Econometrics and Machine Learning

Arthur Charpentier*, Emmanuel Flachaire and Antoine Ly*****

Abstract – On the face of it, econometrics and machine learning share a common goal: to build a predictive model, for a variable of interest, using explanatory variables (or features). However, the two fields have developed in parallel, thus creating two different cultures. Econometrics set out to build probabilistic models designed to describe economic phenomena, while machine learning uses algorithms capable of learning from their mistakes, generally for classification purposes (sounds, images, etc.). Yet in recent years, learning models have been found to be more effective than traditional econometric methods (the price to pay being lower explanatory power) and are, above all, capable of handling much larger datasets. Given this, econometricians need to understand what the two cultures are, what differentiates them and, above all, what they have in common in order to draw on tools developed by the statistical learning community with a view to incorporating them into econometric models.

Codes JEL / JEL Classification : C18, C52, C55
Keywords: learning, Big Data, econometrics, modelling, least squares

Reminder:

The opinions and analyses in this article are those of the author(s) and do not necessarily reflect their institution's or Insee's views.

* *University of Rennes 1 & CREM (arthur.charpentier@univ-rennes1.fr)*
** *Aix-Marseille University, AMSE, CNRS & EHESS (emmanuel.flachaire@univ-amu.fr)*
*** *University of Paris-Est (antoine.ly.pro@gmail.com)*

Received on 2 September 2017, accepted after revision on 29 May 2018
Translated from the original version: "Économétrie et *Machine Learning*"

Pour citer cet article: Charpentier, A., Flachaire, E. & Ly, A. (2018). Econometrics and Machine Learning. *Economie et Statistique / Economics and Statistics*, 505-506, pp. 147–169. https://doi.org/10.24187/ecostat.2018.505d.1970

The earliest use of quantitative techniques in economics probably dates back to the sixteenth century (Morgan, 1990). However, it was not until the twentieth century that the term "econometrics" was first used, giving birth to the Econometric Society in 1933. Machine learning techniques are more recent. It was Arthur Samuel, widely regarded as the father of the first self-learning programme, who first coined the term "machine learning", which he defined as "a field of study that gives a computer the ability without being explicitly programmed" (Samuel, 1959). Among the earliest techniques are Hebb's cell assembly theory (Hebb, 1949) (which later gave birth to the "perceptron" in the 1950s, and then to neural networks), with Widrow and Hoff (1960) demonstrating, around fifteen years later, the links with least-squares methods, the SVM (support vector machine) and, more recently, boosting methods. While the two communities have developed in parallel, big data require links to be built between the two approaches by bridging the "two cultures" referred to by Breiman (2001a), contrasting mathematical statistics, which may be likened to traditional econometrics (Aldrich, 2010), with computational statistics and machine learning more generally.

Econometrics and supervised statistical learning techniques are similar, while also being very different. To start with, the two appear similar, with both using a database (or data table), i.e. observations $\{(y_i, x_i)\}$, with $i = 1, \cdots, n$, $x_i \in \mathcal{X} \subset \mathbb{R}^p$ and $y_i \in \mathcal{Y}$. If y_i is qualitative, we speak of a classification problem,[1] and, otherwise, of a regression problem. The two approaches also share common ground at the other end since, in both cases, the aim is to build a "model", i.e. a function $m : \mathcal{X} \mapsto \mathcal{Y}$ which will be interpreted as a prediction.

