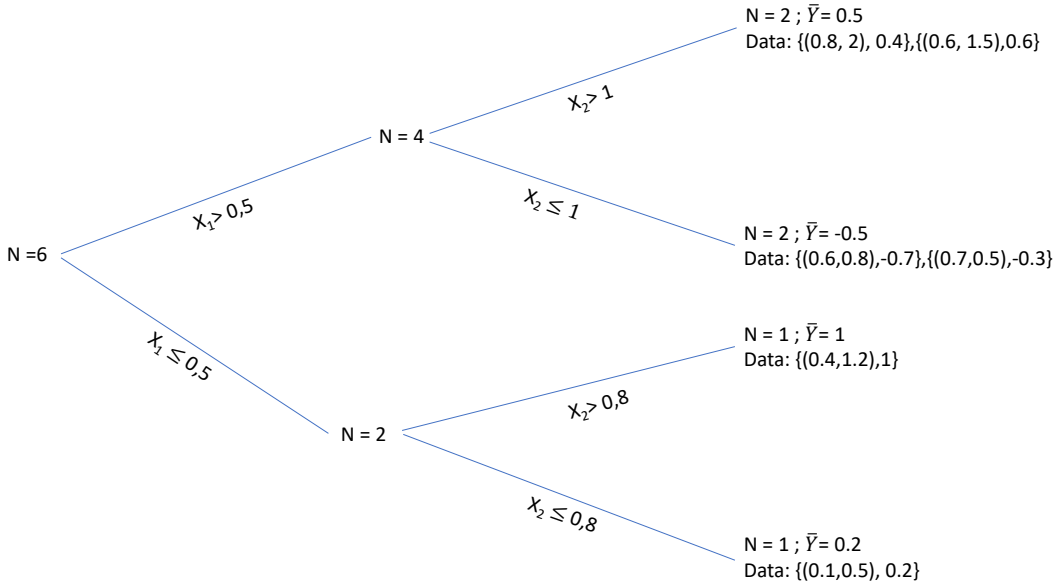
A Decision Tree is a non-parametric supervised learning method that assigns probabilities to different outcomes based on a certain context, another way to think of the Decision Trees is like a decision-making device that is usually used for classification and regression. (Magerman (1995)). These algorithms, by learning simple decision rules inferred from the characteristics of the data, create a model that predicts the value of an objective variable. As we can see in the example of *Figure II* the tree has Nodes, Edges, and Leaves. The Nodes test for the value of a certain attribute. The Edges correspond to the outcome of a test and are connected with the next node or leaf and the Leaves are the terminal nodes that predict the outcome. Moreover, if we take the first leaf we can see that each of them presents: the number of observations (N=2); the mean of that leaf outcome ($\bar{Y} = (0.4 + 0.6)/2 = 0.5$); and which observations where chosen for that leaf (as a vector $\{(X_1, X_2), Y\}$) after they go through the different nodes.

Regardless there is a big family of decision trees algorithms all of them work with the same logic, dividing the problem into different small subproblems -Divide and conquer algorithms-, this means

[2]Capital letters refer to the matrix of the vectors of the previous variables.

that: (1) they select a test for the root and create a branch for each possible outcome of the test; (2) split the observations into subsets (one for each branch extending for the node); (3) repeat the process recursively for each branch using only observations inside the branch; (4) stop the recursion of the branch if all the observations of the branch are for the same class. After the creation of a tree, the best algorithms start a process called pruning, this mean that the algorithms cut some branches of the tree in the base of some criterion. Hence, almost all tree algorithms have two important criterion to be defined, one related to the splitting process (Variance, Gini, Entropy, etc.) and the other one related to pruning (Mean-Squared Error or other numeric error measures).

Figure II: Decision Tree Example



Note: Diagram created by the author, only for visual purposes, not real data used. N is the number of observation in that Leaf or Edge. $\bar{Y}$ is the mean outcome of the observations in that leaf. "Data" show the observations inside that leaf and are presented as a vector $\{(X_1, X_2), Y\}$

These trees allow us to: (1) create "rankings" of observations according to the intensity with which they belong to a class; (2) assign a probability to each observation according to the intensity of belonging to the class; (3) describe historical relationships between the variables and the classes of the target variable; (4) classify the new observations following the historical rules; (5) automatically select the variables that most help classify the data into groups; (6) organize the explanatory variables in order of importance; (7) understand why an observation belongs to a class.

4.2 Regression Trees and CART

Once understood the general framework of a decision tree, it is time to enter into the Regression Tree. These trees are a type of decision tree and as it has been already mentioned are adaptive, that is, they use training data to select the models, which leads to a spurious correlation between the variables and the results, which only decreases as the data increases. So, taking the methodology of Athey and Imbens (2016) on "honest" inference the algorithm of CART (Classification and Regression Tree) will be explained and then transformed from an Adaptive Regression Tree to a Causal Tree.

For a general setup and following professors Athey and Imbens (2016) explanation, let us define Π as a partitioning of the feature space χ, with N(Π) the number of element in the partition or tree. So:

$$\Pi = (l_1, ..., l_{N(\Pi)}), \quad with \quad U_{j=1}^{N(\Pi)} l_j = \chi \tag{10}$$

Athey and Imbens (2016) begin by defining Φ as the space of partitions and $l(x;\Pi)$ the leaf $l \in \Pi$ such that $x \in l$. Also, they denote S as the space of data samples from a population. So, $\pi : S \longrightarrow \Phi$ is an algorithm that based on a sample $F \in S$ constructs a partition. For example, suppose that $\chi = \{A, B\}$, so there are two possible partitions $\Pi_n = \{A, B\}$ or $\Pi_s = \{\{A\}, \{B\}\}$, where the first is not split and the second is fully split, thus the space of trees $\Phi = \{\{A, B\}, \{A\}, \{B\}\}$. Finally, given a sample F, the average outcomes in the two subsamples are $\bar{Y}_A$ and $\bar{Y}_B$. A simple algorithm normally will splits if the difference in average of the outcomes exceeds a threshold m:

$$\Pi(F) = \begin{cases} \{\{A, B\}\} & if\ \bar{Y}_A - \bar{Y}_B \leq m \\ \{\{A\}, \{B\}\} & if\ \bar{Y}_A - \bar{Y}_B > m \end{cases}$$

From this simple split of the algorithm it could be seen the potential bias of the adaptive estimation. Although the unbiased estimator for the difference in the population conditional means $E[Y_i|X_i = A] - E[Y_i|X_i = B]$ is $\bar{Y}_A - \bar{Y}_B$, in this case when it is condition on finding $\bar{Y}_A - \bar{Y}_B > m$ in a particular sample, it is expect that $\bar{Y}_A - \bar{Y}_B$ is larger than the population analog. Furthermore given any partition Π, Athey and Imbens (2016) define the conditional mean function ($\mu(x;\Pi)$) as:

$$\mu(x;\Pi) \equiv E[Y_i|X_i \in l(x;\Pi)] = E[\mu(X_i)|X_i \in l(x;\Pi)] \tag{11}$$

Hence, given a sample F the unbiased estimation for $\mu(x;\Pi)$ is:

$$\hat{\mu}(x;F;\Pi) \equiv \frac{1}{N(i \in F : X_i \in l(x;\Pi))} \sum_{i \in F : X_i \in l(x;\Pi)} Y_i \tag{12}$$

Where N is the cardinality of the subsample, so to simplify the algebra $N(i \in F : X_i \in l(x;\Pi)) = N(F)$. Now, with the conditional mean defined, we can created an expression for the MSE (Mean-Square-Error) for prediction in the adaptive algorithms:

$$MSE_\mu(F^{te}, F^{tr}, \Pi) \equiv \frac{1}{N(F^{te})} \sum_{i \in F^{te}} (Y_i - \hat{\mu}(X_i; F^{tr}, \Pi))^2 \tag{13}$$

Where F^{te} is the test sample and F^{tr} is the training sample. Athey and Imbens (2016) modified this MSE by subtracting $E[Y_i^2]$, because this does not depend on the estimator or affect how the

criterion ranks estimators.

$$MMSE_{\mu}(F^{te}, F^{tr}, \Pi) \equiv \frac{1}{N(F^{te})} \sum_{i \in F^{te}} \{(Y_i - \hat{\mu}(X_i; F^{tr}, \Pi))^2 - Y_i^2\} \tag{14}$$

Then the expected MMSE (modified Mean Square Error) is:

$$EMMSE_{\mu}(\Pi) \equiv E_{F^{te}, F^{tr}}[MMSE_{\mu}(F^{te}, F^{tr}, \Pi)] \tag{15}$$

Finally, it has to be define the criterion that the algorithms will maximize -or minimize-, following CART algorithms the target is:

$$Q^c(\pi) \equiv -E_{F^{te}, F^{tr}}[MMSE_{\mu}(F^{te}, F^{tr}, \pi(F^{tr}))] \tag{16}$$

In CART algorithms the training sample is used to construct and estimate the tree. But, How CART works? Again, Athey and Imbens (2016) have the answer. First, in the tree-creating phase, the algorithms recursively divide the observations of the training sample. Then, for each leaf the algorithms evaluates the candidates splits of the leaf using a "Splitting" criterion that is called the "in-sample goodness-of-fit" criterion ($-MMSE_{\mu}(F^{tr}, F^{tr}, \Pi)$). Normally this conventional criterion will lead to overfitting, and to solve this a penalty term in the tree is introduced, so in that case the criterion doesn't improve just for additional splitting. Secondly, the training sample will be repeated separately in two samples to make cross-validation, where $F^{tr,tr}$ sample is used to create a new tree and used to estimate the conditional mean and the $F^{tr,cv}$ sample is used to evaluate the estimates.

