Expert Systems with Applications 162 (2020) 113783

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Efficiency analysis trees: A new methodology for estimating production frontiers through decision trees

Miriam Esteve, Juan Aparicio *, Alejandro Rabasa, Jesus J. Rodriguez-Sala

Center of Operations Research (CIO). Miguel Hernandez University of Elche (UMH), 03202 Elche (Alicante), Spain

ARTICLE INFO

Article history:
Received 10 December 2019
Revised 19 June 2020
Accepted 19 July 2020
Available online 29 July 2020

Keywords:
Data envelopment analysis
Frontier analysis
Free disposal hull
Overfitting
Classification and Regression Trees

ABSTRACT

In this paper, we introduce a new methodology based on regression trees for estimating production frontiers satisfying fundamental postulates of microeconomics, such as free disposability. This new approach, baptized as Efficiency Analysis Trees (EAT), shares some similarities with the Free Disposal Hull (FDH) technique. However, and in contrast to FDH, EAT overcomes the problem of overfitting by using cross-validation to prune back the deep tree obtained in the first stage. Finally, the performance of EAT is measured via Monte Carlo simulations, showing that the new approach reduces the mean squared error associated with the estimation of the true frontier by between 13% and 70% in comparison with the standard FDH.

1. Introduction

Since the seminal works by Koopmans (1951), Debreu (1951), Shephard (1953) and Farrell (1957), a large amount of literature has been developed on how to estimate production frontiers and how to measure technical efficiency of production units. In particular, Farrell (1957) was the first in showing in practice, for a single output and multiple inputs, how to estimate an isoquant enveloping all the observations. Farrell based his significant contribution on the construction of a production possibility set that satisfied two standard axioms: convexity and free disposability. In this way, the most conservative estimation of the production possibility set may be obtained through the determination of the minimal set that envelops the observations and, at the same time, meets the two aforementioned axioms (Farrell, 1957). Overall, Farrell's principle of 'minimal extrapolation' leads to the estimation of a piecewise linear isoquant in the input space.

Farrell's approach can be categorized in the current area of nonparametric techniques, since it is not necessary to identify a priori the specific mathematical formulation of the production frontier to be estimated. This line of research was later taken up by Charnes et al. (1978) and Banker et al. (1984), resulting in the development of the Data Envelopment Analysis (DEA) approach, in which the determination of the frontier is only restricted via its axiomatic foundation. Another paper working in this line is that by Afriat (1972), showing how to determine a production function with the property P (e.g., non-decreasing concavity) that represents the set of observations that are as efficient as possible. Under the production of only one output, the estimated production functions suggested by Afriat coincide with those associated with what later would become known as DEA. Additionally, Deprins and Simar (1984) introduced the alternative technique known as Free Disposal Hull (FDH), which relies only on free disposability in contrast to DEA, which assumes free disposability and convexity. In fact, FDH may be considered the 'skeleton' of DEA since the convex hull of the frontier estimated by FDH coincides with the DEA frontier (Daraio and Simar, 2005).

Some existing nonparametric approaches for estimating production frontiers, like DEA and FDH, are based upon envelopment techniques (see, for example, Lozano and Calzada-Infante, 2017, Aparicio et al., 2017, Santin and Sicilia, 2017; Li et al., 2018). Their main objective is to analyze the efficiency of a set of observations, termed DMUs (Decision Making Units), which use several inputs to produce several outputs, by comparing their performance with respect to the boundary of a production possibility set, and using to that end a sample of observations operating in a similar technological environment (assumption of homogeneity). The usual methods for measuring technical efficiency of production need explicitly or implicitly to determine the boundary of the underlying technology, which constitutes the reference benchmark. Its estimation allows the calculation of the corresponding technical

* Corresponding author.
E-mail addresses: miriam.estevec@umh.es (M. Esteve), j.aparicio@umh.es (J. Aparicio), a.rabasa@umh.es (A. Rabasa), jesuja.rodriguez@umh.es (J.J. Rodriguez-Sala).

https://doi.org/10.1016/j.eswa.2020.113783
0957-4174/

inefficiency value for each DMU as the deviation of each activity or production plan to the frontier of the production possibility set.

Statistical inference is possible based on existing point estimates such as DEA and FDH but, by construction, they suffer from an overfitting problem. DEA and FDH underestimate the technical inefficiency of the observations[1], as they yield estimated frontiers always located below the (underlying) theoretical frontier. They are able to correctly describe the situation from an efficiency evaluation point of view but are not able to provide a suitable generalization. This scenario is very similar to the comparison between descriptive statistics and statistical inference. The idea behind generalization in statistical learning (Vapnik, 2000) is as follows. If the right balance is struck between the accuracy attained on some particular data (training set), and the ability of the approach to 'learn' any training set without error, then the best generalization performance will be achieved.

The general Operations Research community is moving into the data analytics field, where machine learning techniques are very usual. One of the problems that is dealt with this type of approaches is overfitting. For example, Classification and Regression Trees (CART) by Breiman et al. (1984) construct binary decision trees by repeatedly splitting nodes into two child nodes, beginning with the root node that contains the whole learning sample. However, a tree that is too deep yields estimates that are overly optimistic and a tree that is too short presents low accuracy attained on the training set. In this sense, Breiman et al. introduced a cost-complexity measure and a pruning procedure, based upon cross-validation, in order to overcome this problem.

In this paper, we adapt the technique of regression trees, based on CART, to estimate production frontiers satisfying the property of free disposability. Through the new approach, the input space is partitioned by a sequence of binary splits into terminal nodes. In each terminal node, the predicted response value (the output) is constant. So, graphically, the predictor looks like a step function and presents similarities and differences with respect to FDH when a deep tree is built. In order to overcome the usual overfitting problem, we will apply a pruning procedure based upon cross-validation. Among other advantages, it will allow us to determine efficiency evaluation out-of-sample for the assessed DMUs. Additionally, and although this type of nonparametric technique is not able to provide information on the statistical significativity of each predictor (input variable), it is able to determine a ranking of importance of each of them. Finally, and from a data visualization point of view, the new approach allows the graphical representation of multivariate situations that would otherwise be difficult to draw through simple trees.

As we are aware, this paper represents the first contribution that really connects two key topics: machine learning and frontier analysis. Previous contributions that considered these two types of worlds really do not mix them but apply each one in a different stage. For example, Emrouznejad and Anouze (2010) combined DEA and CART to perform an efficiency analysis of the banking sector. To do that, in a first stage, DEA determines the efficiency scores, and in a second stage, efficiency is used as a response variable with CART to give details of factors related to technical inefficiency in the banking sector. Another recent example is Rebai et al (2019), where, for identifying the key factors that impact Tunisian schools' academic performance, the authors carried out a two-stage analysis. In the first stage, they use the Directional Distance Function approach and DEA to deal with undesirable outputs. In the second stage, they apply machine-learning approaches (regression trees and random forests) to identify and visualize variables that are associated with a school performing very well.

The paper is organized as follows. Section 2 is devoted to briefly introduced the FDH and the standard CART techniques. In Section 3, we extend regression trees to the context of estimating production frontiers, developing the new technique called Efficiency Analysis Trees (EAT). Performance of EAT is investigated via Monte Carlo simulations in Section 4. Finally, Section 5 concludes.

2. Background

In this section, we briefly review the main notions related to Free Disposal Hull and Classification and Regression Trees. Additionally, we will need to introduce some notation.

2.1. Free Disposal Hull (FDH)

Let us consider n Decision Making Units (DMUs) to be evaluated. DMU$_i$ consumes $\boldsymbol{x}_i = (x_{1i}, ..., x_{mi}) \in R_+^m$ amounts of inputs for the production of $\boldsymbol{y}_i = (y_{1i}, ..., y_{si}) \in R_+^s$ amounts of outputs[2]. The relative efficiency of each DMU in the sample is assessed with reference to the so-called production possibility set or technology, which is the set of technically feasible combinations of $(\boldsymbol{x}, \boldsymbol{y})$. It is defined in general terms as:

$$\Psi = \left\{(\boldsymbol{x}, \boldsymbol{y}) \in R_+^{m+s} : \boldsymbol{x} \text{ can produce } \boldsymbol{y}\right\} \tag{1}$$

Certain assumptions are done on this set, such as monotonicity (free disposability) of inputs and outputs, meaning that if $(\boldsymbol{x}, \boldsymbol{y}) \in \Psi$, then $(\boldsymbol{x}', \boldsymbol{y}') \in \Psi$, as soon as $\boldsymbol{x}' \geqslant \boldsymbol{x}$ and $y' \leq y$[3]. Often convexity of Ψ is also assumed (see, e.g., Färe and Primont, 1995).

As far as the measurement of technical efficiency was the concern, a certain part of the boundary of Ψ is of interest. Specifically, the efficient frontier of Ψ may be defined as $\partial(\Psi) := \{(\boldsymbol{x}, \boldsymbol{y}) \in \Psi : \hat{\boldsymbol{x}} < \boldsymbol{x}, \hat{\boldsymbol{y}} > \boldsymbol{y} \Rightarrow (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \notin \Psi\}$. Technical efficiency is defined as the distance from a point that belongs to Ψ to the production frontier $\partial(\Psi)$. For a point located inside Ψ, it is evident that there are many possible paths to the frontier, each associated with a different technical efficiency measure. One of the most used technical efficiency measures in the literature is that known as the output-oriented radial model (see Charnes et al., 1978, Banker et al., 1984), which is the inverse of the classical Shephard output distance function (Shephard, 1953). This measure is especially useful in contexts where inputs are predetermined (such as land and labor on a farm) and generating maximum output is the appropriate objective from a technical perspective. In particular, the output-oriented radial model determines the efficiency score for an evaluated point $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ by equiproportionally increasing all its outputs while maintaining inputs constant:

$$\phi(\boldsymbol{x}_k, \boldsymbol{y}_k) = \max\{\phi_k \in R : (\boldsymbol{x}_k, \phi_k \boldsymbol{y}_k) \in \Psi\} \tag{2}$$

Nowadays, there are different nonparametric methods in the literature for estimating the efficient frontier of Ψ. The most noteworthy are those that follow. The FDH estimator was introduced by Deprins and Simar (1984) and relies only on the free disposability assumption. In contrast, the DEA estimator requires stronger assumptions, such as convexity of the set Ψ. The convexity assumption is widely used in economics, but it is not always valid. The production possibility set might admit increasing returns to scale (i.e. outputs increase at a faster rate than inputs, which which cannot be graphically modeled by convexity), or there might be lumpy goods (i.e. fractional values of inputs or outputs do not

[1] This is only true when the underlying Data Generating Process does not contain measurement error and/or other statistical noise. Therefore, the bias referred here is due to an insufficient sample size.

[2] We use bold for denoting vectors, and non-bold for scalars.

[3] Let $\boldsymbol{z} = (z_1, ..., z_q)$ and $\boldsymbol{z}' = (z_1', ..., z_q')$. Hereinafter, $\boldsymbol{z} \leqslant \boldsymbol{z}'$ means $z_j \leqslant z_j'$ for all $j = 1, ..., q$, and $\boldsymbol{z} < \boldsymbol{z}'$ means $z_j < z_j'$ for all $j = 1, ..., q$.

exist). Hence, the FDH can yield a more general and flexible estimator than DEA (see Aragon et al., 2005). Other alternative nonparametric techniques for estimating production frontiers are those that apply Kernel based approaches and local regression techniques. See, for example, Du et al. (2013), where the authors propose a kernel smoothing method that can handle multiple shape constraints (e.g., monotonicity) for multivariate functions, generalizing Hall and Huang (2001). Another interesting contribution is Parmeter et al. (2014), who showed how constraint weighted bootstrapping may be applied to impose smoothness conditions on linear estimates. In particular, these authors estimated an input distance function both parametrically and nonparametrically, resorting in the last case to local linear generalized kernel regression. See also Henderson and Parmeter (2009). The approach that we develop in this paper is in line with all these contributions.

Regarding the FDH technique, this approach has once more attracted the interest of researchers in recent years. For example, Tavakoli and Mostafaee (2019) adapted Network Data Envelopment Analysis models to Free Disposal Hull models, providing the target intermediate products for inefficient observations. Barbosa et al. (2019) developed a Hybrid Evolutionary Genetic Algorithm and Scatter Search for the discrete and dynamic Berth Allocation Problem, where DEA and FHD are exploited to compare the performance of alternative specifications of the parameters for the algorithm. Kerstens et al. (2019) reconsidered the way metafrontiers are obtained from nonparametric estimates of underlying group-specific frontiers, concluding that the usual convexification strategy consisting in assuming a convex metaset generally leads to erroneous results. Or, from a theoretical point of view, the papers by Cazals et al. (2002) and Daraio and Simar (2005) introduced an FDH-based estimator, robust to extreme values and the concept of conditional efficiency measure of order-*m*, respectively.

The nonparametric models, such as FDH, are particularly appealing since they do not rely on restrictive hypothesis on the data-generating process, a feature shared with usual machine learning techniques, which are clearly data-driven approaches. In particular, Deprins and Simar (1984) proposed the Free Disposal Hull of the set of observations (DMUs) to estimate Ψ, defined as follows.

$$\hat{\Psi}_{FDH} = \left\{(\boldsymbol{x},\boldsymbol{y}) \in R_+^{m+s} : \exists i = 1,...,n \text{ such that } \boldsymbol{y} \leqslant \boldsymbol{y}_i, \boldsymbol{x} \geqslant \boldsymbol{x}_i\right\} \quad (3)$$

In the univariate output case, the production function would be estimated by $\hat{f}_{FDH}(\boldsymbol{x}) = \max_{i:\boldsymbol{x}\geqslant\boldsymbol{x}_i}\{y_i\}$. Next, we present a graphical example of the FDH estimator, showing its typical non-decreasing step shape.

The FDH technique is very engaging because it relies on very few assumptions, but, by construction, it suffers overfitting. Note, in Fig. 1, that the estimated frontier by FDH fits like a glove to the data sample, satisfying additionally free disposability. This problem is shared by other well-known data-driven approaches. In particular, techniques that belong to the machine learning field like, for example, Classification and Regression Trees (CART), present problems of overfitting when a deep tree is generated. This problem can be solved by pruning back the deep tree through, for example, a cross-validation process.

Finally, in the case of the FDH technique, the efficiency score $\phi(\boldsymbol{x}_k,\boldsymbol{y}_k)$ is estimated by plugging $\hat{\Psi}_{FDH}$ into (2) in place of Ψ. This substitution allows determining the estimation by a mixed-integer linear optimization program:

$$\begin{aligned} \phi^{FDH}(\boldsymbol{x}_k,\boldsymbol{y}_k) = \quad & \max \quad && \phi \\ & s.t. \\ & && \textstyle\sum_{i=1}^n \lambda_i x_{ji} \leqslant x_{jk}, && j = 1,...,m \\ & && \textstyle\sum_{i=1}^n \lambda_i y_{ri} \geqslant \phi y_{rk}, && r = 1,...,s \\ & && \textstyle\sum_{i=1}^n \lambda_i = 1, \\ & && \lambda_i \in \{0,1\}, && i = 1,...,n \end{aligned} \quad (4)$$

$\phi^{FDH}(\boldsymbol{x}_k,\boldsymbol{y}_k)$ is always greater than one, with one indicating technical efficiency.

2.2. Classification and Regression Trees (CART)

CART (Breiman et al., 1984) is a machine learning technique with two objectives, depending on the nature of the response variable. CART works like a nonparametric classification model when the response variable is categorical and as a regression model when it is numerical. In this paper, we focus on the latter approach, due to the nature of our response variable.