However, there are significant differences in between. Historically, econometric models have been based on economic theory, generally with parametric models. Traditional statistical inference methods (such as maximum likelihood and the method of moments) are thus used to estimate the values of a vector of parameters θ, in a parametric model $m_\theta(\cdot)$, by a value $\widehat{\theta}$. As in statistics, unbiased estimators are important since a lower bound on the variance can be obtained (Cramér-Rao bound). Asymptotic theory plays an important role (Taylor expansions, law of large numbers and central limit theorem). In statistical learning, by contrast, nonparametric models are often built based almost exclusively on data (i.e. no distribution hypothesis), and the meta-parameters used (tree-depth, penalty parameter, etc.) are optimised by cross-validation.[1]

Beyond the foundations, while the (often asymptotic) properties of $\widehat{\theta}$ (viewed as a random variable, thanks to the underlying stochastic representation) have been extensively studied in econometrics, statistical learning focuses to a greater extent on the properties of the optimal $m^\star(\cdot)$ based on a criterion that remains to be defined, or even simply $m^\star(x_i)$ for observations i deemed to be of interest for example in a test population. The problem of the choice of model is also viewed from a somewhat different perspective. Following Goodhart's law ("When a measure becomes a target, it ceases to be a good measure"), the goodness-of-fit of a model is penalised after the fact in econometrics by its complexity in the validation or selection phase, while in statistical learning it is the objective function which takes account of the penalty.

From High Dimension to Big Data

In this paper, a variable will be a vector of $\mathbb{R}^n$, such that by concatenating the variables, they can be stored in a matrix X, of size $n \times p$, with n and p being potentially large.[2] The fact that n is large is not a problem in itself. Many theorems in econometrics and statistics are obtained when $n \to \infty$. By contrast, the fact that p is large is problematic, particularly if $p > n$.

Portnoy (1988) showed that the maximum likelihood estimator retains the asymptotic normality property if p remains small in relation to n ($p^2 / n \to 0$ where $n, p \to \infty$). Indeed, it is not uncommon to speak of high dimension when $p > \sqrt{n}$. Another important concept is the idea of "sparsity", which is based not on the dimension p but on the actual dimension, in other words the number of truly

1. The term "classification" will be used when $\mathcal{Y}$ is a set of classes, typically a binary classification, $\mathcal{Y} = \{0,1\}$, corresponding to the outcome of an indicator variable. The term is less dated than "discrimination" and more general than the determination of a "score" (often an intermediate step). It should not be confused with unsupervised classification (such as "ascending hierarchical classification"), which involves the creation of homogeneous classes based on a similarity measure (in this case, the term "creation or construction of classes" or "clusters" is sometimes used).

2. Extensions are possible with MRI-type images as predictive variables, or climate data with maps as predictive variables. It is possible fall back on the typical case of data in the form of vectors by using the Tucker decomposition (Kolda & Bader, 2009).

important variables. It is thus possible to have $p > n$ while having convergent estimators.

The high dimension can be frightening because of the curse of dimensionality (Bellman, 1957). The volume of the unit sphere, in dimension p, tends towards 0 when $p \to \infty$. In such cases, the space is said to be "parsimonious" – i.e. the likelihood of finding a point close to another becomes increasingly small (we may even speak of a "sparse" space). While the idea of reducing the dimension by using a principal component analysis may seem attractive, the analysis suffers from a number of flaws in high dimension. The solution often revolves around the selection of variables (which raises the problem of multiple tests or computational time).

To use the terminology of Bühlmann & van de Geer (2011), the problems highlighted here correspond to those encountered in high dimension, an essentially statistical problem. From a computational perspective, we may go a little further, with truly Big Data. In the foregoing, the data were stored in a matrix X, of size $n \times p$. There can be issues with storing such a matrix or even with using a matrix widely used in econometrics, $X^T X$ ($n \times n$). The first-order condition of the linear model is associated with the solution to $X^T (X\beta - y) = 0$. In reasonable dimension, the Gram-Schmidt decomposition is used. In high dimension, the numerical descent and gradient methods may be used, where the gradient is approximated by subsampling (Zinkevich *et al.,* 2010). This computational dimension is often overlooked, despite the fact that it has been the basis of a significant number of methodological advances in econometrics.