Third, the tree is pruned using a penalty parameter that represents the cost of a leaf. Through a process of evaluation of the trees associated with each value of the penalty parameter, the optimal value of it is chosen. Finally, Athey and Imbens (2016) show that the goodness-of-fit criterion for the cross-validation can be written as $-MMSE_{\mu}(F^{tr,tr}, F^{tr,cv}, \Pi)$. It is important to highlight that smaller leaves lead to noisier estimates of leaf means. To account for this fact, the criterion will lead to larger average MSE across the cross-validation samples when the estimates are noisier, because the smaller leaf penalty gives us deeper trees and thus smaller leaves, that as we said before lead to noisier estimates.

4.3 Honest Approach

The Honest target approach created by Athey and Imbens (2016) differs from the conventional CART in two main ways. Firstly, use two different samples to be able to separate between the construction of the partition and estimation of the effects within leaves. (Training sample F^{tr} and Test sample F^{te}) for the job. This change modifies the cross-validation and splitting criteria, since F^{est} is treated as a random variable in the tree creation phase, the results of the estimates using F^{est} are unbiased. Second, it is focused on estimating conditional average treatment effects instead of predicting outcomes. We can begin by expanding the $EMMSE_{\mu}(\Pi)$ and using the property $E_F[\hat{\mu}(x; F, \Pi] = \mu(x; \Pi)$:

$$\begin{aligned} -EMMSE_{\mu}(\Pi) = &-E_{(Y_i, X_i), F^{est}}[(Y_i - \mu(X_i; \Pi))^2 - Y_i^2] \\ &-E_{X_i, F^{est}}[(\hat{\mu}(X_i; F^{est}, \Pi) - \mu(X_i; \Pi))^2] \end{aligned}$$

$$-EMMSE_{\mu}(\Pi) = E_{X_i}[\mu^2(X_i;\Pi)] - E_{X_i,F^{est}}[V(\hat{\mu}^2(X_i;F^{est},\Pi)] \tag{17}$$

Then, it is necessary to estimate the $-EMMSE\mu(\Pi)$ from the training sample F^{tr} and with the sample size of the estimation N^{est}. First, in order to estimate the second term of the equation (17) Athey and Imbens (2016) explain that within each leaf of the tree there is an unbiased estimator for the variance of the estimated mean in that leaf. So, the variance estimator on the training sample of $\hat{\mu}^2(X_i;F^{est},\Pi)$, will be:

$$\hat{V}(\hat{\mu}^2(X_i;F^{est},\Pi)) \equiv \frac{S^2_{F^{tr}}(l(x;\Pi))}{N^{est}(l(x;\Pi))} \tag{18}$$

Where $S^2_{F^{tr}}(l)$ is the within-leaf variance. Then (17) Athey and Imbens (2016) assume that the leaf shares are approximately equal in both samples, making possible to weight the variance estimator by the leaf shares p_l:

$$\hat{E}[V(\hat{\mu}^2(X_i;F^{est},\Pi))|i \in F^{te}] \equiv \frac{1}{N^{est}} \sum_{l\in\Pi} S^2_{F^{tr}}(l) \tag{19}$$

Second, for the estimate of the average of the squared outcome - the first term of equation (17)- Athey and Imbens used the square of the estimated means in the training sample $\hat{\mu}^2(x;F^{tr},\Pi)$, minus an estimate of its variance,

$$\hat{E}[\mu^2(x;\Pi)] = \hat{\mu}^2(x;F^{tr},\Pi) - \frac{S^2_{F^{tr}}(l(x;\Pi))}{N^{tr}(l(x;\Pi))} \tag{20}$$

Combining equations (17) with (19) and (20) we had the following unbiased estimator of $EMMSE_{\mu}(\Pi)$:

$$-\widehat{EMMSE}_{\mu}(F^{tr},N^{est},\Pi) \equiv \frac{1}{N^{tr}} \sum_{i\in F^{tr}} \hat{\mu}^2(X_i;F^{tr},\Pi) - \left(\frac{1}{N^{tr}} + \frac{1}{N^{est}}\right) \sum_{l\in\Pi} S^2_{F^{tr}}(l(x;\Pi)) \tag{21}$$

Finally, we can define the honest criterion that the algorithm will maximize as

$$Q^H \equiv -E_{F^{te},F^{est},F^{tr}}[MMSE_{\mu}(F^{te},F^{est},\pi(F^{tr}))] \tag{22}$$

The difference between the adaptive and the honest approach is in the terms involving the variance. Because, honest criterion penalizes small leaf size, as show in (17) Athey and Imbens (2016) for a given x, $S^2_{F^{tr}}(l(x;\Pi))$ is proportional to the MMSE within the associated leaf, thus, the difference came from how the within-leaf MMSE is weighted.

Nevertheless, the unbiased estimator of $\widehat{EMMSE}_{\mu}(F^{tr},N^{est},\Pi)$, fails when we used it repeatedly to evaluate splits using recursive partitioning on the training data F^{tr}. This happens because in each of the splits the observations with extreme outcome tend to group. The latter provokes that after the training data is divided, the within-leaf sample variance of observations in that data is on average lower than in a new independent sample. Hence, the way of fix this problem is using only outcomes for units from the cross-validation sample $F^{tr,cv}$:

$$-\widehat{EMMSE}_{\mu}(F^{tr,cv},N^{est},\Pi) \tag{23}$$

Now the estimator for the honest criterion is unbiased for fixed Π, despite it could have higher variance than $MMSE_{\mu}(F^{tr,cv},F^{tr,tr},\Pi)$ due to a small size of the cross-validation sample.

4.4 The Honest Approach and Treatment Effects

The already developed algorithm is primarily focused on estimating conditional population means, so it has to be changed to estimate conditional average treatment effects (normally called as "Causal Tree"). The estimation of those effects has some problems because of the value of the treatment effect whose conditional mean we wish to estimate, it is not observed. To be able to fix that problem, we now observe the vectors $[Y_i^{obs}, X_i, W_i]$, hence for each sample F let F_{treat} be the subsample of treated and $F_{control}$ be the subsample of control. Also, let $p = N_{treat}/N$ be the share of treated units, this means that the population average outcome and the average causal effects are:

$$\mu(w, x; \Pi) \equiv E[Y_i(w)|X_i \in l(x; \Pi)] \tag{24}$$

$$t(x; \Pi) \equiv E[Y_i(1) - Y_i(0)|X_i \in l(x; \Pi)] = \mu(1, x; \Pi) - \mu(0, x; \Pi) \tag{25}$$

The estimated counterparts for both equations are:

$$\hat{\mu}(w, x; F, \Pi) \equiv \frac{1}{N(\{i \in F_w : X_i \in l(x; \Pi)\}} \sum_{i \in F_w : X_i \in l(x;\Pi)} Y_i^{obs} \tag{26}$$

$$\hat{t}(x; \Pi) \equiv \hat{\mu}(1, x; F, \Pi) - \hat{\mu}(0, x; F, \Pi) \tag{27}$$

Then, the MMSE for the treatment effects will be:

$$MMSE_t(F^{te}, F^{est}, \Pi) \equiv \frac{1}{N(F^{te})} \sum_{i \in F^{te}} \{(t_i - \hat{t}(X_i; F^{est}, \Pi))^2 - t_i^2\} \tag{28}$$

Now $EMMSE_t(\Pi)$ it can be obtained from the expectation over the estimation and test samples:

$$EMMSE_t(\Pi) \equiv E_{F^{te}, F^{est}}[MMSE_t(F^{te}, F^{est}, \Pi)] \tag{29}$$

But, this Honest approach EMMSE, also have to be modified for treatment effects.

$$-EMMSE_t(\Pi) = E_{X_i}[t^2(X_i; \Pi)] - E_{F^{est}, X_i}[V(\hat{t}^2(X_i; F^{est}, \Pi))] \tag{30}$$

Finally the components of the expectation can be estimated using only the training sample, creating a criterion that only depends on S^{tr} and N^{est}.

$$-\widehat{EMMSE}_t(F^{tr}, N^{est}, \Pi) \equiv \frac{1}{N^{tr}} \sum_{i \in F^{tr}} \hat{t}^2(X_i; F^{tr}, \Pi) - \left(\frac{1}{N^{tr}} + \frac{1}{N^{est}}\right) \sum_{l \in \Pi} \left(\frac{S^2_{F^{tr}_{treat}}(l)}{p} + \frac{S^2_{F^{tr}_{control}}(l)}{1-p}\right)$$

Hence for cross-validation the same expression is used, but now with the cross-validation sample: $-\widehat{EMMSE}_t(F^{tr,cv}, N^{est}, \Pi)$. Professors (17) Athey and Imbens (2016), explain that this expression is similar to the criteria proposed for the Honest version of CART in the previous section. This criterion is focused on penalizing a partition that generates greater variance in the within-leaves estimator and promotes a partition that finds strong heterogeneity in the estimator of the treatment effects. In relation to the above, Athey and Imbens (2016) emphasize that the difference between the prediction and the estimation of treatment effects is that in the case of prediction the two terms tend to choose characteristics that predict the heterogeneity of the results. On the other hand, in the case of treatment effects, the two terms focus on different types of characteristics. There is the possibility that the variance of the estimator of the treatment effects is reduced when a split is incorporated, this occurs regardless of whether the following leaves have the same average treatment effect. All of the above leads to the results that in cases where it is a split, the results obtained more homogenous leaves and logically lower variance estimates in the means of the treatment and control groups.