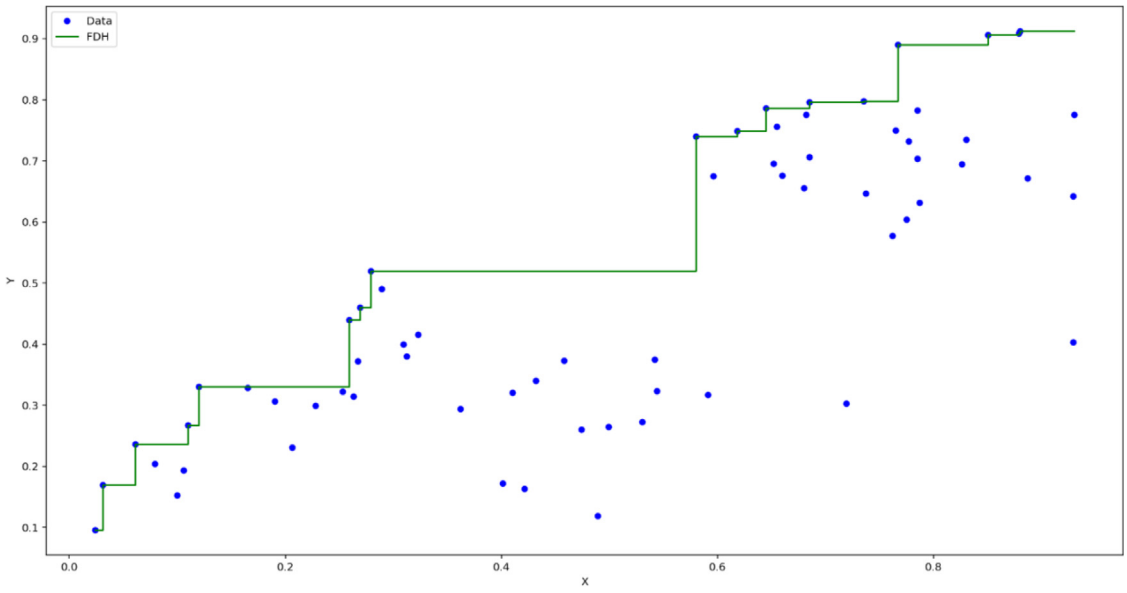

Fig. 1. Example of FDH estimate.

The principle behind CART is relatively simple: A given criterion is chosen to recursively generate binary partitions of the data (the training sample) until no further meaningful division is possible or a stopping rule holds. The graphical result of this process is a tree that begins at the root node, develops through intermediate nodes, and ends at the terminal nodes (leaves). The binary nature of CART is reflected in each (parent) node, except the leaves, giving birth to two child nodes.

The split at any non-terminal node is caused by a predictor variable and a threshold for this variable. Given all the possible ways to split the data at a (parent) node (i.e., the combinations of each predictor and each threshold), CART builds regression trees by choosing the split that minimizes the sum of the mean squared error (MSE) of the two child nodes. More precisely, given a learning sample Ω consisting of $(\boldsymbol{z}_1, y_1), \ldots, (\boldsymbol{z}_n, y_n)$, with $\boldsymbol{z}_i \in R^m$ and $y_i \in R$, $i = 1, \ldots, n$, CART has as an objective to predict the response variable y through the predictors $z_1, \ldots, z_m$. To do that, the first split by CART selects a predictor variable j, $j = 1, \ldots, m$, and a threshold $s_j \in S_j$, where S_j is the set of possible thresholds for variable j, such that the sum of the MSE calculated for the data belonging to the left child node, i.e. the data that satisfies the condition $z_j < s_j$, and the MSE determined with the data belonging to the right child node, i.e. the data that meets $z_j \geqslant s_j$, is minimized. If t is the way of denoting the parent node and, furthermore, t_L and t_R are the left and right child nodes, respectively, then, CART selects the best combination (z_j, s_j) by minimizing

$$R(t_L) + R(t_R) = \frac{1}{n} \sum_{(\boldsymbol{z}_i, y_i) \in t_L} (y_i - y(t_L))^2 + \frac{1}{n} \sum_{(\boldsymbol{z}_i, y_i) \in t_R} (y_i - y(t_R))^2 \qquad (5)$$

where n is the sample size and $y(t_L)$ and $y(t_R)$ are the estimates of response variable y for the data in nodes t_L and t_R, respectively. Usually, $y(t_L) = \frac{1}{n(t_L)} \sum_{(\boldsymbol{z}_i, y_i) \in t_L} y_i$ and $y(t_R) = \frac{1}{n(t_R)} \sum_{(\boldsymbol{z}_i, y_i) \in t_R} y_i$, where $n(t)$ is the sample size in node t. In words, $y(t_L)$ and $y(t_R)$ are the means of the data contained in each child node. In this way, the tree would have one root node and two additional nodes. Once CART generates a split, the entire process is repeated for the resulting child nodes. The process continues until no further meaningful segmentations are possible or a predefined stopping rule is satisfied (the usual one is that $n(t) \leqslant 5 = n_{\min}$ for all leaf nodes).

Even under a multivariate situation as that treated by CART, this technique admits a graphical representation of the final predictor by a simple tree such as the following.

Additionally, in low dimensions, it is possible to draw the predictor through a step function like that shown in Fig. 3. The predictor is constant over each terminal node. Note also that the predictor determined by CART does not envelop the data but deals with the average of the response variable and, additionally, the predictor does not satisfy the property of free disposability. These facts clearly contrast with the features of the FDH predictor (see Fig. 1). Obviously, CART was not designed by Breiman et al. (1984) for dealing with the estimation of production frontiers in microeconomics. However, note that both approaches, FDH and CART, generate step functions as predictors.

It is worth mentioning that CART suffers overfitting. The tree grown is usually too large and the yielded estimates are overly optimistic. Even in the case of using a less strict constraint as a stopping rule, with a threshold for the sample size in each leaf node bigger than five, one could argue that this pre-fixed level is arbitrary. Additionally, in large trees ($n_{\min}$ small), the accuracy of the approach is good with respect to the training sample since the step function associated with the predictor is very close to the observed data. Unfortunately, the predictor is too dependent on the sample, which makes it difficult to generalize the results and provide a suitable estimate. This was a limitation clearly recognized by Breiman et al. in their famous book. To overcome this problem without previously assuming any distribution on the statistical noise, Breiman et al. proposed to prune the tree in an appropriate way. To do that, they defined the error-complexity measure $R_\alpha(T)$, which is based on two indicators. The first one is a measure of accuracy of the tree, defined as the aggregation of the mean squared error determined in each leaf node, while the second one is the number of leaf nodes as a measure of the size or complexity of the tree. At the same time, a third element comes into

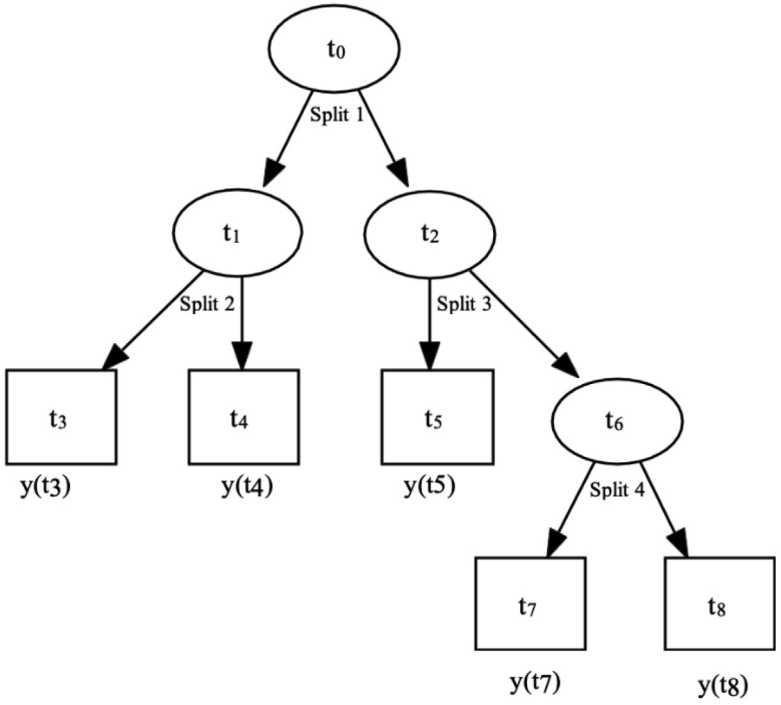

Fig. 2. Example of tree.

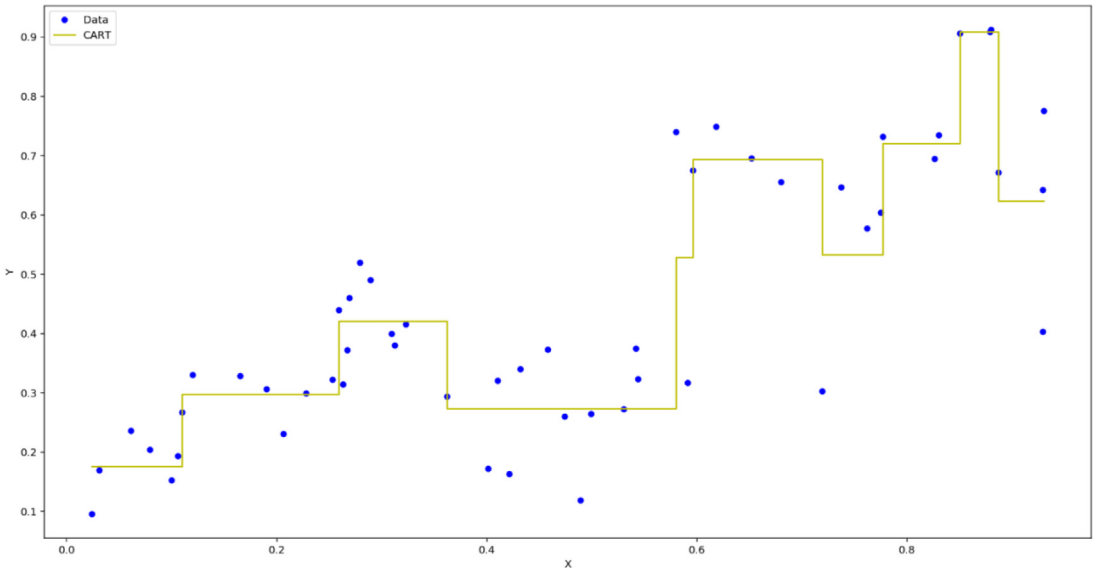

Fig. 3. Example of CART estimate.

play: a parameter α, which works as a weight for balancing the two previous indicators:

$$R_\alpha(T) = R(T) + \alpha\left|\tilde{T}\right| \tag{6}$$

where $\tilde{T}$ is the set of leaf nodes of the tree T, $R(T) = \frac{1}{n}\sum_{t\in\tilde{T}}\sum_{(\boldsymbol{z}_i,y_i)\in t}(y_i - y(t))^2$ and $|S|$ denotes the cardinality of set S.

Now, the idea is pruning by minimizing the error-complexity measure. The result will be a size decreasing sequence of subtrees in the way: $T_{\max} \succ T_1 \succ T_2 \succ \ldots \succ \{t_0\}$, where $T_{\max}$ is the tree obtained by using the stopping rule $n(t) \leqslant n_{\min}$ and t_0 is the root node. Additionally, an increasing sequence of α values associated with each subtree has to be determined in the way $0 = \alpha_1 < \alpha_2 < \ldots$ and satisfying that for α such that $\alpha_k \leqslant \alpha < \alpha_{k+1}$, T_k is the smallest subtree of $T_{\max}$ minimizing $R_\alpha(T)$.

For scoring each subtree in the sequence and selecting the best, it is usual to resort to cross-validation. In V-fold cross-validation, the learning sample Ω is randomly divided into $\Omega_1, \ldots, \Omega_V$ disjoint subsamples with the same sample size or as close as possible. Let the v-th learning subsample be $\Omega^{(v)} = \Omega - \Omega_v$. Then, the process to get a score for each subtree is based on repeating the tree growing and pruning procedure commented above using $\Omega^{(v)}$ instead of Ω. For each v, it yields the trees $T^{(v)}(\alpha)$ which are the minimal error-complexity trees for the parameter α. Define $\alpha'_k = \sqrt{\alpha_k\alpha_{k+1}}$. Denote by $d_k^{(v)}(\boldsymbol{z})$ the predictor corresponding to the tree $T^{(v)}(\alpha'_k)$. Then, the score associated with subtree T_k in the original sequence $T_{\max} \succ T_1 \succ T_2 \succ \ldots \succ \{t_0\}$ is determined as follows.

$$R^{cv}(T_k) = \frac{1}{n}\sum_{v=1}^{V}\sum_{(\boldsymbol{z}_i,y_i)\in\Omega_v}\left(y_i - d_k^{(v)}(\boldsymbol{z}_i)\right)^2 \tag{7}$$

Finally, the subtree selected T_k is the smallest tree satisfying the condition:

$$R^{cv}(T_k) \leqslant R^{cv}\left(T_{k_0}\right) + SE \tag{8}$$

where T_{k_0} is such that $R^{cv}\left(T_{k_0}\right) = \min_k\left\{R^{cv}(T_k)\right\}$ and SE is the standard error estimate for $R^{cv}\left(T_{k_0}\right)$.

In Fig. 4, we show how is the shape of the final tree determined by CART after the pruning process was implemented.

In contrast to traditional parametric regression models, CART for regression does not provide information on what the relationship between each variable $z_j, j = 1, \ldots, m$, and the response variable is like. While parametric models can provide (point and interval) estimations of the coefficients associated with that relationship, CART does not. Nevertheless, Breiman et al. (1984) also suggested how to at least provide a ranking of variables z_j depending on the importance of each variable. A priori, it is not trivial how to rank these variables since by not giving the best split at a node, it could be the second or third best option. For example, in the final selected tree, $z_{j'}$ could never occur in any split. However, if a certain alternative variable $z_{j''}$ is removed from the analysis and another tree is grown, then $z_{j'}$ could appear prominently in the splits, even with a resulting tree as accurate as the original one. In such a context, we need that the variable ranking method detects the importance of $z_{j'}$ in the original tree grown. With such aim, Breiman et al. (1984) proposed a procedure to determine a suitable ranking, playing with the notion of surrogate splits.

3. Efficiency Analysis Trees (EAT)

In this section, we introduce a new technique based on the adaptation of CART for the estimation of production frontiers, which will be denominated Efficiency Analysis Trees (EAT). The new approach will allow the determination of production frontiers, satisfying the usual axioms of microeconomics, through a data-driven approach that does not assume any particular distribution on the data noise and generates a step function as a predictor. These characteristics are shared with the FDH technique. However, while FDH suffers overfitting, the new method will try to overcome this problem through cross-validation and pruning, as in Breiman et al. (1984), exploiting its natural structure as a tree. Another remarkable difference between these two techniques, EAT and FDH, is that the growing of the tree, in the case of EAT, is carried out in a statistical way, minimizing the mean squared error and using stopping rules linked to the sample size, and avoiding yielding empty terminal nodes. An empty node implies a region in the

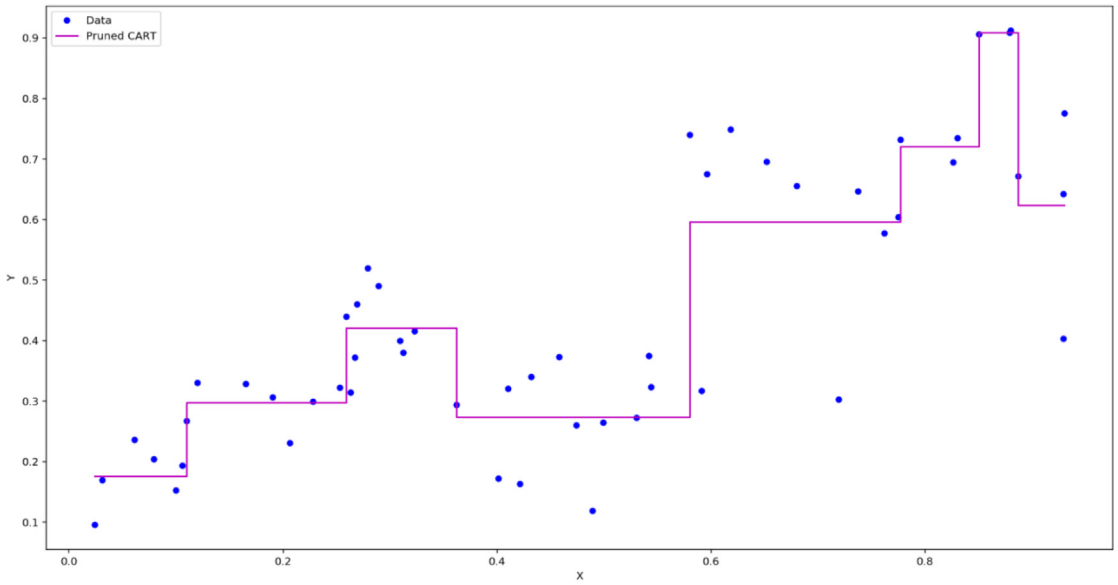

Fig. 4. Example of CART estimate after the pruning process.

input space without data but with an estimation of the response variable (the output). FDH can produce this type of regions, as we will show later, something odd from a data-driven perspective since it is not possible to calculate a measure of error at that particular node.