Nonparametric and Computational Statistics

The purpose of this paper is to explain the major differences between econometrics and statistical learning, which correspond to the cultures alluded to by Breiman (2001a) in referring, in the context of statistical modelling, to the data modelling culture (based on a stochastic model, such as logistic regression or a Cox model) and the algorithmic modelling culture (based on the implementation of an algorithm, such as random forests or support vector machines; for a complete list, see Shalev-Shwartz & Ben-David, 2014). However, the boundary between the two is blurred. At the intersection, we find, for example, nonparametric econometrics, which is based on a probabilistic model (like econometrics) while focusing to a greater extent on algorithms and their performance rather than on asymptotic theorems.

Some Machine Learning Tools

Neural Networks

Neural networks are semiparametric models. Nevertheless, this family of models can be approached in the same way as nonparametric models: the structure of neural networks (presented below) can be modified to extend the class of functions used to approximate a variable of interest. More specifically, Cybenko (1989) showed that the set of neural functions is dense in the space of continuous functions on a compact space. In other words, we have a theoretical framework which guarantees a form of universal approximation. It also requires defining a neuron and emphasises the existence of a sufficient number of neurons to approximate any continuous function on a compact domain. Thus, a continuous phenomenon can be approximated by a sequence of neurons: this sequence is referred to as a "single-layer neural network". While the universal approximation theorem was demonstrated in 1989, the first functional artificial neuron was introduced by Franck Rosenblatt in the mid-twentieth century in Rosenblatt (1958). Referred to now as "basic neuron", this neuron is known as "Perceptron". In its earliest uses, it was used to determine the gender of an individual presented in a photo. It introduced the first mathematical representation of a biological neuron:

- The synapses transmitting the information to the cell are represented by a real vector. The dimension of the input vector of the neuron (which is none other than a function) corresponds biologically to the number of synaptic connections;

- Each signal transmitted by a synapse is then analysed by the cell. Mathematically, the schema is expressed by weighting the different components of the input vector;

- Depending on the information acquired, the neuron decides whether or not to resend a signal. The phenomenon is replicated by introducing an activation function. The output signal

is modelled by a real number computed as an image by the activation function of the weighted input vector.

Thus, an artificial neuron is a semiparametric model. The choice of activation function is left to the user. A basic neuron may then be formally defined by:

1. An input space $\mathcal{X}$, generally $\mathbb{R}^k$ with $k \in \mathbb{N}^*$;

2. An output space $\mathcal{Y}$, generally $\mathbb{R}$ or a finite set (typically $\{0,1\}$, although here we prefer $\{-1,+1\}$);

3. A vector of parameters $w \in \mathbb{R}^p$;

4. An activation function $\phi : \mathbb{R} \to \mathbb{R}$. Ideally, this function should be monotonic, derivable and bounded (here, "saturating") to ensure certain convergence properties.

This last function ϕ is reminiscent of logistic or probit transformations, which are popular in econometrics (which are cumulative distribution functions, of value in $[0,1]$, ideal when $\mathcal{Y}$ is the set $\{0,1\}$). For neural networks, preference is given to the hyperbolic tangent, the arctangent function or the sigmoid functions for classification problems on $\mathcal{Y} = \{-1,+1\}$ (the latter evoke the logistic transformation performed by econometricians). The term neuron is used to refer to any application f_w of $\mathcal{X}$ in $\mathcal{Y}$ defined by:

$$y = f_w(x) = \phi\left(w^T x\right), \quad \forall x \in \mathcal{X}$$

For the perceptron, introduced by Rosenblatt (1958), a basic neuron is assimilated to the function:

$$y = f_w(x) = signe\left(w^T x\right), \quad \forall x \in \mathcal{X}$$

According to this formalisation, many statistical models, such as logistic regressions, may be viewed as neurons. Any GLM (Generalised Linear Model) could be interpreted as an artificial neuron where the activation function ϕ is none other than the inverse of the canonical link function. If g denotes the link function of the GLM, w the vector of parameters, y the variable to be explained and x the vector of explanatory variables of the same dimension as w:

$$g(\mathbb{E}[Y \mid X = x]) = w^T x$$

We return to neural modelling by taking $\phi = g^{-1}$. However, the chief difference between GLMs and the neural model is that the latter requires no distribution hypothesis on $Y \mid X$ (here there is no need to introduce a probabilistic model). Furthermore, when the number of neurons per layer increases, convergence is not necessarily guaranteed if the activation function does not verify certain properties (which is not the case for the majority of the canonical link functions of GLMs). However, neural network theory imposes additional mathematical constraints on the function g (detailed in Cybenko, 1989). Thus, for example, a logistic regression may be viewed as a neuron, whereas generalised linear regressions do not verify all the necessary hypotheses.

To extend the analogy with the functioning of the nervous system, it is then possible to connect different neurons. We speak of a layered neural network structure. Each layer of neurons receives the same observation vector every time. To revert to a more econometric analogy, we might imagine an intermediate step, for example by not performing a regression on the raw variables x but a smaller set of orthogonal variables obtained based on a principal component analysis. Consider A as the matrix associated with this linear transformation, with A of size $k \times p$ if we wish to use the p first components. Take z as the transformation of x, where $z = A^T x$ ($z_j = A_j^T x$). One generalisation of the above model may be to posit:

$$y = f(x) = \phi\left(w^T z\right) = \phi\left(w^T A^T x\right), \quad \forall \mathrm{x} \in \mathcal{X}$$

where $w \in \mathbb{R}^p$. Here we have a linear transformation (by considering a principal component analysis), although we can imagine a generalisation with nonlinear transformations:

$$y = f(x) = \phi\left(w^T F_A(x)\right), \quad \forall x \in \mathcal{X}$$

where F is a function $\mathbb{R}^k \to \mathbb{R}^p$. It is the two-layer neural network. More generally, in order to formalise the construction, the following notations are introduced:

- $K \in \mathbb{N}^*$: number of layers;

- $\forall k \in \{1, \cdots K\}$, p_k represents the number of neurons in the layer k;

- $\forall k \in \{1, \cdots K\}$, W_k denotes the matrix of the parameters associated with the layer k. More specifically, W_k is a matrix $p_k \times p_{k-1}$ and for any $\in \{1, \cdots p_k\}$, $w_{k,l} \in \mathbb{R}^{p_{k-1}}$ denotes the weight

vector associated with the basic neuron l of the layer k;

• $W = \{W_1, .., W_K\}$ denotes the set of parameters associated with the neural network;

• $F^k_{W_k} : \mathbb{R}^{p_{k-1}} \to \mathbb{R}^{p_k}$ denotes the transfer function associated with the layer k. For the purpose of simplification, we may also write F^k;

• $\widehat{y}_k \in \mathbb{R}^{p_k}$ will represent the image vector of the layer $k \in \{1, \cdots, K\}$;

• $F = F_W = F^K \circ \cdots \circ F^1$ will denote the transfer function associated with the global network. In this respect, if $x \in \mathcal{X}$, we may note $\widehat{y} = F_W(x)$.