4.5 Implementation of the DID methodology with Causal Honest Tree

The most important contribution of this document is the ability to implement the DID methodology enhance with machine learning algorithms. As appointed in the equation (3) what DID does is to study the differences between treatment and control group before and after the shock and then subtract them, obtaining the final effect of the treatment controlling for observable and for temporary variations:

$$Treatment_{DID} = (E[Y_{it+1}|T=1] - E[Y_{it+1}|T=0]) - (E[Y_{it}|T=1] - E[Y_{it}|T=0]) \tag{31}$$

On the other hand, the estimator of the Causal Tree is:

$$Treatment_{CausalTree} = \hat{\mu}(1, x; F, \Pi) - \hat{\mu}(0, x; F, \Pi) \tag{32}$$

So, if we assume that $\hat{\mu}(1, x; F, \Pi) - \hat{\mu}(0, x; F, \Pi)$ is equal to $E[Y_i|T=1] - E[Y_i|T=0]$ we can combine both equations, creating a Causal-DID-Tree estimator:

$$\begin{gathered} C - DID - T_{est} = \\ (E[Y_{it+1}|T=1] - E[Y_{it+1}|T=0]) - (E[Y_{it}|T=1] - E[Y_{it}|T=0]) = \\ (\hat{\mu}_{t+1}(1, x; F, \Pi) - \hat{\mu}_{t+1}(0, x; F, \Pi)) - (\hat{\mu}_t(1, x; F, \Pi) - \hat{\mu}_t(0, x; F, \Pi)) = \\ \hat{t}_{t+1}(x; \Pi) - \hat{t}_t(x; \Pi) \end{gathered}$$

As we can see in the previous equation, the Causal-DID-Tree estimator is could be defined just like the difference of two Causal Tree (CT) estimator[3]:

$$C - DID - T_{est} = CT_{t+1} - CT_t \tag{33}$$

The use of the previous estimator allows us to obtain a DID treatment effect estimation for each observation. However, one of the problems with the trees is that they are sensitive to the samples, that is, what portion of the data was used for the training samples and which was used to obtain the treatment effect. To solve this problem, the process will be replicated N times - for our main case will be 1000 times - until enough observations are reached to obtain the normal distribution properties of the results. The capacity of obtaining confidence intervals for C-DID-T, is still in process, but it could be assumed that if the results of each CT is statistically significant, the subtraction of two significant results we still be significant.

5 Data experimentation

In order to know if the proposed methodology really finds the effect of the treatment and if it has efficiency gains with respect to OLS, different data generating processes will be simulated which will receive a treatment shock in a certain period of time and then the results will be extracted and compare. Therefore, this Section is divided into three: Subsection 5.1 will explain the simulations that were made and what metrics will be extracted from the results; Subsection 5.2 will show and discuss the results obtained for each data generating process; finally, Subsection 5.3 will check the robustness of the results - mainly provided by changes in the original data-.

[3] It is important to note that this method can be used with any causal machine learning algorithm, since we will be able to obtain each section of the DID and replicate it an amount N times.

5.1 Simulation Description

To study the behavior of the algorithm along the wide variety of data that usually exist in real life, three different situations will be simulated for three different amounts of data (1000, 4000 and 9000 observations). So there will be a total of 9 different databases. The three situations chosen are thought in a complete linear context, in a non-linear heterogeneous effects and in a case of heterogeneous treatment effects that are related to other independent variables. In a way of structure, all models have four independent variables X_k (with K for 1 to 4) that have a standard normal distribution ($X_k \sim N(0,1)$). In addition, all models have an error $r_1 \sim N(0,1)$, both heterogeneous treatment effects models have a white noise error correlated with the treatment $r_3 \sim N(0,2)$ and the treatment (T) is randomly assigned with an uniform function that takes value 0 or 1 with 50% probability. Finally, there is a constant time effect for (Time) that has a value of 1 for all units and it interacts with a white noise $r_2 \sim N(0,1)$. [4]

The outcome variable before the treatment effect (Y_0) has a difference in level between treatment and control units. Moreover, for the linear model equation (33) represents its base state (Y_0^L), while equation (34) represents the base state for the non-linear and heterogeneous treatment effects model (Y_0^{NL-H}):

$$Y_{0i}^{L} \equiv 1 + (1/2) * X_{1i} + (1/2) * X_{2i} + (1/2) * X_{3i} + (1/2) * X_{4i} + (1/2) * T_i + r_{1i} \tag{34}$$

$$Y_{0i}^{NL-H} \equiv 1 + (1/2) * X_{1i}^2 + (1/2) * X_{2i} + (1/2) * X_{3i}^2 + X_{4i} + T_i + r_{1i} \tag{35}$$

Then the post-treatment outcome variables (Y_1) contain the previously mentioned time trend and the effect of the treatment, that is five time the standard deviation (σ) of the base state, this means:

1.- **Linear Model:** $Y_{1i}^{L} \equiv Y_{0i}^{L} + 5 * \sigma_{Y_{oi}^{L}} * T_i + (Time_i * r_{2i} + 1)$

2.- **Non-Linear Model:** $Y_{1i}^{NL} \equiv Y_{0i}^{NL-H} + 5 * \sigma_{Y_{oi}^{NL-H}} * (T_i * r3)^2 + (Time_i * r_{2i} + 1)$

3.- **Heterogeneous Treatment Model:** $Y_{1i}^{H} \equiv Y_{0i}^{NL-H} + 5 * \sigma_{Y_{oi}^{NL-H}} * ((1/2) * X_{1i} * T_i + (1/2) * r_{3i} * T_i) + (Time_i * r_{2i} + 1)$

Finally, the treatment effect for the DID methodology in each of the models is as follows:

1.- **Treatment in the Linear Model:** $Z_i^{L} \equiv 5 * \sigma_{Y_{oi}} * T_i$

2.- **Treatment in the Non-Linear Model:** $Z_i^{NL} \equiv 5 * \sigma_{Y_{oi}^{NL-H}} * (T_i * r3)^2$

3.- **Treatment in the Heterogeneous Treatment Model:** $Z_i^{H} \equiv 5 * \sigma_{Y_{oi}^{NL-H}} * ((1/2) * X_{1i} * T_i + (1/2) * r_{3i} * T_i)$

The Table I shows the mean, median, standard deviation, minimum and maximum for the outcome variables before (Y_{0i}) and after (Y_{1i}) treatment and for the treatment effect variable (Z_i). It can be seen that for all models there is an increase in the standard deviation once the treatment has occurred. Along with this, it can be observed that the treatment is distributed as mentioned above, that means that since it is assigned with a probability of 50%, it does not necessarily have perfectly equal groups between treatment and control, making the means and median vary a little

[4] The simulation process was made in Stata, for more details refer to appendix section 7.1

bit depending on which group has a greater number of units. Finally, it is important to say that for the case of heterogeneous effects the treatment average is near to zero since the effects of the treatment come from two processes (X_{1i} and r_{3i}) that have this mean.

Table I: Descriptive statistics of the simulations

			Mean	Median	Std. Desv.	Min	Max
Linear Model	1000	Y_0	1,2090	1,2606	1,4719	-4,5489	4,9888
		Y_1	2,3322	2,3811	1,8247	-4,2129	7,4362
		Z	0,1122	0,0000	0,1163	0,0000	0,2327
	4000	Y_0	1,2413	1,2425	1,4749	-3,8552	6,1565
		Y_1	2,2914	2,3083	1,7763	-4,4516	8,3556
		Z	0,0567	0,0000	0,0583	0,0000	0,1166
	9000	Y_0	1,2554	1,2676	1,4424	-4,3257	7,0051
		Y_1	2,2916	2,3121	1,7525	-3,6572	9,7705
		Z	0,0381	0,0760	0,0380	0,0000	0,0760
Non Linear	1000	Y_0	2,4591	2,2890	1,9095	-4,3268	8,9617
		Y_1	4,0448	3,8397	2,6494	-4,1496	17,048
		Z	0,5747	0,0000	1,3097	0,0000	11,557
	4000	Y_0	2,5256	2,4541	1,9325	-4,0641	11,502
		Y_1	3,8173	3,6866	2,3406	-3,7463	13,252
		Z	0,2983	0,0000	0,6746	0,0000	8,2629
	9000	Y_0	2,5353	2,4422	1,9043	-3,9606	1,2800
		Y_1	3,7408	3,6670	2,2386	-3,7839	14,537
		Z	0,2073	0,0000	0,4642	0,0000	5,8973
Heterogeneous Treatment	1000	Y_0	2,4591	2,2890	1,9095	-4,3268	8,9617
		Y_1	3,4656	3,3941	2,1856	-4,1496	10,778
		Z	-0,0045	0,0000	0,2359	-1,1061	0,9294
	4000	Y_0	2,5256	2,4541	1,9325	-4,0641	11,502
		Y_1	3,5221	3,4373	2,1728	-4,5872	12,827
		Z	0,0031	0,0000	0,1197	-0,6846	0,6229
	9000	Y_0	2,5353	2,4422	1,9043	-3,9606	12,800
		Y_1	3,5328	3,4733	2,1411	-3,7839	14,537
		Z	-0,0007	0,0000	0,0806	-0,3898	0,4289

Note: All the results were made by the author with simulated data.

After the explanation about the data generating process and how the distribution of the different dataset are, now it has to be understood how the results are going to be compared between the OLS and the C-DID-T. In order to be able to compare the different estimates the Mean quadratic error (MSE) is going to be used, and its two components, the variance and the squared bias. In the case of OLS, the extraction of this metrics it is really simple because it is a linear estimation model and the real value of the treatment is known, so the MSE is just take the variance of the estimator and added to the quadratic Bias of the estimator:

$$MSE(\hat{Z}) = Variance(\hat{Z}) + Bias(\hat{Z})^2 = E_{\hat{Z}}[(\hat{Z} - E_{\hat{Z}}[\hat{Z}])^2] + (E_{\hat{Z}}[\hat{Z}] - Z)^2 \tag{36}$$

Where the left part of the equation (the variance) is already computed by Stata and the right side (the bias) for any OLS estimation is the square difference between the estimator and the real value.