3.1. The single-output case

In conventional Supervised Statistical Learning, all start with a learning sample Ω, consisting of examples $(\boldsymbol{z}_1, y_1), ..., (\boldsymbol{z}_n, y_n)$, with $\boldsymbol{z}_i \in R^m$ and $y_i \in R$, $i = 1, ..., n$, as in CART. Nevertheless, in our production context, the variables $z_1, ..., z_m$ are interpreted as inputs. So, let us rename $z_1, ..., z_m$ as $x_1, ..., x_m$ for utilizing a more usual notation, where $x_j \in R_+$, $j = 1, ..., m$. Now, mirroring Breiman et al. (1984), three key elements are necessary to determine a tree predictor:

1. A rule for assigning an estimation of the response variable to every node: $y(t)$;
2. A way to select a split at every intermediate node;
3. A rule for determining when a node is terminal.

Furthermore, we need to add two additional elements in our production framework.

4. $y(t)$ should estimate the frontier instead of the mean of the response variable. This feature can be easily linked to point 1 above.
5. The satisfaction of free disposability. This point is probably the most difficult part to be addressed and we deal with it in this paper by proposing a way of selecting the next node that must be split in the growing algorithm and by suggesting how to modify the estimation of the output in each node.

We start with points 1 and 4, proposing a suitable enveloped estimate for the response variable at node t: $y(t)$. In this sense, the value $y(t)$ that minimizes $R(t) = \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}(y_i - y(t))^2$ at node t, i.e. the within node sum of squares, and, at the same time, satisfies $y(t) \geqslant y_i$, $\forall(\boldsymbol{x}_i, y_i) \in t$, is the maximum value of the response variable observed in the data sample belonging to node t. This statement is proven in Proposition 1.

Proposition 1. *The value of $y(t)$ such that minimizes $R(t) = \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}(y_i - y(t))^2$ subject to $y(t) \geqslant y_i$, $\forall(\boldsymbol{x}_i, y_i) \in t$ is $y(t) = \max_{(\boldsymbol{x}_i, y_i)\in t}\{y_i\}$.*

Proof. Let us assume that $y(t) = \max_{(\boldsymbol{x}_i, y_i)\in t}\{y_i\} + \delta$, with $\delta > 0$. Then, $\frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}(y_i - y(t))^2 = \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}(y_i - (\max_{(\boldsymbol{x}_i, y_i)\in t}\{y_i\} + \delta))^2 = \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}\left(\left(y_i - \max_{(\boldsymbol{x}} i, y_i) \in t\{y_i\}\right) - \delta\right)^2 = \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}\left(y_i - \max_{(\boldsymbol{x}_i, y_i)\in t}\{y_i\}\right)^2 + \underbrace{\delta^2\frac{n(t)}{n} - 2\delta\frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}\left(y_i - \max_{(\boldsymbol{x}_i, y_i)\in t}\{y_i\}\right)}_{=A}$. Note that $A \geqslant 0$ since $y_i \leqslant \max_{(\boldsymbol{x}_k, y_k)\in t}\{y_k\}$, $\forall(\boldsymbol{x}_i, y_i) \in t$. Therefore, $\frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}(y_i - \left(\max_{(\boldsymbol{x}_i, y_i)\in t}\{y_i\} + \delta\right))^2 \geqslant \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i)\in t}\left(y_i - \max_{(\boldsymbol{x}_i, y_i)\in t}\{y_i\}\right)^2$, $\forall\delta > 0$, which proves the proposition.

Regarding point 2, let us assume that we have a node to be split. Also, let us remember that, in this situation, standard CART selects a predictor variable j, $j = 1, ..., m$, and a threshold $s_j \in S_j$, where S_j is the set of possible thresholds for variable j, such that the sum of the MSE calculated for the data belonging to the left child node, i.e. the data that satisfies the condition $x_j < s_j$, and the MSE determined with the data that belongs to the right child node, i.e. the data that meets $x_j \geqslant s_j$, is minimized. If t is the way of denoting the parent node, and t_L and t_R are the left and right child nodes, respectively, then, CART selects the best combination (z_j, s_j) by minimizing expression (5). We will follow the same criteria when splitting an intermediate node with EAT.

Additionally, a node is terminal in EAT (point 3) when it satisfies a certain stopping rule, like $n(t) \leqslant n_{\min}$. In this sense, Breiman et al. (1984) recommended $n_{\min} = 5$. Another natural stopping rule is that associated with a situation where all observations at a node share the same values for all the inputs (share the same input profile). In this case, it is not possible to split the node. We will use these thresholds in our computational experiences. Nevertheless, seeking simplicity in the development of the methodological part, we will assume in this section that not all observations at a node share the same input profile.

As we previously mentioned, probably the most difficult task is that related to defining an algorithm that iteratively splits nodes by minimizing (5) and, at the same time, guarantees the satisfaction

of the property of free disposability. In our approach, the key is the algorithm for splitting each intermediate node. Before showing the algorithm, it is necessary to introduce some definitions and additional notation. In this respect, let $k = 1, ..., K$ be the number of executed splits in the growing process of a tree. Let T_k be the created tree after the k-th split. Let $\widetilde{T}_k$ be the set of terminal nodes in tree T_k. For example, T_1 for the graphical example in Fig. 2 is the (sub) tree composed by $\{t_0, t_1, t_2\}$ with terminal nodes t_1 and t_2. For this same example, T_2 could be the tree $\{t_0, t_1, t_2, t_3, t_4\}$ with set of terminal nodes $\widetilde{T}_2 = \{t_2, t_3, t_4\}$.

In the tree structure, each node t is defined by the fulfillment of a series of conditions in the input space of the type $\{x_j < s_j\}$ or $\{x_j \geqslant s_j\}$. Conditions that are associated with each split. Therefore, after executing a certain number of splits, there is a region in the input space in the shape of a segment if $m = 1$, a rectangle if $m = 2$, a parallelepiped if $m = 3$, etc. This is the support of node t.

Definition 1. *Let $k = 1, ..., K$. Let $t \in \widetilde{T}_k$. Then, the support of node t is denoted and defined as* $\text{supp}(t) = \left\{\boldsymbol{x} \in R_+^m : a_j^t \leqslant x_j < b_j^t, j = 1, ..., m\right\}$, *with* $a_j^t < b_j^t, \forall j = 1, ..., m$.

The parameter a_j^t could be zero, whereas the parameter b_j^t could be $+\infty$. In practice, a_j^t is lower bounded by the minimum observed in variable x_j in the learning sample, i.e., $a_j^t = \min_{1 \leqslant i \leqslant n} \{x_{ji}\}$. Note also that if $(\boldsymbol{x}_i, y_i) \in t$, then $\boldsymbol{x}_i \in \text{supp}(t)$. The opposite is also true since if $\boldsymbol{x}_i \in \text{supp}(t)$, then $(\boldsymbol{x}_i, y_i) \in t$ because the splits yield disjoint subsets of the original learning sample.

It is also necessary to define the notion of Pareto-dominant nodes of a given node $t \in \widetilde{T}_k$. This definition will be related to the satisfaction of the property of free disposability of the estimated production frontier through EAT.

Definition 2. *Let $k = 1, ..., K$. Let $t \in \widetilde{T}_k$. Then, the set of Pareto-dominant nodes of node t is denoted and defined as* $I_{T_k}(t) = \left\{t' \in \widetilde{T}_k - t : \exists \boldsymbol{x} \in \text{supp}(t), \exists \boldsymbol{x}' \in \text{supp}(t') \text{ such that } \boldsymbol{x}' \leqslant \boldsymbol{x}\right\}$.

An element in $I_{T_k}(t)$ is a node with at least an input vector in its corresponding support, non-necessarily observed in the learning sample, such that it dominates at least an input vector belonging to the support of node t in the Pareto sense. This notion will be exploited for the satisfaction of the property of free disposability.

Checking if a node $t' \in \widetilde{T}_k$ belongs to $I_{T_k}(t)$ for $t \in \widetilde{T}_k$ is an easy task since it is enough to compare the m components of two vectors, $\boldsymbol{a}^{t'}$ and $\boldsymbol{b}^t$, as shown in the next proposition.

Proposition 2. *$t' \in I_{T_k}(t)$ if and only if $\boldsymbol{a}^{t'} < \boldsymbol{b}^t$.*

Proof. Let us assume that $t' \in I_{T_k}(t)$. Then, by Definition 2, $\exists \boldsymbol{x} \in \text{supp}(t)$, $\exists \boldsymbol{x}' \in \text{supp}(t')$ such that $\boldsymbol{x}' \leqslant \boldsymbol{x}$. By Definition 1, $\boldsymbol{a}^{t'} \leqslant \boldsymbol{x}'$ and $\boldsymbol{x} < \boldsymbol{b}^t$. Therefore, $\boldsymbol{a}^{t'} < \boldsymbol{b}^t$. Let now assume that $\boldsymbol{a}^{t'} < \boldsymbol{b}^t$. Observe also that $\boldsymbol{a}^{t'} \in \text{supp}(t')$ by Definition 1. This is not the case for $\boldsymbol{b}^t$ with respect to $\text{supp}(t)$. We have to search for a point $\widetilde{\boldsymbol{x}}$ in $\text{supp}(t)$ such that $\boldsymbol{a}^{t'} \leqslant \widetilde{\boldsymbol{x}}$. We can approach $\boldsymbol{b}^t$, from the interior of the $\text{supp}(t)$, following the path associated with considering points defined as the convex combination of $\boldsymbol{a}^{t'}$ and $\boldsymbol{b}^t$: $\lambda \boldsymbol{a}^{t'} + (1 - \lambda)\boldsymbol{b}^t$, $\lambda \in (0, 1)$. To do that, it is enough to take lambda values close to zero. Note that $\boldsymbol{a}^{t'} \leqslant \lambda \boldsymbol{a}^{t'} + (1 - \lambda)\boldsymbol{b}^t$. Therefore, $\exists \widetilde{\lambda} \in (0, 1)$ such that $\boldsymbol{a}^{t'} \leqslant \widetilde{\boldsymbol{x}}$, with $\widetilde{\boldsymbol{x}} = \lambda \boldsymbol{a}^{t'} + (1 - \lambda)\boldsymbol{b}^t$ and $\widetilde{\boldsymbol{x}} \in \text{supp}(t)$. Consequently, $t' \in I_{T_k}(t)$.

In Fig. 5, we show an illustration of Proposition 2. Two supports associated with node t and t' appear. Clearly, the hypothesis of the proposition holds, i.e., $\boldsymbol{a}^{t'} < \boldsymbol{b}^t$. Consequently, node t' is a Pareto-dominant node of t.

Additionally, and regarding notation, let $t^* \in \widetilde{T}_k$ such that this node does not satisfy any stopping rule and, therefore, it is possible to split it into two child nodes, t_L and t_R. Then, the tree associated with this specific split will be denoted as $T(k|t^* \to t_L, t_R)$. Note that, in $T(k|t^* \to t_L, t_R)$, $t^* \notin \widetilde{T}(k|t^* \to t_L, t_R)$. Indeed, $\widetilde{T}(k|t^* \to t_L, t_R) = \left(\widetilde{T}_k - t^*\right) \cup \{t_L, t_R\}$.

Next, we show how to implement the process for splitting under our approach given a node $t^* \in \widetilde{T}_k$ that does not satisfy any stopping rule. First of all, for all possible combinations of (splitting) variables x_j and threshold $s_j \in S_j$, i.e. (x_j, s_j), the algorithm must determine the sum of the mean squared errors associated with splitting the node t^* into a left node and a right node with their corresponding estimations of the response variable y. After that, the selection of the best possible combination $\left(x_{j^*}^*, s_{j^*}^*\right)$ is available: the one that minimizes the sum of errors. In order to estimate the response variable y in each child node, given a particular combination (x_j, s_j), we suggest combining Proposition 1 and the concept of Pareto-dominant nodes. As we will show later, this point will be the key for guarantying free disposability. Let us start the description of our algorithm with the evaluation of the left child node, t_L, given the combination (x_j, s_j). First, the algorithm must determine the set of Pareto-dominant nodes of t_L, i.e. $I_{T(k|t \to t_L, t_R)}(t_L)$. Once the elements of this set are known, we are ready to calculate the estimation of the response variable y (the output) at node t_L. In our approach, we force the estimate to never be smaller than the estimations of the response variable at nodes belonging to the set of Pareto-dominant nodes of t_L. This is the trick for satisfying free disposability in a constructive way. So, we suggest the following estimation of the response variable for the child node t_L.

$$y(t_L) = \begin{cases} \max\{y_i : (\boldsymbol{x}_i, y_i) \in t_L\}, & \text{if } \max\{y_i : (\boldsymbol{x}_i, y_i) \in t_L\} \\ \quad \geqslant y(I_{T(k|t^* \to t_L, t_R)}(t_L)) & \\ y(I_{T(k|t^* \to t_L, t_R)}(t_L)), & \text{otherwise} \end{cases} \tag{9}$$

where $y(I_{T(k|t^* \to t_L, t_R)}(t_L)) = \max\{y(t') : t' \in I_{T(k|t^* \to t_L, t_R)}(t_L)\}$ and $y(t')$ is the estimation of the response variable at node $t' \in \widetilde{T}(k|t^* \to t_L, t_R)$. In words, $y(I_{T(k|t^* \to t_L, t_R)}(t_L))$ is the greatest estimation of variable y at Pareto-dominant nodes of t_L. Expression (9)

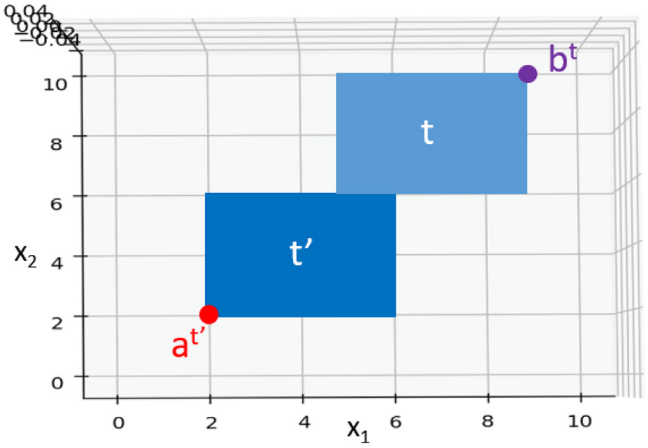

Fig. 5. Illustration of Proposition 2.

may be rewritten in a compact way as $y(t_L) = \max\{\max\{y_i : (\boldsymbol{x}_i, y_i) \in t_L\}, y(I_{T(k|t^* \to t_L, t_R)}(t_L))\}$.