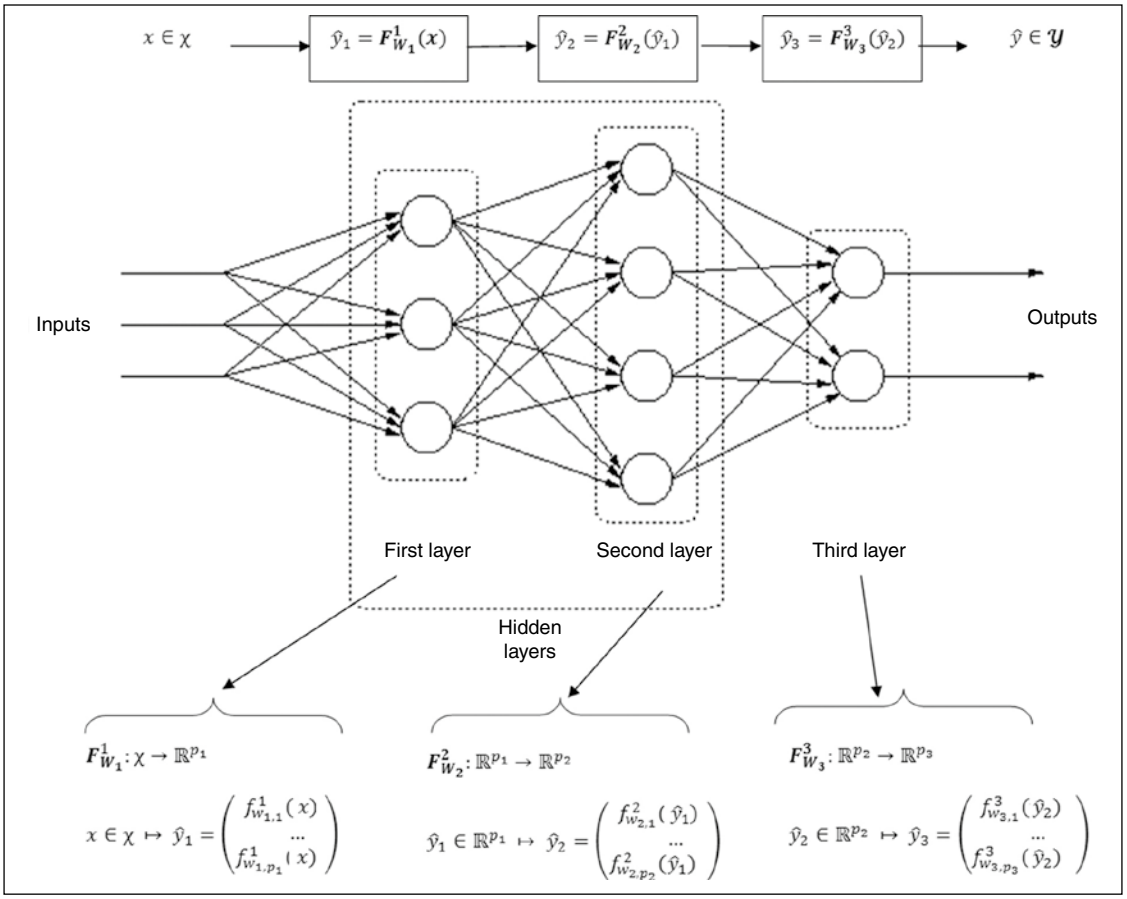
Diagram 1 provides an illustration of the notations presented here.[3] Each circle represents a basic neuron. Each rectangle encompassing several circles represents a layer. The first layer taking as "input" the observations $x \in \mathcal{X}$, is referred to as the input layer, while the output layer denotes the layer providing as "output" the prediction $\widehat{y} \in \mathcal{Y}$. The other layers are commonly known as hidden layers. A multilayer neural network is, therefore, a semiparametric model whose parameters are the set of components of the matrices W_k for any integer k of $\{1, \cdots, K\}$. Each activation function associated with each neuron (each circle of Diagram I) is to be determined by the user.

Once the model parameters to be calibrated have been identified (here, the reals forming the matrices W_k for each layer $k \in \{1, \cdots, K\}$), it is necessary to define a loss function ℓ. Indeed, it is worth recalling that the aim of supervised learning on a learning basis of $n \in \mathbb{N}^*$ couples $(y_i, x_i) \in \mathcal{Y} \times \mathcal{X}$ is to minimise the empirical risk (see Online complements – see the link at the end of the article):

$$\widehat{\mathcal{R}}_n(F_W) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(y_i, F_W(x_i)\right)$$

3. *See: http://intelligenceartificielle.org.*

Diagram 1
Example of Notations Associated with the Multilayer Neural Networks



To illustrate this point, let us consider the following example, which will also serve to illustrate the approach adopted. Let us assume that we are observing a phenomenon through observations $y_i \in [-1,1]$. The aim is to explain this phenomenon based on the independant variables x which are assumed to have actual values. The "universal approximation theorem" tells us that a single-layer neural network should enable the phenomenon to be modelled (subject to it being continuous). However, the theorem provides no indication of convergence speed. The user retains control of the choice of structure, which may be a simple neuron whose activation function is the hyperbolic tangent function:

$$y_1 = \tanh(w_0 + w_1 x)$$

where the parameters w_0, w_1 are to be optimised in order to minimise the empirical risk over the learning data.