On the other hand, in the case of the C-DID-T obtaining the variance for the MSE is somewhat more complex. This is because, when estimating heterogeneous treatment effects for each leaf of the tree, any global variance for the estimator will be biased. Therefore, one way to solve it is by taking the 1000 replications that our C-DID-T estimator generates for each observation and obtain the global variance of the estimator through the variances of each observation. However, this generates a computational problem given that for a dataset of N observations with 1000 replications, there would be N variables with 1000 observations to which it must not only calculate the variance but also the covariance of the system. To address this problem, the Principal Components Analysis (PCA) will be use and it will simplify the large number of variables to a small group which represents a percentage X of the total variance. The latter, will eliminate the problem of covariances since the resulting vectors will be orthogonal by construction, this means that by the Law of Total Variance our final variance will be just the sum of all the individual variances.

Briefly, the PCA uses orthogonal transformations to change or convert a dataset of correlated variables into its "principal components", i.e. a set of linearly uncorrelated variables. In example, if there is a dataset with N variables and 1000 observations each, the number of principal components will be the min(N - 1, 1000). The transformation creates a first component that has the largest part of the variance of the dataset, and then the next components will be orthogonal to the first and will have another part of the variance. In the case of this document, the number of PCA is the minimum that has at least the 95% of the variance. It is important to highlight that the PCA is sensible to the relative scaling of the original variables. In the case of our simulation, the base variables and the errors are all distributed normal standard so the scaling should not be a problem.

5.2 Simulation Results

Once the variance is obtained with the PCA, and given that the the real value of the treatment is known, the MSE can be estimated following equation (36). The Table II shows the results of the methodology followed until this moment, the number of replication of each of the models in the C-DID-T methodology was 1000. The left part of the table show the different metrics that are used to compare each model. On the other hand the right panel shows if the value of the C-DID-Tree is lower than the value of the OLS methodology, also in the case of the P value not only shows if the value is smaller or not, but also shows if the OLS results were significant.

First, from a general point of view it can be seen that the C-DID-T has worse aggregated results than the OLS methodology, this means that our C-DID-T have bigger MSE, Bias and Variance. Nevertheless, the algorithm is capable of finding a treatment in cases where OLS is not able. For example, if the data is normalized and have a heterogeneous process with correlation with other variable treatment, this means that the effect of the treatment is different depending your level of X, OLS will not be able to found a Treatment.

Second, even when it can be seen that as expected OLS Bias and Variance go down when the data get bigger, for Causal-DID-Tree this is not correct, because with bigger amounts of data its get smaller bias but larger variance, these strange results occur because of the way that the variance

is calculated. As mentioned in the last section the variance in our model is calculated through the variance of the replication for each observation, then these original observations in the dataset are variables which covariate between each other, so if there are more original observations there will be more variables and then more variance.

Table II: Results of the simulations

	Obs.	Variance	P value	Square Bias	Bias	MSE	MSE Winning	Bias Winning
Ordinary Least Squares								
Linear	1000	0,0132	0,0197	0,0012	0,0351	0,0144	No	No
	4000	0,0030	0,0291	0,0000	0,0026	0,0030	No	No
	9000	0,0013	0,0141	0,0002	0,0129	0,0015	No	No
Non Linear	1000	0,0268	0,0000	2,1396	1,4628	2,1664	No	No
	4000	0,0056	0,0000	0,5558	0,7455	0,5614	No	No
	9000	0,0023	0,0000	0,2637	0,5135	0,2660	No	No
Heterogeneous Treatment	1000	0,0214	0,8604	0,0565	0,2377	0,0780	No	No
	4000	0,0052	0,9011	0,0144	0,1198	0,0196	No	No
	9000	0,0023	0,8089	0,0067	0,0816	0,0089	No	No
Causal - DID - Tree							Variance Winning	P value Winning
Linear	1000	0,3889	0,0000	0,4016	0,6337	0,7905	No	Yes
	4000	0,7809	0,0000	0,3302	0,5746	1,1111	No	Yes
	9000	0,8700	0,0000	0,2680	0,5177	1,1380	No	Yes
Non Linear	1000	0,3879	0,0000	2,6289	1,6214	3,0169	No	No
	4000	0,7789	0,0000	1,1579	1,0760	1,9368	No	Yes
	9000	0,8669	0,0000	0,7657	0,8751	1,6327	No	Yes
Heterogeneous Treatment	1000	0,3889	0,0000	0,6947	0,8335	1,0837	No	OLS - NST
	4000	0,7809	0,0000	0,6764	0,8225	1,4574	No	OLS - NST
	9000	0,8660	0,0000	0,5614	0,7493	1,4274	No	OLS - NST

Note: All the results were made by the author with simulated data. The left panel shows the results of the different metrics calculated by the previous methodologies. The right panel shows if the value of the algorithm is lower than the value of the OLS methodology. The first column (Obs.) is the number of observation of the dataset. The answer "OLS - NST" means that the OLS treatment parameter for that case was not statistically significant.

A more deep analysis is possible to perform on these results, since our Square Bias is calculated as the average of the Square Bias of each repetition, it can be obtained the confidence interval and see if the results found between OLS and C-DID-T are statistically different. The Table III, shows for each model the confidence interval of the C-DID-T Square Bias and the values of that metric in the case of OLS and the algorithm. The results demonstrated that the OLS and C-DID-T values are statistically different with a 99% probability for all models and all sample sizes, confirming what was found previously.

Table III: Confident Intervals

		Square Bias Confidence Interval					
	Obs.	Inf. 1%	Sup. 1%	Inf. 10%	Sup. 10%	C-DID-T Mean	OLS Value
Linear	1000	0,3839	0,4193	0,3903	0,4129	0,4016	0,0012
	4000	0,3206	0,3397	0,3240	0,3363	0,3302	0,0000
	9000	0,2606	0,2755	0,2633	0,2728	0,2680	0,0002
Non Linear	1000	2,5915	2,6664	2,6050	2,6528	2,6289	2,1396
	4000	1,1409	1,1748	1,1471	1,1687	1,1579	0,5558
	9000	0,7513	0,7802	0,7565	0,7750	0,7657	0,2637
Heterogeneous Treatment	1000	0,6686	0,7209	0,6780	0,7114	0,6947	0,0565
	4000	0,6607	0,6922	0,6664	0,6865	0,6764	0,0144
	9000	0,5483	0,5746	0,5530	0,5698	0,5614	0,0067

Note: All the results were made by the author with simulated data. The table shows the confidence interval of the C-DID-T Square Bias and the values of that metric in the case of OLS and the algorithm. The first column (Obs.) is the number of observation of the dataset.

Finally, the Table IV compare the relative efficiency of the C-DID-T with respect to OLS. The results in the table are calculated as the results of the C-DID-T divided by the results of OLS minus one. From this table, three main ideas can be concluded. First, as previously mentioned, the variance is increasing in a explosive way; this is most likely due to the way in which the variance is calculated. Second, it is observed that the bias of this algorithm is quite close in the Non-Linear model (11% less efficient) but if the data increase OLS is getting more efficient faster than the C-DID-T, that is really strange because it is supposed that ML algorithms have big rates of improvement with bigger amounts of data. Third, for the case of heterogeneous effect related to other independent variables the algorithm has quite big deficiency.

Table IV: Comparison Summary

	Obs.	Variance	Square Bias	Bias	MSE
1000 Repetitions - X $\sim$ N(0,1) Model					
No - Linear	1000	1349%	23%	11%	39%
	4000	13823%	108%	44%	245%
	9000	36916%	190%	70%	514%
Heterogeneous Treatment	1000	1715%	1129%	251%	1290%
	4000	14858%	4611%	586%	7344%
	9000	38189%	8342%	819%	15916%

Note: All the results were made by the author with simulated data. Each of the results were calculate by the previous methodologies. The first column (Obs.) is the number of observation of the dataset. The numbers of the table were calculated as the results of the Causal-DID-Tree divided by the results of OLS minus one.

Looking at these first results, it is clear that there is no gain in the case of linear effects, because the main use of the Causal-DID-Tree algorithms is to find treatment in more complex environment of data and the fact that it has specific treatment effects for the different groups of subjects inside our data. The latter, is demonstrated for the case of the Non linear heterogeneous effects where as show in Table I the treatment goes from 0 to 11 in the case of 1000 observations and the OLS estimator is 1,2 for all individuals what it is a really big difference from 11, instead the Causal-DID-tree have

a different value for each of the individuals of the sample. The next subsection will change the structure of the data and the number of replication, waiting for find situations where the C-DID-T algorithm improve its results.

5.3 Robustness checks

This section is going to replicate the Non-linear and the heterogeneous effect models but with some changes. First, we are going to change the distribution of the X_k from N(0,1) to a N(5,1), because there is the possibility that OLS rejects the effects of the treatment if the mean of the processes is zero. The second modification is to re-run the dataset with distribution of mean zero and mean five but instead of 1000 replications in the DID, it will have 3000 replications. The summary of the results are presented in Table V, the rest of the results are in Tables VII-XII in the Results Appendix.