A similar process is carried out with node t_R for determining an estimation of y at this node.

$$y(t_R) = \begin{cases} \max\{y_i : (\boldsymbol{x}_i, y_i) \in t_R\}, & \text{if } \max\{y_i : (\boldsymbol{x}_i, y_i) \in t_R\} \\ \quad \geqslant y(I_{T(k|t^* \to t_L, t_R)}(t_R)) & \\ y(I_{T(k|t^* \to t_L, t_R)}(t_R)), & \text{otherwise} \end{cases} \tag{10}$$

In this way, we are able to calculate the error made when y is estimated by $y(t_L)$ at node t_L and when y is estimated by $y(t_R)$ at node t_R, i.e. $R(t_L) = \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i) \in t_L} (y_i - y(t_L))^2$ and $R(t_R) = \frac{1}{n}\sum_{(\boldsymbol{x}_i, y_i) \in t_R} (y_i - y(t_R))^2$, respectively. The sum of these two errors is the function to be minimized. So, $\left(x^*_{j^*}, s^*_{j^*}\right)$ is defined as the combination of variables x_j and threshold $s_j \in S_j$, $j = 1, \ldots, m$, such that $R(t_L) + R(t_R)$ is minimum. Let also t^*_L and t^*_R be the child nodes associated with the split $\left(x^*_{j^*}, s^*_{j^*}\right)$. In this way, we have that $T_{k+1} = T(k|t^* \to t^*_L, t^*_R)$.

Once the tree $T(k|t^* \to t^*_L, t^*_R)$ is obtained, the way to update the parameters a and b corresponding of the supports of nodes t^*_L and t^*_R is as follows:

$$\begin{aligned} a_j^{t_L} &= a_j^{t^*}, \quad \forall j = 1, \ldots, m \\ b_j^{t_L} &= b_j^{t^*}, \quad \forall j = 1, \ldots, m, j \neq j^* \\ b_{j^*}^{t_L} &= s^*_{j^*}, \\ a_j^{t_R} &= a_j^{t^*}, \quad \forall j = 1, \ldots, m, j \neq j^* \\ a_{j^*}^{t_R} &= s^*_{j^*}, \\ b_j^{t_R} &= b_j^{t^*}, \quad \forall j = 1, \ldots, m \end{aligned} \tag{11}$$

After performing the $k+1$ split, we have an estimation of the response variable for each node belonging to $\widetilde{T}_{k+1}$. In particular, we have that $y_{T_{k+1}}(t) = y_{T_k}(t)$ if $t \neq t^*$ and, by (9) and (10), $y_{T_{k+1}}(t^*_L) = y(t^*_L)$ and $y_{T_{k+1}}(t^*_R) = y(t^*_R)$. Additionally, we define the estimation of the response variable corresponding to the root node t_0 as $y_{T_0}(t_0) = \max\{y_i : (\boldsymbol{x}_i, y_i) \in t_0\} = \max\{y_i : i = 1, \ldots, n\}$, i.e., this estimation coincides with the maximum value observed for the response variable in the learning sample.

In our approach, the way of selecting the node to be split among all the nodes belonging to $\widetilde{T}_k$ will be random. Additionally, there are a lot of possibilities of defining the sets S_j of thresholds for each variable x_j, $j = 1, \ldots, m$. In particular, we decided to implement a way based on the observations. So, for each node $t \in \widetilde{T}_k$, and each variable x_j, $j = 1, \ldots, m$, $S_j = \{x : \exists (\boldsymbol{x}_i, y_i) \in t \text{ such that } x = x_{ji}\}$. In words, the elements of set S_j are the observed values of variable x_j in the examples belonging to t.

The process is repeated until a split is found, such that every terminal node satisfies the stopping rule. Let K be the number of executed splits in the growing process that corresponds to this last split and let T_K be the final grown tree. So, $T_{\max} = T_K$. Now, we are ready to show probably the main result of this paper, which establishes that the predictor associated with the tree structure $T_{\max}$ meets free disposability. In particular, the predictor $d(\boldsymbol{x})$ defined from the information of $T_{\max}$ is as follows: $d_{T_{\max}}(\boldsymbol{x}) = y_{T_{\max}}(t)$, for $t \in \widetilde{T}_{\max}$ such that $\boldsymbol{a}^t \leqslant \boldsymbol{x} < \boldsymbol{b}^t$, i.e. $\boldsymbol{x} \in \text{supp}(t)$.

Theorem 1. *$d_{T_{\max}}(\boldsymbol{x}) : R^m_+ \to R$ is a monotone non-decreasing function.*

Proof. We have to prove that if $\boldsymbol{x} \leqslant \boldsymbol{x}'$, then $d_{T_{\max}}(\boldsymbol{x}) \leqslant d_{T_{\max}}(\boldsymbol{x}')$. First, we will prove that if $d_{T_k}(\boldsymbol{x})$ is a monotone non-decreasing function, then $d_{T_{k+1}}(\boldsymbol{x})$ is also a monotone non-decreasing function. The difference between the trees T_k and T_{k+1} is that one node of $\widetilde{T}_k$, say t^*, has been split into two child nodes, t^*_L and t^*_R. In this way, $T_{k+1} = T(k|t^* \to t^*_L, t^*_R)$. Furthermore, $d_{T_{k+1}}(\boldsymbol{x}) = d_{T_k}(\boldsymbol{x})$ for all $\boldsymbol{x} \in (R^m_+ - \text{supp}(t^*))$ and $d_{T_{k+1}}(\boldsymbol{x}) = y_{T_{k+1}}(t^*_L)$ if $\boldsymbol{x} \in \text{supp}(t^*_L)$ and $d_{T_{k+1}}(\boldsymbol{x}) = y_{T_{k+1}}(t^*_R)$ if $\boldsymbol{x} \in \text{supp}(t^*_R)$. Then, we have to study several cases. (i) Let us assume that $\boldsymbol{x}, \boldsymbol{x}' \notin \text{supp}(t^*)$. Then, $d_{T_{k+1}}(\boldsymbol{x}) = d_{T_k}(\boldsymbol{x})$ and $d_{T_{k+1}}(\boldsymbol{x}') = d_{T_k}(\boldsymbol{x}')$ since only the estimation related to the node t^* could have been modified after the $k+1$ split. Consequently, $d_{T_{k+1}}(\boldsymbol{x}) \leqslant d_{T_{k+1}}(\boldsymbol{x}')$ by the assumption that $d_{T_k}(\boldsymbol{x})$ is a monotone non-decreasing function. (ii) Let us assume that $\boldsymbol{x} \notin \text{supp}(t^*)$, $\boldsymbol{x}' \in \text{supp}(t^*)$. Then, we can assume that $\boldsymbol{x} \in \text{supp}(t)$ for some $t \in \widetilde{T}_{k+1}$. If $\boldsymbol{x}' \in t^*_L$, by Definition 2, $t \in I_{T_{k+1}}(t^*_L)$. By (9), $y_{T_{k+1}}(t^*_L) = \max\{\max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_L\}, y(I_{T(k|t^* \to t_L, t_R)}(t^*_L))\} \geqslant y(I_{T(k|t^* \to t_L, t_R)}(t^*_L)) \geqslant y_{T_k}(t)$. Now, $d_{T_{k+1}}(\boldsymbol{x}) \leqslant d_{T_{k+1}}(\boldsymbol{x}')$ since $y_{T_k}(t) = d_{T_k}(\boldsymbol{x}) = d_{T_{k+1}}(\boldsymbol{x})$ and $d_{T_{k+1}}(\boldsymbol{x}') = y_{T_{k+1}}(t^*_L)$. Otherwise, if $\boldsymbol{x}' \in t^*_R$, we can follow similar steps and achieve the same result. (iii) Let us assume that $\boldsymbol{x} \in \text{supp}(t^*), \boldsymbol{x}' \notin \text{supp}(t^*)$. By the hypothesis of monotonicity of $d_{T_k}(\boldsymbol{x})$, we have $d_{T_k}(\boldsymbol{x}) \leqslant d_{T_k}(\boldsymbol{x}')$. Recall that $d_{T_k}(\boldsymbol{x}') = d_{T_{k+1}}(\boldsymbol{x}')$ because $\boldsymbol{x}' \in (R^m_+ - \text{supp}(t^*))$. We are going to prove that $d_{T_{k+1}}(\boldsymbol{x}) \leqslant d_{T_k}(\boldsymbol{x})$. To do that, we first need to prove that $y_{T_k}(t^*) \geqslant \max\left\{y_{T_{k+1}}(t^*_L), y_{T_{k+1}}(t^*_R)\right\}$. By (9) and (10), we have $y_{T_{k+1}}(t^*_L) = \max\left\{\max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_L\}, y\left(I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_L)\right)\right\}$, and $y_{T_{k+1}}(t^*_R) = \max\left\{\max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_R\}, y\left(I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_R)\right)\right\}$. Moreover, $y_{T_k}(t^*) \geqslant \max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*\}$ by the way that the estimation of the response variable of a node is defined [expressions (9) and (10)] and that $y_{T_0}(t_0) = \max\{y_i : (\boldsymbol{x}_i, y_i) \in t_0\}$. Therefore, $y_{T_k}(t^*) \geqslant \max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_L\}$ and $y_{T_k}(t^*) \geqslant \max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_R\}$. Additionally, if $t' \in I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_L)$, then, by Definition 2, $\exists \hat{\boldsymbol{x}} \in \text{supp}(t^*_L)$ and $\exists \boldsymbol{x}'' \in \text{supp}(t')$ such that $\boldsymbol{x}'' \leqslant \hat{\boldsymbol{x}}$. But then, $t' \in I_{T_k}(t^*)$ since $\text{supp}(t^*_L) \subset \text{supp}(t^*)$. By the hypothesis of monotonicity of $d_{T_k}(\boldsymbol{x})$, we have $d_{T_k}(\boldsymbol{x}'') \leqslant d_{T_k}(\hat{\boldsymbol{x}})$. Nevertheless, note that $d_{T_k}(\boldsymbol{x}'') = y_{T_k}(t')$ and $d_{T_k}(\hat{\boldsymbol{x}}) = y_{T_k}(t^*)$. So, $y_{T_k}(t^*) \geqslant y_{T_k}(t')$ for all $t' \in I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_L)$. This implies that $y_{T_k}(t^*) \geqslant y\left(I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_L)\right)$. And, consequently, $y_{T_k}(t^*) \geqslant \max\left\{\max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_L\}, y\left(I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_L)\right)\right\} = y_{T_{k+1}}(t^*_L)$. By analogy, we may follow similar steps with respect to the right child node t^*_R and get that $y_{T_k}(t^*) \geqslant \max\left\{\max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_R\}, y\left(I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_R)\right)\right\} = y_{T_{k+1}}(t^*_R)$. Thus, $y_{T_k}(t^*) \geqslant \max\left\{y_{T_{k+1}}(t^*_L), y_{T_{k+1}}(t^*_R)\right\}$. In this way, we have that $d_{T_{k+1}}(\boldsymbol{x}) = y_{T_{k+1}}(t^*_L)$ if $\boldsymbol{x} \in t^*_L$ and $d_{T_{k+1}}(\boldsymbol{x}) = y_{T_{k+1}}(t^*_R)$ if $\boldsymbol{x} \in t^*_R$ and, additionally, $d_{T_k}(\boldsymbol{x}) = y_{T_k}(t^*)$. Hence, we get that $d_{T_{k+1}}(\boldsymbol{x}) \leqslant d_{T_k}(\boldsymbol{x})$. Recall that we had $d_{T_k}(\boldsymbol{x}) \leqslant d_{T_k}(\boldsymbol{x}') = d_{T_{k+1}}(\boldsymbol{x}')$. So, we get what we wanted to prove: $d_{T_{k+1}}(\boldsymbol{x}) \leqslant d_{T_{k+1}}(\boldsymbol{x}')$. (iv) Let us assume that $\boldsymbol{x}, \boldsymbol{x}' \in \text{supp}(t^*)$. If $\boldsymbol{x}, \boldsymbol{x}' \in \text{supp}(t^*_L)$ or $\boldsymbol{x}, \boldsymbol{x}' \in \text{supp}(t^*_R)$, then $d_{T_{k+1}}(\boldsymbol{x}) = d_{T_{k+1}}(\boldsymbol{x}')$ and we get the result that we desired. So, let us suppose that $\boldsymbol{x} \in \text{supp}(t^*_L)$ and $\boldsymbol{x}' \in \text{supp}(t^*_R)$. The other case is not possible since by (11) $\boldsymbol{a}^{t^*_L} = \boldsymbol{a}^{t^*}$ and $\boldsymbol{b}^{t^*} = \boldsymbol{b}^{t^*_R}$, by hypothesis $\boldsymbol{a}^{t^*} < \boldsymbol{b}^{t^*}$ [otherwise, $\text{supp}(t^*) = \varnothing$] and, by applying Proposition 2, we have that $t^*_L \in I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_R)$ and $t^*_R \notin I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_L)$. Now, by (10), $y_{T_{k+1}}(t^*_R) = \max\{\max\{y_i : (\boldsymbol{x}_i, y_i) \in t^*_R\}, y(I_{T(k|t^* \to t^*_L, t^*_R)}(t^*_R))\} \geqslant$

$y\left(I_{T(k|t^*\to t_L^*,t_R^*)}(t_R^*)\right) \geqslant y_{T_{k+1}}(t_L^*)$ since $t_L^* \in I_{T(k|t^*\to t_L^*,t_R^*)}(t_R^*)$. Thus, $d_{T_{k+1}}(\boldsymbol{x}) \leqslant d_{T_{k+1}}(\boldsymbol{x}')$. Finally, by induction, we get the desired result since $d_{T_0}(\boldsymbol{x})$ is trivially a monotone non-decreasing function.

For single-output production processes, free disposability is translated into monotonicity of the predictor. In this way, Theorem 1 indicates that the induced technology $\hat{\Psi}_{T_{\max}} := \left\{(\boldsymbol{x},y) \in R_+^{m+1} : y \leqslant d_{T_{\max}}(\boldsymbol{x})\right\}$, defined from the deep tree $T_{\max}$, satisfies the well-known property in microeconomics of free disposability. Following a similar reasoning, and due to the update process of the estimation of the output variable, every subtree T_k is also associated with a monotone non-decreasing predictor $d_{T_k}(\boldsymbol{x})$ and an induced technology $\hat{\Psi}_{T_k} := \left\{(\boldsymbol{x},y) \in R_+^{m+1} : y \leqslant d_{T_k}(\boldsymbol{x})\right\}$ satisfying free disposability. Furthermore, EAT generates predictors that are step functions on R_+^m like the technique known as FDH in frontier analysis. Consequently, it seems appropriate to compare both approaches.

Although, EAT and FDH does not yield the same predictors in general, the next proposition shows that both techniques build the same step function as output estimator when the number of variables x_j is one, i.e. $m = 1$, and the stop rule sets $n_{\min} = 1$.