Based on the universal approximation theorem, by adding several neurons, the error is expected to reduce. However, since the function to be estimated is not known, it can only be observed through the sample. Mechanically, learning-based error decreases when parameters are added. Error analysis by means of a test enables our ability to generalise to be assessed (Box 1).

A second model, which uses several neurons, may thus be considered. For example:

$$y_2 = w_a \tanh(w_0 + w_1 x) + w_b \tanh(w_2 + w_3 x) + w_c \tanh(w_4 + w_5 x)$$

where the parameters $w_0, .., w_5$ and w_a, w_b, w_c are the parameters to be optimised. Calibrating a neural network thus amounts to reiterating these structural modification steps until the risk is minimised on a test basis.

For a fixed neural network structure (i.e. fixed number of layers, number of neurons per layer and activation functions), the programme therefore amounts to determining the set of parameters $W^* = (W_1, ..., W_K)$ in such a way that:

$$W^* \in \underset{W=(W_1,...,W_K)}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \ell\left(y_i, F_W(x_i)\right) \right\}.$$

This formula underlines the importance of the choice of function ℓ. This loss function quantifies the average error of our model F_W based on learning. *A priori*, ℓ can be chosen arbitrarily. However, from the point of view of working out an optimisation programme, subdifferentiable and convex cost functions are preferable for guaranteeing the convergence of the optimisation algorithms. In addition to the quadratic loss function $\ell_2(y, \hat{y}) = (y - \hat{y})^2$, traditional loss functions include the hinge function $-\ell(y, \hat{y}) = max(0, 1 - y\hat{y})$ – and the logistic function $-\ell(y, \hat{y}) = \log(1 - e^{-y\hat{y}})$.

Neural networks were used very early on in economics and finance, notably on corporate defaults (Tam & Kiang, 1992; Altman *et al.,* 1994) and, more recently, credit rating (Blanco *et al.,* 2013; Khashman, 2011). However, structures such as those presented above are generally limited. Deep learning is more particularly characteristic of more complex neural networks (sometimes more than ten layers with hundreds of neurons per layer). Today, these

Box 1 – **Learning and Test Samples**

In the literature on learning, assessing the quality of a model based on the data used to build it says nothing about how the model will behave with new data. This is what is known as the "generalisation" problem. The traditional approach thus involves splitting the sample (of size n) in two: one part to train the model (the learning base, in-sample, of size m) and another to test it (the test base, out-of-sample, of size $n - m$). The latter allows for the measurement of a real predictive risk Let us suppose that the data are generated by a linear model $y_i = x_i^T \beta_0 + \varepsilon_i$ where the ε_i are realisations of independent centred distributions. The in-sample empirical quadratic risk is:

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{E}\left(\left[x_i^T \hat{\beta} - x_i^T \beta_0 \right]^2 \right) = \mathbb{E}\left(\left[x_i^T \hat{\beta} - x_i^T \beta_0 \right]^2 \right)$$

for any observation i. If the residuals ε are Gaussian, this risk equals $\sigma^2 p / m$, where p is the size of the vectors x_i. By contrast, the out-of-sample empirical quadratic risk is:

$$\mathbb{E}\left(\left[\mathrm{x}^{\mathrm{T}} \hat{\beta} - x^T \beta_0 \right]^2 \right)$$