Table V: Results Summary

	Obs.	MSE Winning			Bias Winning			Statistically different	
		(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)
Non Linear	1000	No	No	No	No	No	No	Yes	Yes
	4000	No	No	No	No	No	No	Yes	Yes
	9000	No	No	No	No	No	No	Yes	Yes
Heterogeneous Treatment	1000	No	No	No	No	No	No	Yes	Yes
	4000	No	No	No	No	No	No	Yes	Yes
	9000	No	No	No	No	No	No	Yes	Yes
		Variance Winning			P value Winning				
		(1)	(2)	(3)	(1)	(2)	(3)	(3)	-
Non Linear	1000	No	No	No	No	No	No	Yes	-
	4000	No	No	No	No	Yes	No	Yes	-
	9000	No	No	No	No	Yes	No	Yes	-
Heterogeneous Treatment	1000	No	No	No	No	OLS - NST	No	Yes	-
	4000	No	No	No	No	OLS - NST	No	Yes	-
	9000	No	No	No	No	OLS - NST	No	Yes	-

Note: All the results were made by the author with simulated data. Each of the results were calculated by the previous methodologies. For the left panel the table shows if the value of the algorithm is smaller than the value of OLS. For the right panel the table shows if the results of OLS are different statistically from the ones of the C-DID-T. The first column (Obs.) is the number of observation of the dataset. Model (1) is with 1000 replication and X_k distributed N(5,1). Model (2) is with 3000 replication and X_k distributed N(0,1). Model (3) is with 3000 replication and X_k distributed N(5,1). The answer "OLS - NST" means that the OLS treatment parameter for that case was not statistically significant.

The results behave in the same way as those in the previous subsection, the algorithm of Causal-DID-Tree is still less efficient regardless of the form of distribution of the data or the number of repetitions that are used for the algorithm. In addition, the results show that with a higher number of repetitions the MSE falls, both on the side of the variance and on the side of the bias. However, this fall is not strong enough to be more efficient than OLS. Another point to highlight is the fact that for processes with a mean of zero or with standardized data the heterogeneous treatment in

OLS is not significant but changing the mean of the distribution changes this fact. Therefore, the fact that C-DID-T finds significant results for the treatment regardless of the distribution of data used is maintained. Finally, confidence intervals continue to show that the value of OLS is not only minor, but it is statistically different from that of C-DID-T.

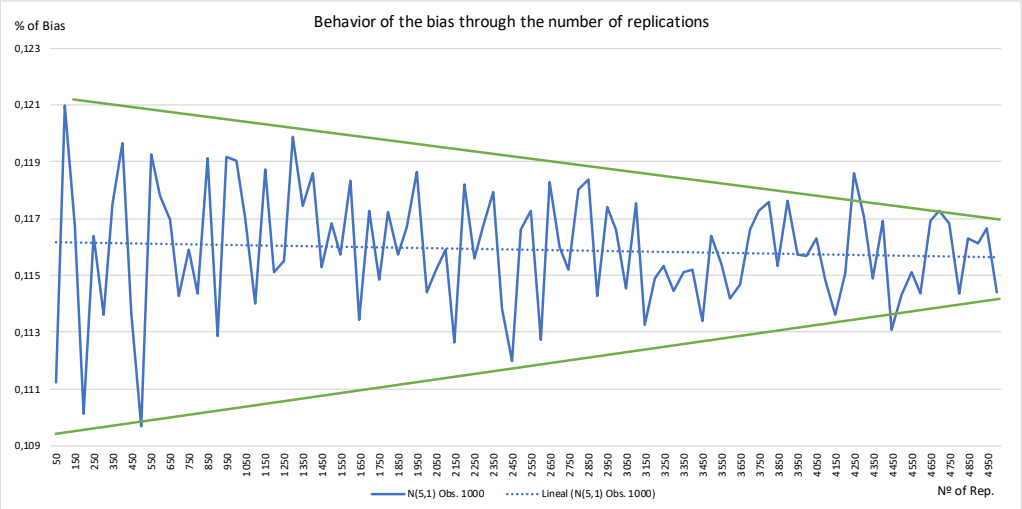
Table VI: Efficiency comparison of the models

	Obs.	Variance	Square Bias	Bias	MSE
1000 Repetitions - X ∼ N(5,1) Model					
Non Linear	1000	279%	24%	11%	25%
	4000	7164%	69%	30%	78%
	9000	25146%	117%	47%	139%
Heterogeneous Treatment	1000	1481%	139%	55%	146%
	4000	14266%	358%	114%	412%
	9000	37689%	760%	193%	906%
3000 Repetitions - X ∼ N(0,1) Model					
Non Linear	1000	450%	22%	11%	27%
	4000	8984%	107%	44%	196%
	9000	30726%	191%	71%	460%
Heterogeneous Treatment	1000	587%	1114%	248%	969%
	4000	9118%	4564%	583%	5778%
	9000	31823%	8124%	807%	14138%
3000 Repetitions - X ∼ N(5,1) Model					
Non Linear	1000	44%	25%	12%	25%
	4000	4538%	68%	30%	74%
	9000	20875%	27%	13%	46%
Heterogeneous Treatment	1000	497%	141%	55%	143%
	4000	8882%	361%	115%	395%
	9000	31297%	760%	193%	880%

Note: All the results were made by the author with simulated data. Each of the results were calculated by the previous methodologies for each of the different types of datasets. The first column (Obs.) is the number of observation of the dataset. The numbers of the table were calculated as the results of the Causal-DID-Tree divided by the results of OLS minus one.

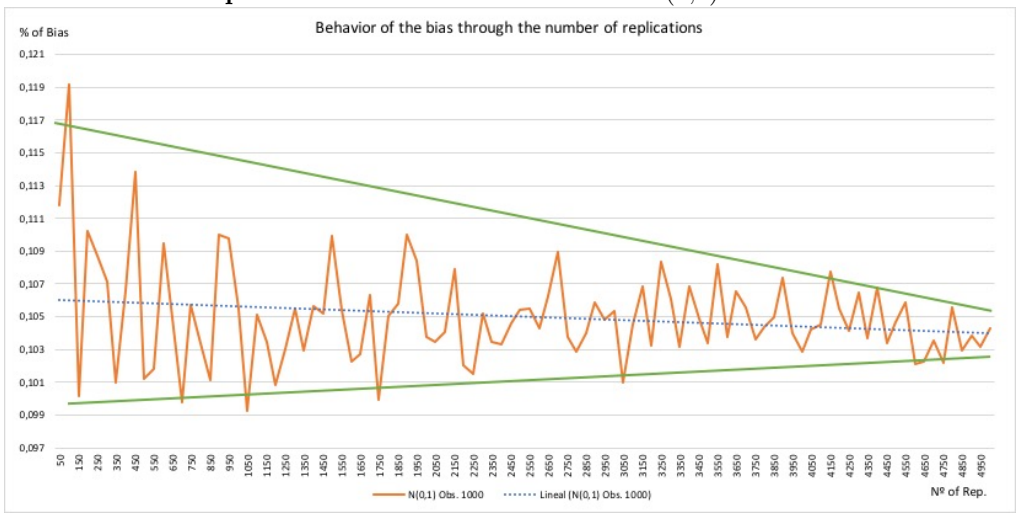
In the same manner that as in the previous section, Table VI displays the difference in efficiency between the estimators for the different groups of dataset. The results maintain the previous conclusions but with some small changes. First, for the models with distributions of mean different from zero the C-DID-T is more effective relative to OLS that the case of mean zero. Nevertheless, the algorithms never get a difference in efficiency lower than 11%. Second, even with the problem of the explosion in variance, with 3000 replications and with a mean different from zero, the MSE of the Non-linear model is only a 46% less efficient than the case of OLS. Finally, when it is compared the results of the dataset of distribution with mean zero and 9000 observations, for the cases of 1000 and 3000 replications there is not much gain from the side of the bias, but from the side of the variance there is difference of almost 6190% between the first and the second results. Moreover, the previous conclusion remains when it is analyzed with the mean different from zero.

Graph I: Behavior of Bias for dataset N(5,1) Obs. 1000



Note: All the results were made by the author with simulated data. The results were calculated for the datasets N(5,1) with 1000 observations. The numbers of the graph were calculated as the results of the Bias of the Causal-DID-Tree divided by the bias of OLS minus one.

Graph II: Behavior of Bias for dataset N(0,1) Obs. 1000



Note: All the results were made by the author with simulated data. The results were calculated for the datasets N(0,1) with 1000 observations. The numbers of the graph were calculated as the results of the Bias of the Causal-DID-Tree divided by the bias of OLS minus one.

The final analysis, that we are able to do is to see how the algorithm behaves along the different number of replications. To do so, the Graph I and II, show the movement of the Bias relative to OLS from 50 to 5000 replications for the two dataset where the bias was more close to the OLS results

-N(0,1) and N(5,1) with 1000 replications each- [5]. The Graph I shows the number of replications doesn't make that the algorithm behave better related to bias -because the trend line is flat-, but it is possible to observe that the bias converges to its true value when the number of replications goes up. In the same line, Graph II supports the results of Graph I, showing a big convergence from the first number of replications (50) until the last one (5000). Also, the trend line of the Graph II has a negative slope, this means that the bias is getting smaller through the number of replications. Finally, for the case of N(5,1) and 9000 observations (Graph III), the trend goes up instead of down in the case of bias, but the conclusion related with convergence remains. Therefore, we can conclude that with a bigger number of replications the algorithm converges to its real value, but in relation with the improvement of the results related to OLS the results are inconclusive.

Having in consideration the results and the robustness checks that we did in the last two section, there is a key question to answer, Why the algorithm is not having better results than the OLS? First, in consideration of the variance we are not able to really obtain a correct measure of it for the C-DID-T, so the results of OLS and C-DID-T in the variance and the MSE (because the variance is part of the MSE) are not comparable. Second, from the bias point of view, we can observe that the results are different between both methodologies, and one explanation for this is the fact the each CT give us mean leaf estimator, and the C-DID-T use the leaves estimator from the CT and subtracts them, creating a big amount of noise that came from the fact the depending the iteration of the CT the observation could be concentrate in different leaves, provoking that sometimes the difference between the leaves average of one tree and the other is considerably big. A way to solve this problem is to restructure our approach so it is able to create leaf estimator instead of individuals, minimizing the noise and of course the bias.