Proposition 3. *Let $m = 1$ and $n_{\min} = 1$. Then, $d_{T_{\max}}(x) = \hat{f}_{FDH}(x)$ for all $x \in R_+$.*

Proof. Let us assume, without loss of generality, that the learning sample may be sorted in the way (x_1,y_1), (x_2,y_2), ..., (x_n,y_n), with $x_1 < x_2 < \ldots < x_n$. After applying the EAT algorithm, we would get that each learning example (x_i,y_i) would belong to a terminal node $t_i \in T_{\max}$ with a support in the way $\text{supp}(t_i) = [x_i, x_{i+1})$, $i = 1,\ldots,n$ $[x_{n+1} := +\infty]$. Let $x_{i^*} = \min_{i:\, y_i = \max_{1\leqslant k\leqslant n}\{y_k\}} \{x_i\}$. Then, $\hat{f}_{FDH}(x) = y_{i^*}$ for all $x \geqslant x_{i^*}$. Regarding the EAT technique, by construction, $d_{T_{\max}}(x) \leqslant \max_{1\leqslant k\leqslant n}\{y_k\} = y_{i^*}$ for any $x \in R_+$. Additionally, $d_{T_{\max}}(x) \geqslant y_{i^*}$ for all $x \geqslant x_{i^*}$. This last inequality is due to Theorem 1 and that $d_{T_{\max}}(x_{i^*}) \geqslant y_{i^*}$. Therefore, $d_{T_{\max}}(x) = y_{i^*}$ for all $x \geqslant x_{i^*}$ and $d_{T_{\max}}(x) = \hat{f}_{FDH}(x)$ for all $x \geqslant x_{i^*}$. Note that both the estimation generated from FDH and the estimation yielded by EAT for a point $x < x_{i^*}$ do not depend on the estimation of the response variable for values $x \geqslant x_{i^*}$. In the case of FDH, this fact is evident as consequence of its definition $\hat{f}_{FDH}(x) = \max_{i:x\geqslant x_i}\{y_i\}$. In the case of EAT, during the application of the corresponding algorithm, when the estimation of the response variable for any node t_i, $i = 1,\ldots,i^*-1$, must be determined, only nodes with a support contained in the segment $[x_1,x_i)$ could be Pareto-dominant nodes of t_i and, therefore, they are the unique nodes that can influence the final estimation for t_i. Now, we can repeat the process but focusing our attention on the learning subsample (x_1,y_1), (x_2,y_2), ..., (x_{i^*-1},y_{i^*-1}). If we repeat it as many times as needed, we get that $d_{T_{\max}}(x) = \hat{f}_{FDH}(x)$ for all $x \in R_+^m$.

Therefore, Fig. 1, which illustrates the typical shape of the output predictor determined by the application of the FDH technique, is also valid for the EAT technique assuming that $n_{\min} = 1$. However, Proposition 3 does not hold in general. Indeed, we will show in Section 4 through several numerical experiences that the discrepancies between FDH and EAT increase as the sample size and number of predictor variables x_j augment.

Next, we show a simple instance with $m = 2$ in order to illustrate some specific differences between the two techniques regarding the results obtained. Let us assume that our learning sample consists of the following examples (x_{1i},x_{2i},y_i): A = (1, 4, 2), B = (2, 2, 1), C = (3, 1, 3), D = (2, 6, 5), E = (3, 4, 1), F = (4, 3, 2), G = (5, 1, 1), H = (4, 8, 10), I = (5, 5, 6), J = (6, 4, 4) and K = (7, 2, 7). Then, if the FDH technique is applied, we get the following predictor of the efficient frontier (see Fig. 6). In this figure, we show the regions determined by FDH in the input space together with the data and the value of the output estimation.

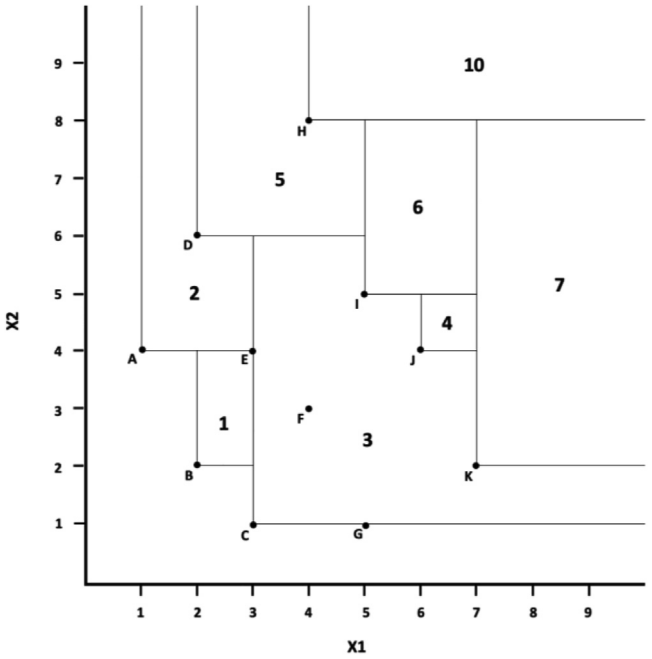

Fig. 6. Example of FDH with two inputs.

Regarding the EAT technique, playing with the same examples, it generates with $n_{\min} = 1$ the predictor shown in Fig. 7. Note that each terminal node of the tree is associated with a support in the figure, i.e. a rectangle. Note also that each node has at least one observation. Moreover, it is easy to check that both techniques do not yield the same estimation of the response variable for all $(x_1,x_2) \in R_+^2$. For example, points at the northwest corner of the fig-

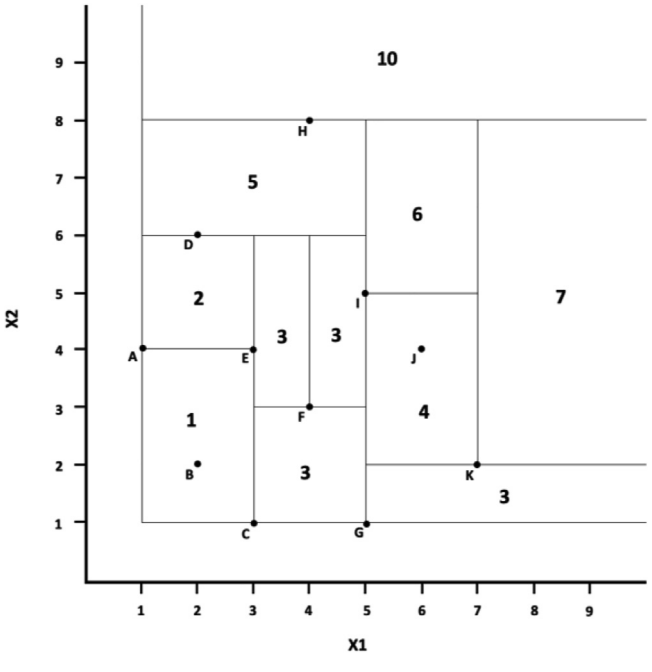

Fig. 7. Example of EAT with two inputs ($n_{\min} = 1$).

ure present an estimation of 2 in the case of FDH and 10 in the case of EAT.

Furthermore, although the FDH technique generates a step function as a predictor, like EAT, we are aware that it cannot be interpreted as a structure based on nodes and supports in the input space. Nevertheless, in order to reinterpret the results associated with the FDH technique that way, we could get the structure of supports shown in Fig. 8. In this structure, the node related to the support that appears at the northwest corner would be empty, i.e., no observation belongs to this support. The technique is able to determine an output estimation for it, due to the assumptions of the FDH methodology. However, usual data-driven approaches need data to determine an estimation in that region of the input space. EAT, like CART, is not able to determine terminal nodes without data, since to execute a split both require calculating the mean squared error in each child node, something that is only possible if we have at least one example in each corresponding support.

Unfortunately, as we mentioned above, FDH exhibits noticeable overfitting in the data. Indeed, FDH yields efficiency estimates that are overly optimistic. The same happens for the EAT technique when a deep tree is allowed to grow. So, at this point, one might think that the introduction of a new approach was pointless; except that it has allowed us to bridge two fields that have grown in parallel: frontier analysis and machine learning. However, it is not true. The new approach, which is based upon CART, may exploit certain techniques usually linked to CART, as pruning and cross-validation to overcome overfitting. Hereinafter, we will denote the tree resulting from the pruned process as T^*.

Overall, we suggest to always apply EAT along with pruning and cross-validation to determine a suitable estimation of production frontier. To do that, the standard process carried out by CART in regression for pruning should be slightly adapted to the context of frontier analysis. In Section 2.2, we mentioned how to prune a deep tree in CART through cross-validation. In the case of EAT, the process is identical except for how trees grow, which is based on the algorithm introduced in this section, and for the order of subtrees followed in the pruning process. In our context, the key for the modification of the standard approach will be fixing the way in which the sequence of subtrees is determined. Given that the EAT algorithm generates a size increasing sequence of trees in such a way that $\{t_0\} \prec T_1 \prec T_2 \prec \ldots \prec T_K = T_{\max}$, guaranteeing that each tree T_k satisfies free disposability, we will use the opposite sequence, i.e., $T_{\max} \succ T_{K-1} \succ T_{K-2} \succ \ldots \succ \{t_0\}$, as a sequence of subtrees for the pruning process.

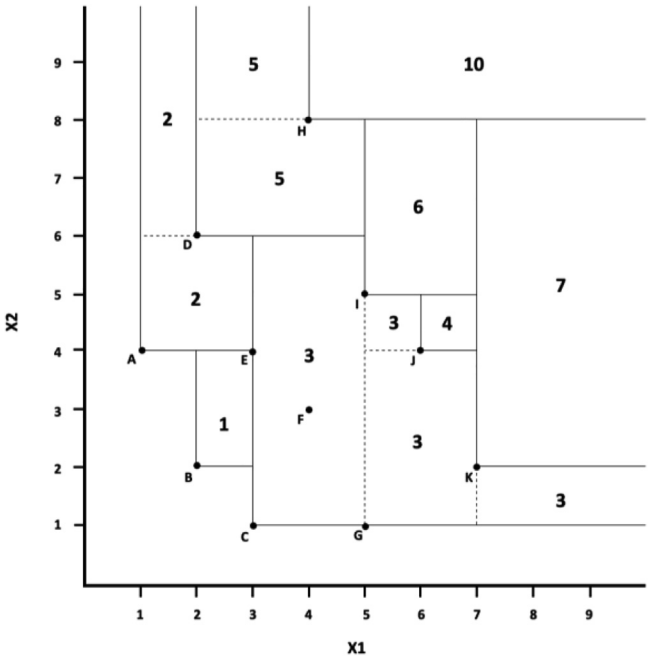

Fig. 8. Example of FDH as a regression tree.

In Fig. 9, we show the predictor generated by EAT after pruning, based upon the same example developed in Fig. 1.

Additionally, we show an example in three dimensions in Figs. 10 and 11. In Fig. 10, the predictor generated by EAT is shown, whereas Fig. 11 illustrates the EAT predictor after performing the pruning process.

The above discussion allows us to interpret the EAT technique as a data-driven way of estimating production frontiers based on the grounds of machine learning, determining a monotone non-decreasing step function as a predictor. In some sense, the new approach is similar to the standard FDH technique. However, while FDH suffers from a problem of overfitting, the EAT technique can be improved through pruning and cross-validation (Breiman et al., 1984) solving the initial problems.

Next, we establish the relationship that exists among the output estimation of FDH, EAT and pruned EAT.

Proposition 4. $d_{T^*}(\boldsymbol{x}) \geqslant d_{T_{\max}}(\boldsymbol{x}) \geqslant \hat{f}_{FDH}(\boldsymbol{x})$

Proof. It is straightforward.

3.2. The multi-output case

Several attempts have been proposed in the literature for extending single-response decision tree techniques to the multi-response context: De'Ath (2002), Appice and Džeroski (2007), Stojanova et al. (2012) and Levatić et al. (2014) to name but a few. Among them, De'Ath (2002) seems to be the natural extension of the CART univariate recursive partitioning method (see Borchani et al., 2015). This approach works by following the same steps as CART, i.e., starting with all data in the root node, then recursively finding the best possible split at parent nodes and partitioning the data accordingly until a certain stopping rule is satisfied. The main difference with the standard CART is the redefinition of the measure use for selecting the best combination $\left(x_{j^*}^*, s_{j^*}^*\right)$ at each split. Whereas in the single-response case, the criterion used coincides with the mean squared error associated with the unique response variable, in the multi-response case, the squared errors linked to all the response variables aggregate in a final measure. Lastly, each leaf of the tree generated by the De'ath approach is characterized by the multi-variate mean of the data belonging to it, as a direct generalization of the univariate standard CART. De'Ath (2002) also indicated some interesting features of the technique. The multi-variate trees are easy to construct and visualize. They are also robust to the addition of random noise. Moreover, they automatically detect the interactions between variables. Finally, they handle missing values in the predictor variables (the inputs in our context) with minimal loss of information.

Next, we show how to extend the univariate EAT algorithm to the multi-output context through the application of the De'ath approach. The steps are the same with changes in the split process. In particular, and inspired by the single-output algorithm of EAT, we suggest to estimate the value of the response variable y_r, $r = 1, \ldots, s$, through the formula $y_r(t_L) = \max\{\max\{y_{ri} : (\boldsymbol{x}_i, \boldsymbol{y}_i) \in t_L\}, y_r\left(I_{T(k|t^* \to t_L, t_R)}(t_L)\right)\}$, for the left child node, and the formula $y_r(t_R) = \max\{\max\{y_{ri} : (\boldsymbol{x}_i, \boldsymbol{y}_i) \in t_R\}, y_r\left(I_{T(k|t^* \to t_L, t_R)}(t_R)\right)\}$, for the right child node; where $y_r\left(I_{T(k|t^* \to t_L, t_R)}(t_L)\right) = \max\{y_r(t') : t' \in I_{T(k|t^* \to t_L, t_R)}(t_L)\}$, $y_r\left(I_{T(k|t^* \to t_L, t_R)}(t_R)\right) = \max\{y_r(t') : t' \in$

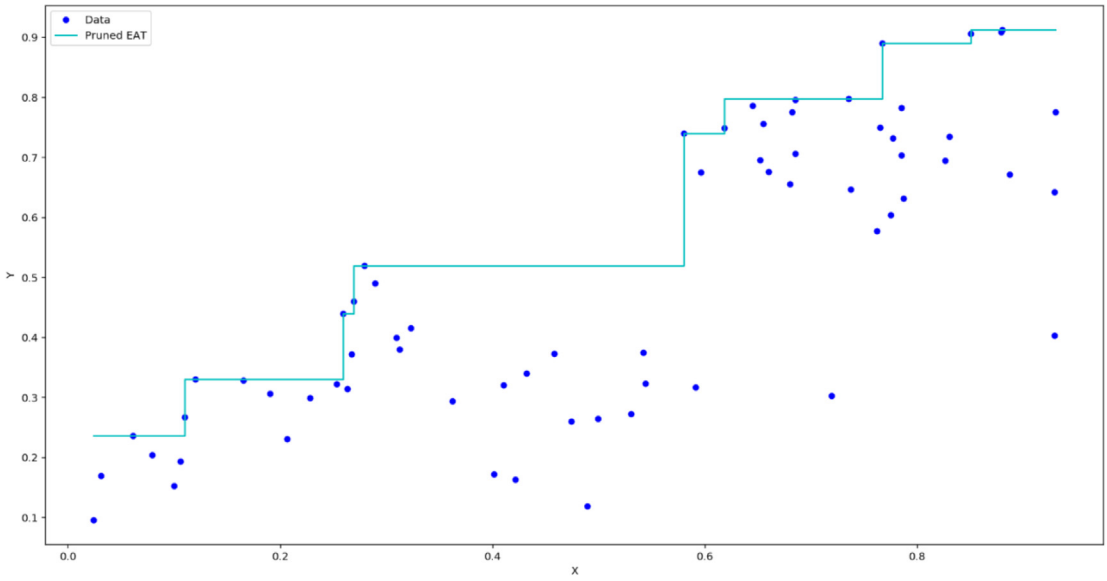

Fig. 9. Example of EAT estimate after the pruning process.