6 Conclusion

In recent years, the use of machine learning algorithms in the debate of the social sciences and specially in economics has been taking an increasingly important relevance. In a first stage, these tools were viewed with distrust or with little use for economists given that they have a high predictive capacity but a poor capacity to explain the processes that guide the results that are found. Thanks to the efforts of a series of academics, the use of Machine Learning tools to obtain causal effects has opened a new door in the incorporation of ML algorithms in economics. As already discussed in the previous sections, studies that use ML algorithms to obtain causal effects have been related in three fields of action. The first is the use of ML algorithms to reduce the number of covariates in the models. The second has been in the line of creating counterfactuals through matching processes or others methodologies. The third application of ML algorithms has been in the modification of them to obtain causal effects in cross-sectional data, this means, a causal effect in the classical logic of Rubin (1974). Therefore, the application of ML algorithms in Panel type data, that is, using ML algorithms to control for temporary effects and find causal effects is quite scarce and is still in a development stage.

Based on the above, this document was immersed in the study of causality and how the ML algorithms have managed to be modified to obtain causal effects, proposing the use of the algorithm of Honest Causal Tree by Athey and Imbens (2016) to estimate causal effects from the Difference in

[5] Also in the appendix, we present the results for the dataset N(5,1) with 9000 observations.

Difference methodology. To achieve the latter, we use two Honest Causal Trees to estimate each area of the DID estimator (pre and post treatment) and then we subtract the results from each tree, obtaining the causal effect of the treatment for each individual. Then, we repeat the process N times to eliminate the bias coming from the random selection of the training and estimation samples. Once the Causal-DID-Tree algorithm was developed, we compared its behavior with the classic results of OLS, with different types of data generating processes and different extensions of them (number of observations).

The results related to the algorithm are quite varied. First, the Causal-DID-Tree allows us to find an effect of the policy for each of the individuals, that is, unlike OLS that we have an average treatment effect with the C-DID-T we can understand how the treatment behaves to throughout the different groups of individuals of the population. Second, we see that when we have data distributions with heterogeneous treatments with zero mean and dependent of other variables of the model - the existence of collinearity between the treatment and some covariables-, OLS says that the policy is not statistically significant instead C-DID-T finds a significant result -under some assumptions-. Third, for cases of non-linear effects, the bias depending on the amount of data can vary from 11% to 71% more than that of OLS, that is very good for a first attempt. Fourth, the algorithm improves its results slower than OLS before the increase in data, the reason why this happens should still be investigated. Moreover, it can be seen that the behavior of the algorithm with a higher number of repetitions, does not lead to an improvement in its bias with respect to OLS, but more repetitions make the bias converges to its real value.Therefore, we believe that some changes in the algorithm must be done, e.g. moving from individual to leaves results could improve the performance of the algorithm.

The usual measure to compare how good is a methodology is the quadratic mean error (MSE), which is composed of the squared bias and the variance. Throughout this document, it was not possible to find a representative measure of the variance for the C-DID-T, since, using the PCA to obtain the variance, the value of it instead of decreasing with the number of observations it grows explosively. Therefore, the MSE is not representative or comparable, only the bias is. In addition to the problem with the variance estimator, the algorithm takes large amounts of time to be executed with high repetitions and observations, for example, for the case of 9000 observations and 3000 repetitions, it takes about 1 hour to be executed on a external server of 64 cores. Therefore, until having improvements in the efficiency of the algorithm code, it is quite difficult to see its behavior before large amounts of data. One way to fix this issue is by using different causal algorithms with the same methodology (e.g. Causal Tree, Causal Random Forest, among others.) and choosing the more effective one.

A final issue to comment is the use of different types of data, since throughout this document we only used three fairly simple models with variables that had normal distributions in most cases. Therefore, it would be interesting to review other types of treatment, such as Non-Linear heterogeneous effects correlated with some independent variable included in the model. The latter is necessary to be able to see the behavior of the algorithm with more realistic data. In this same line, we could test the algorithm through the replication of previous studies that use DID and see how effective it is in understanding public policies. However, it is important to point out that to apply this algorithm one must have highly balanced panels and therefore, prior to the use of the algorithm it is necessary to do some Matching process or another methodology that allows to create

counterfactuals.

The analysis of public policies through average effects have limited their understanding, among the diversity of individuals within a society. Added to this, the increase in government databases and the development of Machine Learning algorithms have made possible the beginning of the understanding of the effects of treatment in a much more heterogeneous way. It is in this line, that this document sought to continue developing the existing algorithms of ML and creates a way to use those in conjunction with the DID methodology. As Sir Isaac Newton (1675) expressed, "If I have seen further it is by standing on the shoulders of the Giants", so if any scientist, student or thinker wants to understand the world around him and generate change, he must be aware that it would not be possible without those who came before him and that any idea or research he does is just a little grain of sand on the beach of the scientific advance.

7 Results Appendix

Table VII: Results Model N(5,1) - Repetitions 1000

	Obs.	Variance	P value	Square Bias	Bias	MSE	MSE Winning	Bias Winning
Ordinary Least Squares								
Non Linear	1000	0,1026	0,0000	31,131	5,5796	31,234	No	No
	4000	0,0108	0,0000	8,3222	2,8848	8,3330	No	No
	9000	0,0034	0,0000	3,9198	1,9798	3,9232	No	No
Heterogeneous Treatment	1000	0,0247	0,0000	5,1815	2,2763	5,2061	No	No
	4000	0,0054	0,0000	1,3806	1,1750	1,3861	No	No
	9000	0,0023	0,0000	0,5803	0,7618	0,5826	No	No
Causal - DID - Tree							Variance Winning	P value Winning
Non Linear	1000	0,3889	0,0000	38,632	6,2155	39,021	No	No
	4000	0,7809	0,0000	14,064	3,7503	14,845	No	No
	9000	0,8690	0,0000	8,4979	2,9151	9,3669	No	No
Heterogeneous Treatment	1000	0,3899	0,0000	12,403	3,5219	12,793	No	No
	4000	0,7800	0,0000	6,3218	2,5143	7,1017	No	No
	9000	0,8690	0,0000	4,9905	2,2339	5,8595	No	No

Note: All the results were made by the author with simulated data. The left panel shows the results of the different metrics calculated by the previous methodologies. The right panel shows if the value of the algorithm is lower than the value of the OLS methodology. The first column (Obs.) is the number of observation of the dataset. The answer "OLS - NST" means that the OLS treatment parameter for that case was not statistically significant.

Table VIII: Confident Intervals Model N(5,1) - Repetitions 1000

		Square Bias Confidence Interval					
	Obs.	Inf. 1%	Sup. 1%	Inf. 10%	Sup. 10%	C-DID-T Mean	OLS Value
Non Linear	1000	38,137	39,128	38,316	38,949	38,632	31,131
	4000	13,906	14,223	13,963	14,166	14,065	8,3222
	9000	8,3873	8,6085	8,4272	8,5686	8,4979	3,9198
Heterogeneous Treatment	1000	12,131	12,677	12,229	12,578	12,404	5,1815
	4000	6,1888	6,4548	6,2368	6,4067	6,3218	1,3806
	9000	4,8844	5,0967	4,9227	5,0583	4,9905	0,5803

Note: All the results were made by the author with simulated data. The table show the the confidence interval of the C-DID-T Square Bias and the values of that metric in the case of OLS and the algorithm. The first column (Obs.) is the number of observation of the dataset.

Table IX: Confident Intervals Model N(0,1) - Repetitions 3000

		Square Bias Confidence Interval					
	Obs.	Inf. 1%	Sup. 1%	Inf. 10%	Sup. 10%	C-DID-T Mean	OLS Value
Non Linear	1000	2,5932	2,6365	2,6011	2,6287	2,6149	2,1396
	4000	1,1412	1,1620	1,1450	1,1583	1,1516	0,5558
	9000	0,7605	0,7767	0,7634	0,7737	0,7686	0,2637
Heterogeneous Treatment	1000	0,6709	0,7011	0,6764	0,6957	0,6860	0,0565
	4000	0,6607	0,6785	0,6639	0,6753	0,6696	0,0144
	9000	0,5393	0,5547	0,5420	0,5519	0,5470	0,0067

All the results were made by the author with simulated data. The table show the the confidence interval of the C-DID-T Square Bias and the values of that metric in the case of OLS and the algorithm. The first column (Obs.) is the number of observation of the dataset.

Table X: Results Model N(0,1) - Repetitions 3000

		Variance	P value	Square Bias	Bias	MSE	MSE Winning	Bias Winning
Ordinary Least Squares								
Non Linear	1000	0,0268	0,0000	2,1396	1,4628	2,1664	No	No
	4000	0,0056	0,0000	0,5558	0,7455	0,5614	No	No
	9000	0,0023	0,0000	0,2637	0,5135	0,2660	No	No
Heterogeneous Treatment	1000	0,0214	0,8604	0,0565	0,2377	0,0780	No	No
	4000	0,0052	0,9011	0,0144	0,1198	0,0196	No	No
	9000	0,0023	0,8089	0,0067	0,0816	0,0089	No	No
Causal - DID - Tree							Variance Winning	P value Winning
Non Linear	1000	0,1473	0,0000	2,6149	1,6171	2,7621	No	No
	4000	0,5082	0,0000	1,1516	1,0731	1,6598	No	Yes
	9000	0,7220	0,0000	0,7686	0,8767	1,4905	No	Yes
Heterogeneous Treatment	1000	0,1473	0,0000	0,6860	0,8283	0,8333	No	OLS - NST
	4000	0,4813	0,0000	0,6696	0,8183	1,1509	No	OLS - NST
	9000	0,7220	0,0000	0,5470	0,7396	1,2689	No	OLS - NST

All the results were made by the author with simulated data. The left panel shows the results of the different metrics calculated by the previous methodologies. The right panel shows if the value of the algorithm is lower than the value of the OLS methodology. The first column (Obs.) is the number of observation of the dataset. The answer "OLS - NST" means that the OLS treatment parameter for that case was not statistically significant.