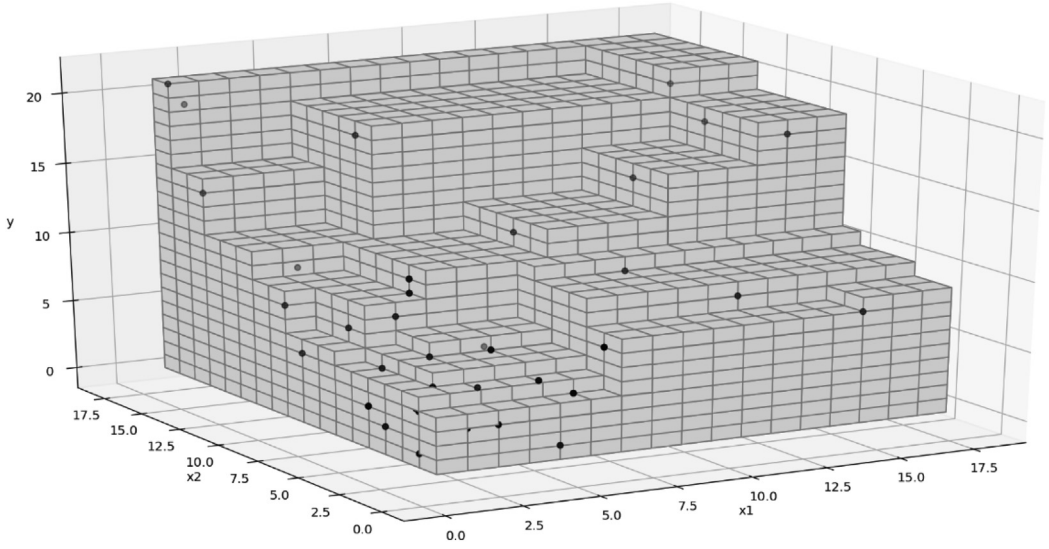

Fig. 10. Example of EAT in three dimensions.

$I_{T(k|t^*\to t_L,t_R)}(t_R)\}$ and $y_r(t')$ is the estimation of the response variable y_r at node $t' \in \widetilde{T}(k|t^* \to t_L, t_R)$, $r = 1, \ldots, s$. Additionally, the multi-variate EAT algorithm chooses the combination (x_j, s_j) which minimizes $R(t_L) + R(t_R) = \frac{1}{n}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in t_L}\sum_{r=1}^{s}(y_{ri} - y_r(t_L))^2 + \frac{1}{n}\sum_{(\boldsymbol{x}_i,\boldsymbol{y}_i)\in t_R} \sum_{r=1}^{s}(y_{ri} - y_r(t_R))^2$, i.e., the objective is minimizing the aggregation of the squared error over all the response variables. Finally, as in the single-output case, the splitting process is repeated until a split k' is found, such that for all $t \in \widetilde{T}_{k'}$ the stopping rule $n(t) \leqslant 5$ holds. Then, k' will be K and the grown tree will be the deep tree $T_{\max} = T_K$. Moreover, let $\boldsymbol{d}_{T_{\max}}(\boldsymbol{x}) = \boldsymbol{y}_{T_{\max}}(t)$, for $t \in \widetilde{T}_{\max}$ such that $\boldsymbol{x} \in \text{supp}(t)$, i.e., $\boldsymbol{d}_{T_{\max}}(\boldsymbol{x})$ is the multi-dimensional predictor defined from the tree $T_{\max}$. From this predictor, it is possible to define the technology or production possibility set induced by $\boldsymbol{d}_{T_{\max}}(\boldsymbol{x})$ as:

$$\hat{\Psi}_{T_{\max}} = \left\{(\boldsymbol{x},\boldsymbol{y}) \in R_+^{m+s} : \boldsymbol{y} \leqslant \boldsymbol{d}_{T_{\max}}(\boldsymbol{x})\right\} \tag{12}$$

However, although the proposed extension of the single output EAT algorithm seems valid *a priori* because it corresponds to a direct application of the multi-response CART approach by De'ath (2002), the multi-output EAT method would not be suitable for dealing with the estimation of production frontiers unless the estimations of the underlying technologies produced by this new algorithm satisfy the axiom of free disposability.

Proposition 5. *The set $\hat{\Psi}_{T_{\max}}$ meets free disposability.*

Proof. Let $(\boldsymbol{x},\boldsymbol{y}) \in \hat{\Psi}_{T_{\max}}$ and let $(\boldsymbol{x}',\boldsymbol{y}') \in R_+^{m+s}$ such that $\boldsymbol{x}' \geqslant \boldsymbol{x}$ and $\boldsymbol{y}' \leqslant \boldsymbol{y}$. We want to prove that $(\boldsymbol{x}',\boldsymbol{y}') \in \hat{\Psi}_{T_{\max}}$. On the one hand, we have that $\boldsymbol{y} \leqslant \boldsymbol{d}_{T_{\max}}(\boldsymbol{x})$ because $(\boldsymbol{x},\boldsymbol{y}) \in \hat{\Psi}_{T_{\max}}$. On the other hand, by the way we defined the update of the estimation of each component of the output vector $\boldsymbol{y}(t)$ for each node under the multi-output EAT algorithm, which follows the definition for the single output EAT approach, we have that if $\boldsymbol{x}' \geqslant \boldsymbol{x}$,

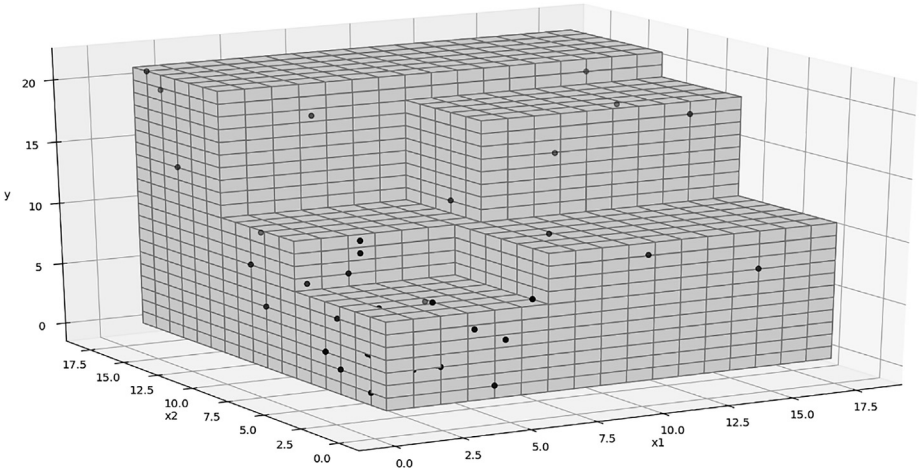

Fig. 11. Example of pruned EAT in three dimensions.

then $\boldsymbol{d}_{T_{\max}}(\boldsymbol{x}') \geqslant \boldsymbol{d}_{T_{\max}}(\boldsymbol{x})$ by Theorem 1. Therefore, $\boldsymbol{y}' \leqslant \boldsymbol{y} \leqslant \boldsymbol{d}_{T_{\max}}(\boldsymbol{x}) \leqslant \boldsymbol{d}_{T_{\max}}(\boldsymbol{x}')$, which implies, by (12), that $(\boldsymbol{x}', \boldsymbol{y}') \in \hat{\Psi}_{T_{\max}}$.

As we have shown, the multi-output EAT algorithm is able to yield estimates of a technology that satisfies the neoclassical axioms of microeconomics. In particular, it satisfies free disposability, as FDH. Additionally, the EAT algorithm generates step functions; as happens with FDH. However, FDH is also based on the postulate of minimal extrapolation, which states that the most conservative estimation of the production frontier would be that associated with a surface enveloping the data and as close as possible to them. Recall that the satisfaction of this approach is linked to the overfitting problem mentioned above in the text. A priori, the EAT algorithm is not grounded on the axiom of minimal extrapolation. Nevertheless, and given that the tree that has been generated at the moment in this subsection by the EAT approach is the deepest tree, in some sense, this tree as a predictor also suffers the problem of overfitting (see Breiman et al., 1984). The way of correcting the problem of overfitting with decision trees is well-known. One standard solution consists of pruning the deep tree by resorting to a cross-validation process. This is one of the advantages of relating the estimation of production frontiers to CART. In particular, for the multi-output case, we apply the same steps previously defined for the single-output EAT algorithm.

After the pruning process, we get a final subtree T^* that allows defining an estimation of the underlying technology as $\hat{\Psi}_{T^*} := \{(\boldsymbol{x}, \boldsymbol{y}) \in R_+^{m+s} : \boldsymbol{y} \leqslant \boldsymbol{d}_{T^*}(\boldsymbol{x})\}$, where $\boldsymbol{d}_{T^*}(\boldsymbol{x})$ is the predictor associated with T^*. As before for $\hat{\Psi}_{T_{\max}}$, it is possible to prove that $\hat{\Psi}_{T^*}$ meets the axiom of free disposability. Nevertheless, T^* is not as deep as $T_{\max}$.

The resemblances between the standard FDH and the multi-output EAT can be summarized in the next result, which establishes that the Free Disposal Hull generated from a certain 'virtual' learning sample coincides with the estimation of the underlying technology yielded by the multi-output EAT algorithm. The examples that are part of this special learning sample are the following points in the input-output space: $(\boldsymbol{a}^t, \boldsymbol{d}_{T^*}(\boldsymbol{a}^t))$, for all $t \in \tilde{T}^*$. In words, they are 'virtual' units, i.e. they are not necessarily observed in the original learning sample, which consume the input vector $\boldsymbol{a}^t$, corresponding to the bottom corner of the support linked to the leaf node t in tree T^*, and produces the output vector $\boldsymbol{d}_{T^*}(\boldsymbol{a}^t)$, corresponding to the EAT estimation of the set of outputs for input vector $\boldsymbol{a}^t$.

Proposition 6. *The FDH technology constructed from the set of points* $\{(\boldsymbol{a}^t, \boldsymbol{d}_{T^*}(\boldsymbol{a}^t))\}_{t \in \tilde{T}^*}$ *coincides with* $\hat{\Psi}_{T^*}$:

$$\hat{\Psi}_{T^*} = \left\{ (\boldsymbol{x}, \boldsymbol{y}) \in R_+^{m+s} : \exists t \in \tilde{T}^* \text{ such that } \boldsymbol{y} \leqslant \boldsymbol{d}_{T^*}(\boldsymbol{a}^t), \boldsymbol{x} \geqslant \boldsymbol{a}^t \right\}. \tag{13}$$

Proof. Let us denote the right-hand side of expression (13) as Ψ_{FDH}^{EAT}. Let $(\boldsymbol{x}, \boldsymbol{y}) \in \hat{\Psi}_{T^*}$. Then, by definition, $\boldsymbol{y} \leqslant \boldsymbol{d}_{T^*}(\boldsymbol{x})$. Let $t_x \in \tilde{T}^*$ such that $\boldsymbol{x} \in \text{supp}(t_x)$. Then, $\boldsymbol{d}_{T^*}(\boldsymbol{x}) = \boldsymbol{y}_{T^*}(t_x)$. Note that $\boldsymbol{a}^{t_x} \in \text{supp}(t_x)$, which implies that $\boldsymbol{d}_{T^*}(\boldsymbol{a}^{t_x}) = \boldsymbol{y}_{T^*}(t_x)$. Additionally, we have that $\boldsymbol{a}^{t_x} \leqslant \boldsymbol{x}$ because $\boldsymbol{x} \in \text{supp}(t_x)$. In this way, we have that $\boldsymbol{d}_{T^*}(\boldsymbol{x}) = \boldsymbol{y}_{T^*}(t_x) = \boldsymbol{d}_{T^*}(\boldsymbol{a}^{t_x})$ and $\boldsymbol{x} \geqslant \boldsymbol{a}^{t_x}$. Consequently, $\exists t \in \tilde{T}^*$ such that $\boldsymbol{y} \leqslant \boldsymbol{d}_{T^*}(\boldsymbol{a}^t)$ and $\boldsymbol{x} \geqslant \boldsymbol{a}^t$ and, therefore, $(\boldsymbol{x}, \boldsymbol{y}) \in \Psi_{FDH}^{EAT}$. Let now $(\boldsymbol{x}, \boldsymbol{y}) \in \Psi_{FDH}^{EAT}$. Then, $\exists t \in \tilde{T}^*$ such that $\boldsymbol{y} \leqslant \boldsymbol{d}_{T^*}(\boldsymbol{a}^t)$ and $\boldsymbol{x} \geqslant \boldsymbol{a}^t$. By the way we defined the update of the estimation of each component of the output vector $\boldsymbol{y}(t)$ for each node under the multi-output EAT algorithm, we have that if $\boldsymbol{x}' \geqslant \boldsymbol{x}$, then $\boldsymbol{d}_{T^*}(\boldsymbol{x}') \geqslant \boldsymbol{d}_{T^*}(\boldsymbol{x})$. Hence, $\boldsymbol{d}_{T^*}(\boldsymbol{x}) \geqslant \boldsymbol{d}_{T^*}(\boldsymbol{a}^t)$. Then, we have $\boldsymbol{y} \leqslant \boldsymbol{d}_{T^*}(\boldsymbol{a}^t) \leqslant \boldsymbol{d}_{T^*}(\boldsymbol{x})$, which implies, by the definition of $\hat{\Psi}_{T^*}$, that $(\boldsymbol{x}, \boldsymbol{y}) \in \hat{\Psi}_{T^*}$.

Proposition 6 has more important implications. This result shows how to calculate any measure of technical efficiency using the estimation $\hat{\Psi}_{T^*}$ as a basis. In the case of the output-oriented radial measure, the efficiency score $\phi(\boldsymbol{x}_k, \boldsymbol{y}_k)$ can be estimated by plugging $\hat{\Psi}_{T^*}$ into (2) in place of Ψ. In our case, and thanks to the relationship between the FDH and the multi-output EAT technique established in Proposition 6, by analogy with (4), the mixed-integer linear program that one should solve is as follows:

$$\phi^{EAT}(\boldsymbol{x}_k, \boldsymbol{y}_k) = \max_{s.t.} \begin{array}{c} \phi \\ \sum_{t \in \tilde{T}^*} \lambda_t \boldsymbol{a}_j^t \leqslant x_{jk},\ j = 1, \ldots, m \\ \sum_{t \in \tilde{T}^*} \lambda_t \boldsymbol{d}_{rT^*}(\boldsymbol{a}^t) \geqslant \phi y_{rk},\ r = 1, \ldots, s \\ \sum_{t \in \tilde{T}^*} \lambda_t = 1, \\ \lambda_t \in \{0, 1\},\ i = 1, \ldots, n \end{array} \tag{14}$$

In the next section, we show how the new approach performs in comparison with FDH in several numerical experiences.

4. Monte Carlo simulations

This section describes simulation results that serve for the comparison of methods: FDH vs (pruned) EAT. To do that, we present a

systematic comparison of these two frontier estimation methods in four alternative simulated environments for the single-output case, and in another one for the multi-output case. The description of the five scenarios appears in Table 1.

Scenario 1 represents a single-input case with a very well-known production function using the logarithmic function. Scenarios 2 and 3 involve two and three inputs, respectively. For all scenarios, we tested three data set sizes of 50, 100 and 150 observations. The input data were randomly sampled from $Uni[1,10]$ independently for each input and observation. Then, the efficient output level was calculated and a random inefficiency term $u \sim |N(0,0.4)|$ was subtracted to obtain the data used for the analysis. We ran 100 trials ($l = 1,...,100$) for each combination of scenario and data set size to investigate the relative performance of the methods. These three scenarios were taken from Kuosmanen and Johnson (2010). The fourth scenario, however, was taken from Lee and Cai (2020), where one output and nine inputs were considered.

Additionally, we simulated a multi-input ($m = 2$) and multi-output ($s = 2$) data set. We followed the ideas proposed by Perelman and Santin (2009). The multi-output situation is more difficult to be simulated. In this case, the difficulty is associated with how to generate suitable data for multi-output production processes satisfying microeconomic behavioral regularity conditions. As in Perelman and Santín (2009), we considered a half-normal distribution for generating the inefficiency term and we also incorporated two independent random statistical perturbations (random noise), allowing random shocks to affect outputs in a different quantity and direction. Moreover, we allowed 0%, 10% and 25% of the simulated DMUs to be on the true frontier. We ran 100 trials ($l = 1,...,100$) for each combination of sample size and percentage of units on the frontier.

Table 1
Scenarios to be simulated.

Scenario	#Outputs| #Inputs	Functional form $f(x)$
(1)	1|1	$f(x) = \ln(x) + 3$
(2)	1|2	$f(x) = 0.1x_1 + 0.1x_2 + 0.3(x_1x_2)^{1/2}$
(3)	1|3	$f(x) = 0.1x_1 + 0.1x_2 + 0.1x_3 + 0.3(x_1x_2x_3)^{1/3}$
(4)	1|9	$f(x) = \prod_{j=1}^{9} x_j^{\frac{1}{10}}$
(5)	2|2	$-\ln y_1 = -1 + 0.5 \cdot \left(\frac{\ln y_2}{\ln y_1}\right) + 0.25 \cdot \left(\frac{\ln y_2}{\ln y_1}\right)^2 - 1.5 \cdot (\ln x_1)$ $-0.6 \cdot (\ln x_2) + 0.2 \cdot (\ln x_1)^2 + 0.05 \cdot (\ln x_2)^2$ $-0.1 \cdot (\ln x_1) \cdot (\ln x_2) + 0.05 \cdot (\ln x_1) \cdot \left(\frac{\ln y_2}{\ln y_1}\right)$ $-0.05 \cdot (\ln x_2) \cdot \left(\frac{\ln y_2}{\ln y_1}\right)$

Performance of each method is evaluated by two standard criteria: the mean squared error (MSE) and the bias. The MSE statistic is defined as $\sum_{l=1}^{100}\sum_{i=1}^{n}\left(d_{T_l^*}(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_i^l)\right)^2/100n$ for the EAT technique and as $\sum_{l=1}^{100}\sum_{i=1}^{n}\left(\hat{f}_{FDH_l}(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_i^l)\right)^2/100n$ for the FDH method, where l is associated with the simulated data and the EAT and FDH predictors at trial l, $l = 1,...,100$. As usual, the MSE statistic measures the precision of estimates in quadratic terms. The bias statistic, however, is calculated in two different ways. The first one as $\sum_{l=1}^{100}\sum_{i=1}^{n}\left(d_{T_l^*}(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_i^l)\right)/100n$ for the EAT approach and as $\sum_{l=1}^{100}\sum_{i=1}^{n}\left(\hat{f}_{FDH_l}(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_i^l)\right)/100n$ for the FDH technique. In this case, the sign of the bias statistic indicates whether the estimated frontier systematically underestimates (negative sign) or overestimates (positive sign) the true frontier. However, since positive and negative deviations cancel out when we average over all observations and trials, we also calculate a second bias based on the absolute value of the deviations: $\sum_{l=1}^{100}\sum_{i=1}^{n}\left|d_{T_l^*}(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_i^l)\right|/100n$ for EAT and $\sum_{l=1}^{100}\sum_{i=1}^{n}\left|\hat{f}_{FDH_l}(\boldsymbol{x}_i^l) - f(\boldsymbol{x}_i^l)\right|/100n$ for FDH (see Kuosmanen and Johnson, 2010).

Table 2 reports the MSE and bias statistics for the two assessed approaches in the single-output scenarios considered and the measure of discrepancy between (deep) EAT and FDH. The first two columns indicate the scenario and the sample size. The next two columns indicate the MSE of the EAT and FDH techniques. The next two columns show the bias, whereas the following two are related to the bias based on absolute value. Finally, the last column in Table 2 reports the mean of the percentage of observations over all the trials in which the measure of discrepancy between (deep) EAT and FDH is greater or equal to 10%. Moreover, we have also calculated (and reported in brackets) the relative difference between the pruned EAT and FDH with respect to MSE and bias in order to make the comparison of the results easier. These values could be seen as the percentages of reduction of the MSE and bias when pruned EAT is applied instead of FDH.

Regarding the results, all methods were affected by the increase in dimensionality from a single input to multiple inputs since MSE increases as the number of inputs augments. Also, MSE of the pruned EAT method was smaller than the MSE of the FDH technique for all scenarios and sample sizes. The improvements ranged from 13% to 70%. A certain trend is observed in the results. In particular, the bigger the sample size is, the greater the improvement. Additionally, the scenario with nine inputs is the one with the biggest differences between the two approaches considered in this analysis. As for the bias, pruned EAT outperforms FDH for all the computational experiences carried out, with a reduction ranging from 12% to 93%, in the case of the first bias measure calculated,

Table 2
Results for scenarios (1)–(4).

Scenario	Number of obs.	Mean squared error		Bias		Absolute Bias		Discrepancies FDH & Deep EAT
		FDH	Pruned EAT	FDH	Pruned EAT	FDH	Pruned EAT	
(1)	50	0.032	0.028 (13%)	−0.145	−0.115 (21%)	0.145	0.130 (10%)	0%
(1)	100	0.019	0.015 (18%)	−0.109	−0.079 (28%)	0.109	0.093 (15%)	0%
(1)	150	0.013	0.010 (19%)	−0.089	−0.057 (36%)	0.089	0.076 (15%)	0%
(2)	50	0.121	0.102 (16%)	−0.281	−0.241 (14%)	0.281	0.253 (10%)	2.23%
(2)	100	0.104	0.086 (18%)	−0.268	−0.235 (12%)	0.268	0.240 (10%)	4.27%
(2)	150	0.091	0.073 (19%)	−0.251	−0.218 (13%)	0.251	0.222 (12%)	5.93%
(3)	50	0.146	0.109 (25%)	−0.297	−0.192 (35%)	0.297	0.250 (16%)	5.46%
(3)	100	0.136	0.097 (29%)	−0.287	−0.188 (35%)	0.287	0.236 (18%)	12.72%
(3)	150	0.130	0.086 (34%)	−0.278	−0.188 (33%)	0.278	0.220 (21%)	20.59%
(4)	50	0.037	0.013 (65%)	−0.161	−0.048 (70%)	0.161	0.089 (45%)	46.00%
(4)	100	0.038	0.011 (70%)	−0.164	−0.028 (83%)	0.164	0.084 (49%)	62.00%
(4)	150	0.037	0.011 (70%)	−0.161	−0.011 (93%)	0.161	0.082 (49%)	65.33%

and from 10% to 49% in the case of the bias based upon the absolute value. Finally, the measure of discrepancy between the (deep) EAT and the FDH shows that it is zero in the context of considering only one input (scenario 1), something that was expected following Proposition 3; relatively low in the case of scenario 2 based on two inputs; and is higher in the case of the third and fourth scenarios with more inputs. In all the cases, the discrepancy observed increases as the sample size augments. Fig. 12 shows an example of the result of one of our simulations.

Another feature of the EAT technique is that the estimated frontier can be graphically illustrated in an easy way taking advantage of its tree structure. This is important from a data visualization point of view, and this characteristic contrasts to the limitations of the FDH in more than two or three dimensions. For example, Fig. 13 shows the final tree obtained after executing the algorithm corresponding to the (pruned) EAT technique for one of the trials associated with scenario 3, where three inputs are consumed for producing one output. In each node, we can find an identification number, the mean squared error, the sample size, the split executed and the estimation of the output. A label like "X3 < 7.69 | X3 $\geq$ 7.69" means that the left child node is associated with the examples of the parent node that satisfies the condition $x_3 < 7.69$, while the right child node is related to examples that meets $x_3 \geqslant 7.69$. At the bottom of the tree, there are the terminal nodes (the leaves) where the final estimation of the response variable (the output) appears. In this manner, we can visualize the stepwise frontier by visiting the different branches of the tree from top to bottom.

Additionally, and in order to show another advantage of the EAT technique with respect to FDH, we calculate a measure of importance of each input variable in the analysis (see Breiman et al., 1984) for the numerical example analyzed in Fig. 13. In this way, we get the following values. The most important variable is x_2 with a (normalized) score of 100. The second variable in the ranking is x_1 with a value of 62.5, while the less important variable is x_3 with a score of 52.5.

Tables 3 and 4 show the results associated with the multi-output multi-input simulation. The structure of these tables is similar to Table 2, except for the fact that we now consider the percentage of DMUs on the true frontier and the possibility of random noise. Again, we observe that the EAT outperforms FDH with respect to MSE and Bias.

It is worth mentioning that a drawback of the new approach in comparison with the FDH technique is the computing time spent. In this section, the experiments were conducted on a PC with a 2.6 GHz dual-core Intel Core i5 processor, 8 Gigabyte of RAM and operating system OS X 10.12.1. The algorithm was implemented in a Python code. CPLEX v12.8 was used as the kernel for solving the optimization problems; the default options were used. Regarding the execution time, for an instance consisting in two outputs, two inputs and 200 DMUs, the FDH technique used 20.04 s for getting all the efficiency scores, whereas the EAT technique utilized 85.55 s (approximately four times more).

Finally, we focus our attention on the curse of dimensionality. In the case of technical efficiency measures, this problem is linked to the lack of discrimination between efficient and inefficient DMUs. It is a drawback usually related to the use of too many inputs and outputs in comparison with the sample size of the problem. We next show how the EAT algorithm works when it is applied to a real dataset of 15 DMUs, using 6 inputs to produce 6 outputs, although we are aware that this is an issue that merits further research. To do that, we refer to the data set of Fortune Magazine's US 15 best cities in 1996 (see Charles et al., 2019). In the example (Table 5), the FDH technique shows all DMUs as technically efficient, whereas the new approach may determine a score that helps to discriminate between the efficient and inefficient cities (only 5 out of 15 are technically efficient).

5. Conclusions and future work

In this paper, a bridge between frontier analysis and machine learning has been built. So far, these two fields have been growing in parallel with few contact points. However, they clearly present certain connections and the new wave in Operations Research linked to big data, Data Science and machine learning strongly encourages efficiency analysis researchers to harness the data analytics field (see, for example, the recent paper by Khezrimotlagh et al., 2019). In our case, this has meant introducing a new method to estimate production frontiers by growing trees, called Efficiency Analysis Trees (EAT). The new technique is clearly inspired in the famous CART (Classification and Regression Trees) by Breiman et al. (1984). In EAT, the minimization of the mean squared error is chosen as a criterion to recursively generate binary partitions

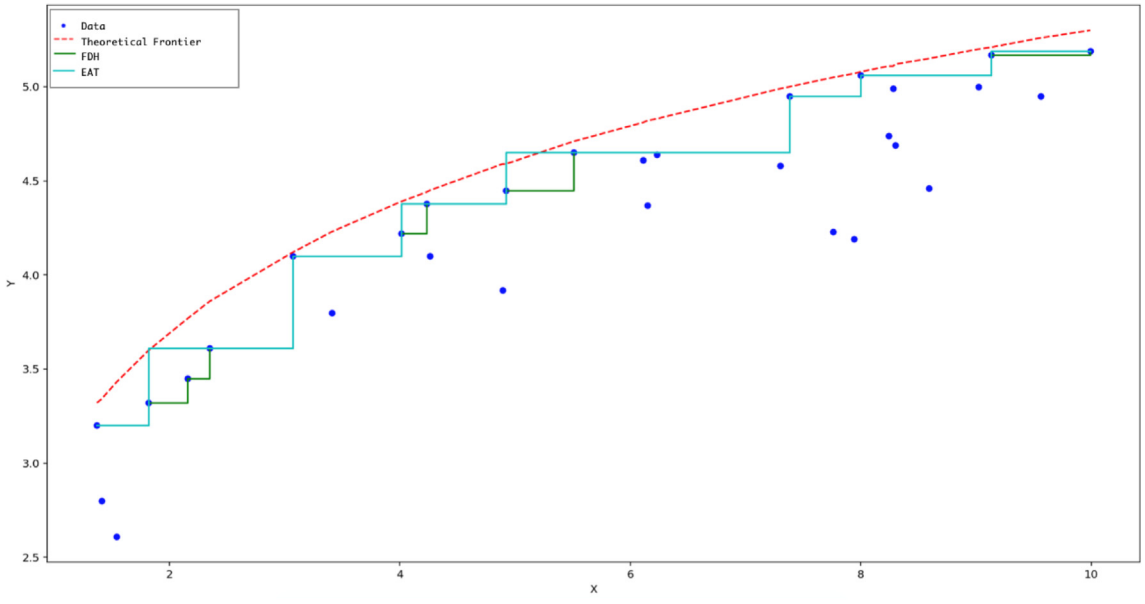

Fig. 12. Example of illustrating the result of one of our simulations.

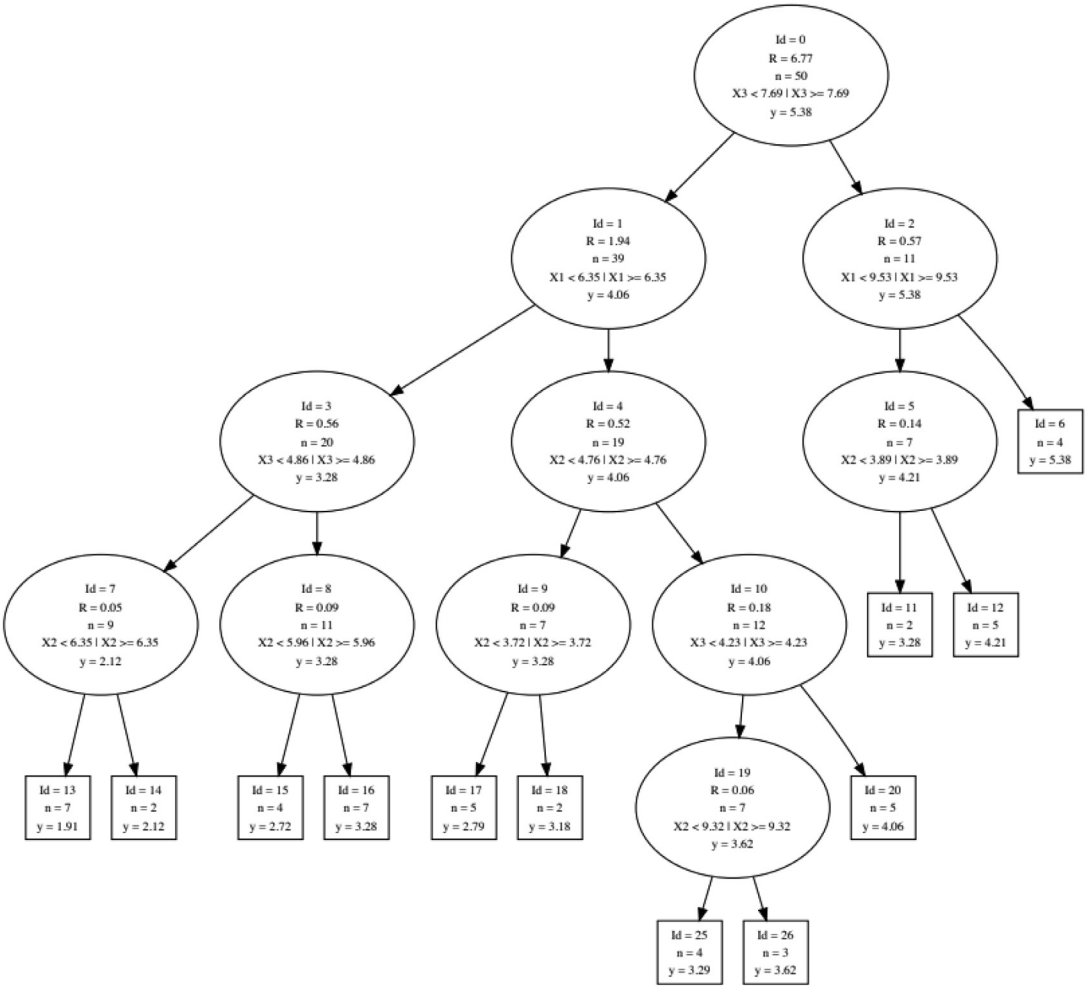

Fig. 13. Example of pruned EAT with three inputs visualized as a tree.