Table XI: Confident Intervals Model N(5,1) - Repetitions 3000

		Square Bias Confidence Interval					
	Obs.	Inf. 1%	Sup. 1%	Inf. 10%	Sup. 10%	C-DID-T Mean	OLS Value
Non Linear	1000	38,635	39,202	38,737	39,100	38,765	31,131
	4000	13,919	14,097	13,951	14,065	14,008	8,3222
	9000	4,9281	5,0475	4,9496	5,0259	4,9878	3,9198
Heterogeneous Treatment	1000	12,338	12,670	12,398	12,610	12,504	5,1810
	4000	6,2915	6,4465	6,3195	6,4185	6,3690	1,3806
	9000	4,9281	5,0475	4,9496	5,0259	4,9878	0,5803

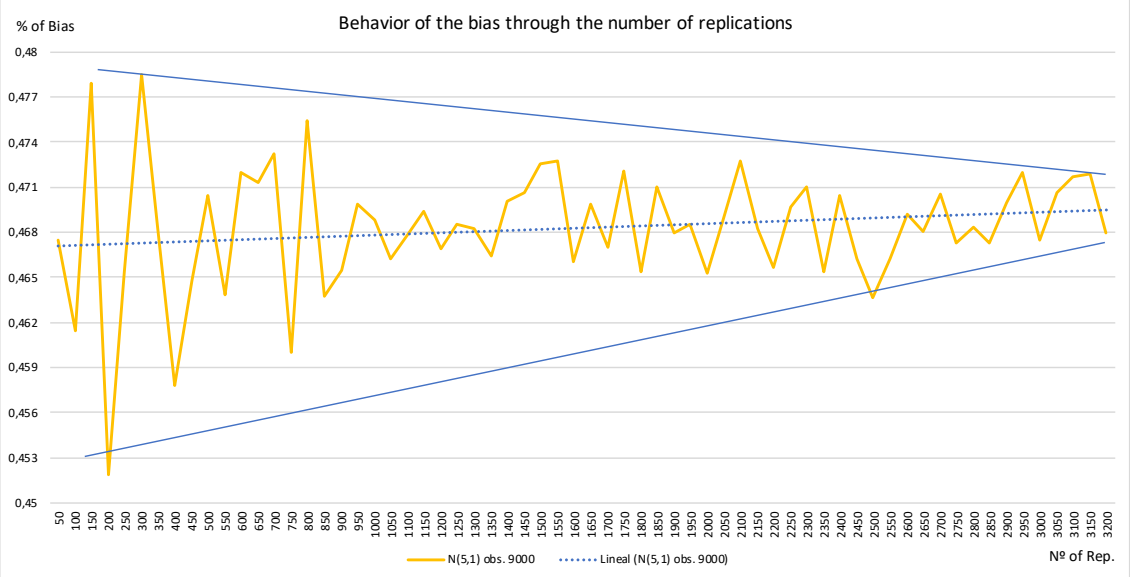
All the results were made by the author with simulated data. The table show the the confidence interval of the C-DID-T Square Bias and the values of that metric in the case of OLS and the algorithm. The first column (Obs.) is the number of observation of the dataset.

Table XII: Results Model N(5,1) - Repetitions 3000

		Variance	P value	Square Bias	Bias	MSE	MSE Winning	Bias Winning
Ordinary Least Squares								
Non Linear	1000	0,1026	0,0000	31,131	5,5796	31,234	No	No
	4000	0,0108	0,0000	8,3222	2,8848	8,3330	No	No
	9000	0,0034	0,0000	3,9198	1,9798	3,9232	No	No
Heterogeneous Treatment	1000	0,0247	0,0000	5,1815	2,2763	5,2061	No	No
	4000	0,0054	0,0000	1,3806	1,1750	1,3861	No	No
	9000	0,0023	0,0000	0,5803	0,7618	0,5826	No	No
Causal - DID - Tree							Variance Winning	P value Winning
Non Linear	1000	0,1473	0,0000	38,764	6,2261	38,911	No	No
	4000	0,4986	0,0000	14,008	3,7428	14,506	No	No
	9000	0,7220	0,0000	4,9878	2,2333	5,7098	No	No
Heterogeneous Treatment	1000	0,1473	0,0000	12,503	3,5361	12,651	No	No
	4000	0,4877	0,0000	6,3690	2,5237	6,8567	No	No
	9000	0,7220	0,0000	4,9878	2,2333	5,7098	No	No

Note: All the results were made by the author with simulated data. The left panel shows the results of the different metrics calculated by the previous methodologies. The right panel shows if the value of the algorithm is lower than the value of the OLS methodology. The first column (Obs.) is the number of observation of the dataset. The answer "OLS - NST" means that the OLS treatment parameter for that case was not statistically significant.

Graph III: Behavior of Bias for dataset N(5,1) Obs. 9000



All the results were made by the author with simulated data. The results were calculated for the datasets N(5,1) with 9000 observations. The first column (Obs.) is the number of observation of the dataset. The numbers of the graph were calculated as the results of the Bias of the Causal-DID-Tree divided by the bias of OLS minus one.

8 R and Stata Code Appendix

The following appendix was prepared based on the code used during the construction processes of this document. Probably, a user with more advanced knowledge in R, could have carried out all the processes under the same umbrella, without using stata.

8.1 Simulation Code

The Stata code that is presented below contains the process by which the different dataset that were used in the initial results were created. The code is quite simple, a loop that generates three sizes of dataset for each type of configuration. Comments on what is being done are with "*".

```
********

cd "/Volumes/Macintosh HD/Users/Juanjo/Desktop/Universidad/Groningen/Tesis/Simulacion"

*Simulations

*1.- N=1000-4000-9000, lineal model, lineal treatment,
* white noise effect on time trend
forval x=1(1)3 {
set seed 1
local i=1000*(`x')^2
set obs `i'
```

```
*Treatment and Time
gen treatment = ceil(runiform()*2)
replace treatment=0 if treatment==2
gen time=1

*Independent variables
gen x1 = rnormal(0, 1)
gen x2 = rnormal(0, 1)
gen x3 = rnormal(0, 1)
gen x4 = rnormal(0, 1)
gen r = rnormal(0,1)
gen r2= rnormal(0,1)

*Dependent variable pre and post time

gen y_t0 = 1 + (1/2)*x1 + (1/2)*x2 + (1/2)*x3 + (1/2)*x4 + (1/2)*treatment + r

mean y_t0
matrix b=e(V)
local p= b[1,1]

gen y_t1 = y_t0 + 5*(`p'^(1/2))*treatment + (time*r2 + 1)

gen treat_effect = 5*(`p'^(1/2))*treatment
save DataA`x'

clear all
}
*

*2.- N=1000-4000-9000, no-lineal model, No-lineal-hetero-treatment,
* white noise effect on time trend
forval x=1(1)3 {
set seed 1
local i=1000*(`x')^2
set obs `i'

*Treatment and Time
gen treatment = ceil(runiform()*2)
replace treatment=0 if treatment==2
gen time=1

*Independent variables
gen x1 = rnormal(0, 1)
gen x2 = rnormal(0, 1)
gen x3 = rnormal(0, 1)
gen x4 = rnormal(0, 1)
gen r = rnormal(0,1)
gen r2= rnormal(0,1)
```

```
gen r3= rnormal(0,2)

*Dependent variable pre and post time

gen y_t0 = 1 + (1/2)*x1^2 + (1/2)*x2 + (1/2)*x3^2 + x4 + treatment + r

mean y_t0
matrix b=e(V)
local p= b[1,1]

gen y_t1 = y_t0 + 5*(`p'^(1/2))*(treatment*r3)^2 + (time*r2 + 1)

gen treat_effect = 5*(`p'^(1/2))*(treatment*r3)^2

save DataB`x'

clear all
}
*

*3.- N=1000-4000-9000, no-lineal model, Lineal-hetero-treatment,
* white noise effect on time trend
forval x=1(1)3 {
set seed 1
local i=1000*(`x')^2
set obs `i'

*Treatment and Time
gen treatment = ceil(runiform()*2)
replace treatment=0 if treatment==2
gen time=1

*Independent variables
gen x1 = rnormal(0, 1)
gen x2 = rnormal(0, 1)
gen x3 = rnormal(0, 1)
gen x4 = rnormal(0, 1)
gen r = rnormal(0,1)
gen r2= rnormal(0,1)
gen r3= rnormal(0,2)

*Dependent variable pre and post time

gen y_t0 =  1 + (1/2)*x1^2 + (1/2)*x2 + (1/2)*x3^2 + x4 + treatment + r

mean y_t0
matrix b=e(V)
local p= b[1,1]

gen y_t1 = y_t0 + 5*(`p'^(1/2))*((1/2)*x1*treatment + (1/2)*r3*treatment) + (time*r2 + 1)
```

```
gen treat_effect = 5*(`p'^(1/2))*((1/2)*x1*treatment + (1/2)*r3*treatment)

save DataC`x'

clear all
}
*
```