Table 3
Multi-output simulation: Results without random noise.

No. of obs.	%. of obs. on the true frontier	MSE			Absolute Bias		
		FDH	Deep EAT	EAT	FDH	Deep EAT	EAT
50	25	0.394	0.279	0.346 (12%)	0.398	0.327	0.339 (15%)
50	10	0.481	0.361	0.368 (24%)	0.480	0.403	0.371 (23%)
50	0	0.567	0.431	0.405 (29%)	0.549	0.473	0.411 (25%)
100	25	0.309	0.187	0.190 (38%)	0.357	0.265	0.258 (28%)
100	10	0.396	0.253	0.240 (39%)	0.440	0.341	0.304 (31%)
100	0	0.506	0.336	0.250 (51%)	0.525	0.419	0.341 (35%)
200	25	0.221	0.111	0.121 (45%)	0.301	0.203	0.205 (32%)
200	10	0.311	0.179	0.159 (49%)	0.394	0.284	0.248 (37%)
200	0	0.398	0.237	0.175 (56%)	0.474	0.357	0.287 (39%)

Table 4
Multi-output simulation: Results with random noise.

No. of obs.	%. of obs. on the true frontier	MSE			Bias		
		FDH	Deep EAT	EAT	FDH	Deep EAT	EAT
50	25	0.398	0.263	0.380 (4%)	0.398	0.318	0.342 (14%)
50	10	0.506	0.371	0.368 (27%)	0.478	0.396	0.369 (23%)
50	0	0.596	0.440	0.429 (28%)	0.555	0.468	0.418 (25%)
100	25	0.301	0.182	0.208 (31%)	0.349	0.258	0.265 (24%)
100	10	0.413	0.257	0.261 (37%)	0.447	0.341	0.313 (30%)
100	0	0.482	0.306	0.258 (46%)	0.515	0.400	0.342 (34%)
200	25	0.481	0.361	0.368 (24%)	0.480	0.403	0.371 (23%)
200	10	0.313	0.171	0.137 (56%)	0.390	0.271	0.240 (38%)
200	0	0.400	0.228	0.198 (50%)	0.471	0.343	0.291 (38%)

Table 5
Inputs and outputs of Fortune's best US cities with efficiency scores.

DMU	x_1	x_2	x_3	x_4	x_5	x_6	y_1	y_2	y_3	y_4	y_5	y_6	FDH score	EAT score
Seattle	586,000	581	1.45	4.50	21	542.3	46,928	0.297	4.49	7	117	22	1.00	1.00
Denver	475,000	558	0.97	4.00	14	595.6	42,879	0.291	2.79	5	60	71	1.00	1.09
Philadelphia	201,000	600	1.50	4.75	21	693.6	43,576	0.227	3.64	25	216	166	1.00	1.00
Minneapolis	299,000	609	1.49	4.00	24	496.5	45,673	0.270	2.67	6	131	125	1.00	1.03
Ral-Durham	318,000	613	0.99	4.50	18	634.7	40,990	0.319	4.94	7	33	47	1.00	1.00
St. Louis	265,000	558	0.89	3.00	18	263.0	39,079	0.206	3.40	10	104	62	1.00	1.20
Cincinnati	467,000	580	1.25	3.75	20	551.5	38,455	0.199	2.80	4	71	94	1.00	1.22
Washington	583,000	625	1.29	3.75	33	714.5	54,291	0.373	3.35	30	148	105	1.00	1.00
Pittsburgh	347,000	535	0.99	3.75	17	382.1	34,534	0.188	3.66	8	124	112	1.00	1.35
Dallas-FW	296,000	650	1.50	5.00	18	825.4	41,984	0.271	1.96	3	98	77	1.00	1.12
Atlanta	600,000	740	1.19	6.75	20	846.6	43,249	0.263	2.23	9	118	102	1.00	1.09
Baltimore	575,000	775	0.99	3.99	18	1296.3	43,291	0.233	4.02	8	102	45	1.00	1.08
Boston	351,000	888	1.09	4.25	34	686.6	46,444	0.325	5.69	25	240	55	1.00	1.00
Milwaukee	283,000	727	1.53	3.50	26	518.9	41,841	0.214	3.11	6	52	50	1.00	1.12
Nashville	431,000	695	1.19	4.00	26	1132.5	40,221	0.215	3.25	4	37	37	1.00	1.17

of the data (the training sample) until no further meaningful division is possible or a stopping rule holds. The graphical result of this process is a tree that begins at the root node, develops through intermediate nodes, and ends at the terminal nodes (leaves). In contrast to CART, EAT is able to estimate production frontiers since it has been defined to capture maximum trends instead of mean trends and to guarantee the satisfaction of free disposability.

Through the new approach, the input space is partitioned by a sequence of binary splits into terminal nodes. In each terminal node, the predicted response value (the output) is constant. So, graphically, the predictor looks like a step function and presents similarities and differences with respect to FDH when a deep tree is built. In particular, we showed that in the simple production case of one input and one output, FDH and EAT generate the same estimate of the efficiency frontier. However, while FDH suffers from a problem of overfitting, the EAT technique can be improved through pruning and cross-validation (Breiman et al., 1984), solving the initial problems. This connection between Free Disposal Hull and Efficiency Analysis Trees further contributes to developing the Data Science foundation of FDH, opening up new ways for adapting and integrating machine learning techniques to the efficiency analysis world.

Historically speaking, before the introduction of CART by Breiman et al. (1984), there were already other tree generating techniques for regression, such as AID (Automatic Interaction Detection) by Morgan and Sonquist (1963). So, the utilization of regression trees by Breiman et al. (1984) was not original at all, except for one crucial point: avoiding the problem of overfitting through a data-driven objective process. As Breiman et al. (1984, p. 216) claim: "*The major difference between AID and CART lies in the pruning and estimation process, that is, in the process of 'growing an honest tree*'". In the same way, we would like to highlight a certain parallelism between AID and CART and FDH and EAT, the latter defined for estimating production frontiers. And, in some sense, EAT could be interpreted as a 'pruned' FDH or a FDH-type out-of-sample predictor, overcoming its problem of data overfitting if the aim is to estimate the true theoretical frontier. Binary trees could give an interesting way of looking at data in frontier analysis. EAT should not be used to the exclusion of other frontier methods. Nevertheless, we believe that EAT does add a flexible and interesting new nonparametric tool to the data analyst's toolbox.

Additionally, performance of the new approach was investigated via Monte Carlo simulation. The results indicated that (pruned) EAT outperforms FDH with respect to several traditional error measures like the mean squared error (MSE), the bias and the bias based on the absolute value. Regarding the MSE, we observed that the determined improvements ranged from 13% to 70% in our simulations. Additionally, it was observed that the bigger the sample size, the higher the reduction in the MSE.

The new technique presents both additional advantages and drawbacks in comparison with FDH. As for the extra advantages, it is worth mentioning that the EAT methodology permits the graphical representation of the estimated production frontier through trees, even in the case of dealing with a high number of inputs. This could be an interesting feature from a data visualization viewpoint. Moreover, EAT also allows to determine a ranking of inputs with respect to the importance of these variables regarding the prediction of the output variable (the response variable). These two properties positively contrast with the standard FDH technique.

Finally, we finish by mentioning several lines that pose interesting avenues for further research. The first one is the possibility of extending the EAT technique in such a way that the splits yield more than two child nodes in each iteration of the tree growth algorithm. Other lines of research are related to solving some weaknesses of the technique. In particular, the standard CART presents a series of well-known disadvantages that are probably inherited by the new EAT technique: (1) instability, i.e., decision trees are unstable due to their oversensitivity to the training set, irrelevant predictors, and noise; (2) the fragmentation problem, i.e., each leaf node can be made up of a relatively small number of instances and, consequently, its prediction confidence is limited; (3) decision trees tend to perform well if a few highly relevant predictors exist, performing less well if many complex interactions among these variables are present (see Rokach, 2019). Most of these drawbacks of the (single) decision trees may be mitigated by growing a (random) forest of trees. Nowadays, random forest (Breiman, 2001) is one of the most applied extensions of CART in practice, since it allows to deal with the notion of robustness regarding the predictor variables considered and the observed dataset. In this sense, extending EAT for dealing with random forests seems a natural future research line. A third evident research line to be followed is the application of the new approach to real databases in different empirical contexts, thus checking the validity of the technique in practice. Other research lines could be analyzing in detail the multi-output framework, considering more outputs and inputs, and the curse of dimensionality.

CRediT authorship contribution statement

Miriam Esteve: Methodology, Software, Data curation, Visualization, Investigation, Validation, Writing - original draft, Writing - review & editing. **Juan Aparicio:** Conceptualization, Methodol-

ogy, Writing - original draft, Writing - review & editing, Funding acquisition. **Alejandro Rabasa:** Conceptualization, Writing - review & editing. **Jesus J. Rodriguez-Sala:** Software, Data curation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank two anonymous reviewers for providing constructive comments and help in improving the contents and presentation of this paper. Additionally, the authors are grateful for the financial support from the Spanish Ministry for Economy and Competitiveness, the State Research Agency and the European Regional Development Fund under grant PID2019-105952GB-I00 (AEI/FEDER, UE). This work was also supported by the Spanish Ministry of Science, Innovation and Universities under Grant FPU17/05365.

References

Afriat, S. N. (1972). Efficiency estimation of production functions. *International Economic Review*, 568–598.
Aparicio, J., Pastor, J. T., Vidal, F., & Zofío, J. L. (2017). Evaluating productive performance: A new approach based on the product-mix problem consistent with Data Envelopment Analysis. *Omega, 67*, 134–144.
Appice, A., & Džeroski, S. (2007). In *Stepwise induction of multi-target model trees* (pp. 502–509). Berlin, Heidelberg: Springer.
Aragon, Y., Daouia, A., & Thomas-Agnan, C. (2005). Nonparametric frontier estimation: A conditional quantile-based approach. *Econometric Theory, 21*(2), 358–389.
Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science, 30*(9), 1078–1092.
Barbosa, F., Rampazzo, P. C. B., Yamakami, A., & Camanho, A. S. (2019). The use of frontier techniques to identify efficient solutions for the Berth Allocation Problem solved with a hybrid evolutionary algorithm. *Computers & Operations Research, 107*, 43–60.
Borchani, H., Varando, G., Bielza, C., & Larrañaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5*(5), 216–233.
Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. Taylor & Francis.
Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.
Cazals, C., Florens, J. P., & Simar, L. (2002). Nonparametric frontier estimation: A robust approach. *Journal of Econometrics, 106*(1), 1–25.
Charles, V., Aparicio, J., & Zhu, J. (2019). The curse of dimensionality of decision-making units: A simple approach to increase the discriminatory power of data envelopment analysis. *European Journal of Operational Research, 279*(3), 929–940.
Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research, 2*(6), 429–444.
Daraio, C., & Simar, L. (2005). Introducing environmental variables in nonparametric frontier models: A probabilistic approach. *Journal of Productivity Analysis, 24*(1), 93–121.
De'Ath, G. (2002). Multivariate regression trees: A new technique for modeling species–environment relationships. *Ecology, 83*(4), 1105–1117.
Debreu, G. (1951). The coefficient of resource utilization. *Econometrica: Journal of the Econometric Society*, 273–292.
Deprins, D., & Simar, L. (1984). Measuring labor efficiency in post offices, The Performance of Public Enterprises: Concepts and Measurements, M. Marchand, P. Pestieau and H. Tulkens.
Douglas, P. C., & Cobb, C. W. (1928). A theory of production. *The American Economic Review, 18*(1), 139–165.
Du, P., Parmeter, C. F., & Racine, J. S. (2013). Nonparametric kernel regression with multiple predictors and multiple shape constraints. *Statistica Sinica*, 1347–1371.
Emrouznejad, A., & Anouze, A. L. (2010). Data envelopment analysis with classification and regression tree–a case of banking efficiency. *Expert Systems, 27*(4), 231–246.
Färe, R., & Primont, D. (1995). *Multiple-output production and duality: Theory and applications*. Boston: Kluwer Academic Publishers.
Farrell, M. J. (1957). The measurement of productive efficiency. *Journal of the Royal Statistical Society: Series A (General), 120*(3), 253–281.
Hall, P., & Huang, L. S. (2001). Nonparametric kernel regression subject to monotonicity constraints. *Annals of Statistics*, 624–647.
Henderson, D. J., & Parmeter, C. F. (2009). Imposing economic constraints in nonparametric regression: Survey, implementation, and extension. *Advances in Econometrics, 25*(2009), 433–469.
Kerstens, K., O'Donnell, C., & Van de Woestyne, I. (2019). Metatechnology frontier and convexity: A restatement. *European Journal of Operational Research, 275*(2), 780–792.
Khezrimotlagh, D., Zhu, J., Cook, W., & Toloo, M. (2019). Data envelopment analysis and big data. *European Journal of Operational Research, 274*(3), 1047–1054.
Koopmans TC (1951) Analysis of production as an efficient combination of activities. In: Koopmans TC (ed) Activity analysis of production and allocation. Cowles Commission for Research in Economics Monograph No. 13. Wiley, New York.
Kuosmanen, T., & Johnson, A. L. (2010). Data envelopment analysis as nonparametric least-squares regression. *Operations Research, 58*(1), 149–160.
Lee, C. Y., & Cai, J. Y. (2020). LASSO variable selection in data envelopment analysis with small datasets. *Omega, 91* 102019.
Levatić, J., Ceci, M., Kocev, D., & Džeroski, S. (2014). *Semi-supervised learning for multi-target regression. In International workshop on new frontiers in mining complex patterns* (pp. 3–18). Cham: Springer.
Li, F., Zhu, Q., & Liang, L. (2018). Allocating a fixed cost based on a DEA-game cross efficiency approach. *Expert Systems with Applications, 96*, 196–207.
Lozano, S., & Calzada-Infante, L. (2017). Dominance network analysis of economic efficiency. *Expert Systems with Applications, 82*, 53–66.
Morgan, J. N., & Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association, 58*(302), 415–434.
Parmeter, C. F., Sun, K., Henderson, D. J., & Kumbhakar, S. C. (2014). Regression and inference under smoothness restrictions. *Journal of Productivity Analysis, 41*, 111–129.
Perelman, S., & Santín, D. (2009). How to generate regularly behaved production data? A Monte Carlo experimentation on DEA scale efficiency measurement. *European Journal of Operational Research, 199*(1), 303–310.
Rebai, S., Yahia, F. B., & Essid, H. (2019). A graphically based machine learning approach to predict secondary schools' performance in Tunisia. *Socio-Economic Planning Sciences, 100724*.
Rokach, L. (2019). *Ensemble Learning: Pattern Classification Using Ensemble Methods*. World Scientific Publishing Co Pte Ltd..
Santin, D., & Sicilia, G. (2017). Dealing with endogeneity in data envelopment analysis applications. *Expert Systems with Applications, 68*, 173–184.
Shephard, R. W. (1953). *Cost and production functions*. Princeton University Press.
Stojanova, D., Ceci, M., Appice, A., & Džeroski, S. (2012). Network regression with predictive clustering trees. *Data Mining and Knowledge Discovery, 25*(2), 378–413.
Tavakoli, I. M., & Mostafaee, A. (2019). Free disposal hull efficiency scores of units with network structures. *European Journal of Operational Research, 277*(3), 1027–1036.
Vapnik, V. (2000). *The nature of statistical learning theory*. Springer science & business media.