8.2 Regression Tree Code

The code of the regression tree is based mainly on the R code of Athey and Imbens (2016) along with some modifications made by the author in order to reach the DID estimator.

```
#Usefull packages "foreign", "haven", "rpart", "rpart.plot", "causalTree", "iterators" #
# "doParallel" , "Cluster" #
Data <- read_dta("/Users/macbookpro/Dropbox/Simulacion/Data2B1.dta")

## Function creation:
DID = function() {
  n = nrow(Data)
  trIdx = which(Data$treatment == 1)
  conIdx = which(Data$treatment == 0)
  train_idx = c(sample(trIdx, length(trIdx) / 2),
                sample(conIdx, length(conIdx) / 2))
  train_data = Data[train_idx, ]
  est_data = Data[-train_idx, ]
  Ht <- data.frame(row(est_data[1]))
  results <- data.frame(row(est_data[1]))
  Htt <- data.frame(row(est_data[1]))
  SB <-data.frame(1)
  honestTree <- honest.causalTree(y_t0 ~ x1 + x2 + x3 + x4,
                                  data = train_data,
                                  treatment = train_data$treatment,
                                  est_data = est_data,
                                  est_treatment = est_data$treatment,
                                  split.Rule = "CT", split.Honest = T,
                                  HonestSampleSize = nrow(est_data),
                                  split.Bucket = T, cv.option = "fit",
                                  cv.Honest = F)
  opcp =  honestTree$cptable[,1][which.min(honestTree$cptable[,4])]
  opTree = prune(honestTree, opcp)
  Ht = data.frame(predict(opTree, newdata=est_data, type="vector"))
  honestTreet <- honest.causalTree(y_t1 ~ x1 + x2 + x3 + x4,
                                   data = train_data,
                                   treatment = train_data$treatment,
                                   est_data = est_data,
                                   est_treatment = est_data$treatment,
                                   split.Rule = "CT", split.Honest = T,
                                   HonestSampleSize = nrow(est_data),
```

```
                                    split.Bucket = T, cv.option = "fit",
                                    cv.Honest = F)
  opcp =  honestTreet$cptable[,1][which.min(honestTreet$cptable[,4])]
  opTree = prune(honestTreet, opcp)
  Htt = predict(opTree, newdata=est_data, type="vector")
  results = data.frame(Htt - Ht)
  SB = data.frame(mean((results - est_data$treat_effect)^2))
return(list(SB,results))
}


## Running the Algorithm
eachTrees = 10
iters = iter(rep(eachTrees, 3000))

cl = makeCluster(3)
registerDoParallel(cl)
result = foreach(ntree=iters, .combine=c, .multicombine=TRUE, .packages="causalTree") %dopar%
  DID()
stopCluster(cl)

## Extracting the data

r = seq(3, 6000, by=2)
SB <- data.frame(1)
for (i in r) {
  SB[(i-1)/2] = result[[ i ]]
}

trIdx = which(Data$treatment == 1)
conIdx = which(Data$treatment == 0)
train_idx = c(sample(trIdx, length(trIdx) / 2),
              sample(conIdx, length(conIdx) / 2))
train_data = Data[train_idx, ]
est_data = Data[-train_idx, ]

results <- data.frame(row(est_data[1]))


for (i in 1:3000) {
  results[i] = result[[ (i)*2 ]]
}


write.csv(SB, "/Users/macbookpro/Dropbox/Simulacion/SB4B1.csv")
write.csv(results, "/Users/macbookpro/Dropbox/Simulacion/results4B1.csv")
```

8.3 OLS Regression Code

This final section of the code is the one used to obtain the OLS results in Stata. Mainly it panelized the data that it had been previously created where the X_{ik} is kept constant by individual and only the Y_i change over time. The results are extracted and stored in a matrix.

```
*****

*Regression Model

*First we create a Panel from the database and then regress
matrix resultsA=J(3,3,0)
matrix resultsB=J(3,3,0)
matrix resultsC=J(3,3,0)

foreach z in A B C{
forval x=1(1)3{


use Data`z'`x'
gen id=_n
expand 2
gen id2=_n
egen aux1=count(id2)
gen time2=0 if id2<=aux1/2
replace time2=1 if time2==.

replace y_t0=y_t1 if time2==1

xtset id time2
gen did=treatment*time2
quietly reg y_t0 x1 x2 x3 x4 time2 treatment did
matrix b=e(b)
matrix c=e(V)
matrix d=r(table)
return list
matrix results`z'[1,`x']=b[1,7]
matrix results`z'[2,`x']=c[7,7]
matrix results`z'[3,`x']=d[4,7]
clear
}
}
*
svmat resultsA
svmat resultsB
svmat resultsC
```

References

- Athey, S., Bayati, M., Doudchenko, N., Imbens, G., & Khosravi, K. (2017). Matrix completion methods for causal panel data models. arXiv preprint arXiv:1710.10251.
- Athey, S., & Imbens, G. W. (2016). Recursive partitioning for heterogeneous causal effects. Proceedings of the National Academy of Sciences, 113(27), 7353-7360.
- Athey, S., & Imbens, G. W. (2017). The state of applied econometrics: Causality and policy evaluation. Journal of Economic Perspectives, 31(2), 3-32.
- Athey, S., Imbens, G. W., Pham, T., & Wager, S. (2017). Estimating average treatment effects: Supplementary analyses and remaining challenges. American Economic Review, 107(5), 278-81.
- Athey, S., Imbens, G. W., & Wager, S. (2016). Efficient inference of average treatment effects in high dimensions via approximate residual balancing (No. 3408).
- Athey, S., Tibshirani, J., & Wager, S. (2016). Generalized Random Forests. arXiv preprint arXiv:1610.01271.
- Ashenfelter, O., & Card, D. (1985). Using the Longitudinal Structure of earnings to estimate the effect of training programs. The Review of Economics and Statistics, 67(4), 648-660.
- Belloni, A., Chernozhukov, V., Fernández-Val, I., & Hansen, C. (2017). Program Evaluation and Causal Inference With High-Dimensional Data. Econometrica, 85(1), 233-298.
- Bertrand, M., Duflo, E., & Mullainathan, S. (2004). How much should we trust differences-in-differences estimates?. The Quarterly journal of economics, 119(1), 249-275.
- Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
- Beygelzimer, A., & Langford, J. (2009, June). The offset tree for learning with partial labels. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 129-138). ACM.
- Burlig, F., Knittel, C., Rapson, D., Reguant, M., & Wolfram, C. (2017). Machine learning from schools about energy efficiency (No. w23908). National Bureau of Economic Research.
- Cameron, A. C., & Trivedi, P. K. (2005). Microeconometrics: methods and applications. Cambridge university press.
- Card, D., & Krueger, A. (1994). Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania. The American Economic Review, 84(4), 772-793.
- Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., & Hansen, C. (2016). Double machine learning for treatment and causal parameters. arXiv preprint arXiv:1608.00060.
- Cicala, S. (2017). Imperfect markets versus imperfect regulation in US electricity generation (No. w23053). National Bureau of Economic Research.

- Graham, B. S., Pinto, C. C. D. X., & Egel, D. (2016). Efficient estimation of data combination models by the method of auxiliary-to-study tilting (AST). Journal of Business & Economic Statistics, 34(2), 288-301.
- Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society, 424-438.
- Hainmueller, J. (2012). Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. Political Analysis, 20(1), 25-46.
- Holland, P. (1986). Statistics and Causal Inference. Journal of the American Statistical Association, 81(396), 945-960.
- Imai, K., & Ratkovic, M. (2013). Estimating treatment effect heterogeneity in randomized program evaluation. The Annals of Applied Statistics, 7(1), 443-470.
- Imai, K., & Ratkovic, M. (2014). Covariate balancing propensity score. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 76(1), 243-263.
- Imbens, G. W., & Wooldridge, J. M. (2009). Recent developments in the econometrics of program evaluation. Journal of economic literature, 47(1), 5-86.
- Magerman, D. M. (1995, June). Statistical decision-tree models for parsing. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics (pp. 276-283). Association for Computational Linguistics.
- McCaffrey, D. F., Ridgeway, G., & Morral, A. R. (2004). Propensity score estimation with boosted regression for evaluating causal effects in observational studies. Psychological methods, 9(4), 403.
- Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of educational Psychology, 66(5), 688.
- Taddy, M., Gardner, M., Chen, L., & Draper, D. (2016). A nonparametric bayesian analysis of heterogenous treatment effects in digital experimentation. Journal of Business & Economic Statistics, 34(4), 661-672.
- Tian, L., Alizadeh, A. A., Gentles, A. J., & Tibshirani, R. (2014). A simple method for estimating interactions between a treatment and a large number of covariates. Journal of the American Statistical Association, 109(508), 1517-1532.
- Van Der Laan, M. J., & Rubin, D. (2006). Targeted maximum likelihood learning. The International Journal of Biostatistics, 2(1).
- Wager, S., & Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association, (just-accepted).
- Weisberg, H. I., & Pontes, V. P. (2015). Post hoc subgroups in clinical trials: Anathema or analytics?. Clinical trials, 12(4), 357-364.

- Wyss, R., Ellis, A. R., Brookhart, M. A., Girman, C. J., Jonsson Funk, M., LoCasale, R., & Sturmer, T. (2014). The role of prediction modeling in propensity score estimation: an evaluation of logistic regression, bCART, and the covariate-balancing propensity score. American journal of epidemiology, 180(6), 645-655.
- Zubizarreta, J. R. (2015). Stable weights that balance covariates for estimation with incomplete outcome data. Journal of the American Statistical Association, 110(511), 910-922.