Where x is a new observation, which is independent of the others. We may note that:

$$\mathbb{E}\left(\left[x^T \hat{\beta} - x^T \beta_0 \right]^2 \mid x \right) = \sigma^2 x^T (X^T X)^{-1} x$$

and by integrating in relation to x:

$$\mathbb{E}\left(\left[x^T \hat{\beta} - x^T \beta_0 \right]^2 \right) = \mathbb{E}\left(\mathbb{E}\left(\left[x^T \hat{\beta} - x^T \beta_0 \right]^2 \mid x \right) \right) = \sigma^2 \operatorname{trace}\left(\mathbb{E}[x x^T] \mathbb{E}[X^T X]^{-1} \right)$$

➔

Box 1 (contd.)

The expression is then different from that obtained in-sample, and by drawing on Groves & Rothenberg (1969), we can show that:

$$\mathbb{E}\left(\left[x^T\widehat{\beta}-x^T\beta_0\right]^2\right)\geq\sigma^2\frac{p}{m}$$

Except for certain simple cases, there is no simple formula. We may note, however, that if $x\sim\mathcal{N}\left(0,\sigma^2\mathbb{I}\right)$, then x^Tx follows a Wishart distribution, and:

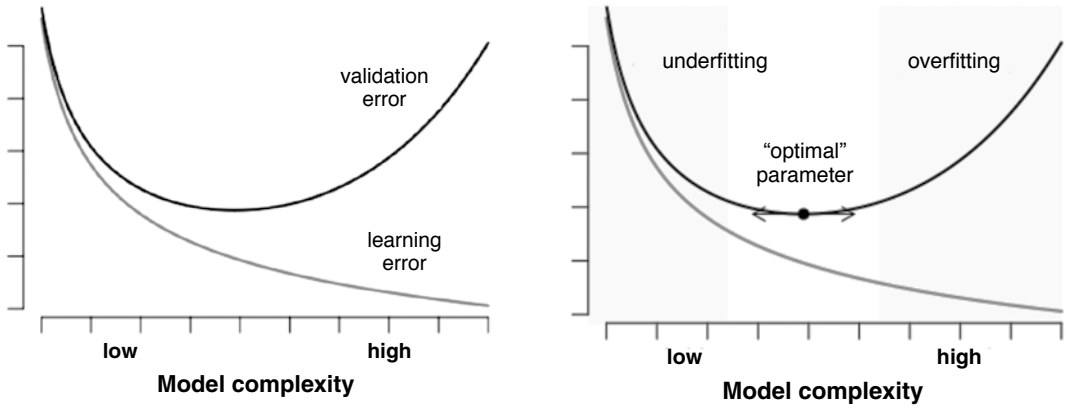
$$\mathbb{E}\left(\left[x^T\widehat{\beta}-x^T\beta_0\right]^2\right)=\sigma^2\frac{p}{m-p-1}$$

Let us now consider the empirical version: if $\widehat{\beta}$ is estimated on the m first observations,

$$\widehat{\mathcal{R}}^{\text{IS}}=\sum_{i=m+1}^{m}[y_i-x_i^T\widehat{\beta}]^2 \text{ and } \widehat{\mathcal{R}}^{\text{OS}}=\sum_{i=m+1}^{n}[y_i-x_i^T\widehat{\beta}]^2$$

and as noted by Leeb (2008), $\widehat{\mathcal{R}}^{\text{IS}}-\widehat{\mathcal{R}}^{\text{OS}}\approx 2\cdot\nu$ where ν represents the number of degrees of freedom. Figure A shows the respective evolution of $\widehat{\mathcal{R}}^{\text{IS}}$ and $\widehat{\mathcal{R}}^{\text{OS}}$ according to the complexity of the model (number of degrees in a polynomial regression, number of nodes in splines, etc.). $\widehat{\mathcal{R}}^{\text{IS}}$ always decreases with complexity (light curve). However, $\widehat{\mathcal{R}}^{\text{IS}}$ is non-monotonic (dark curve). If the model is too simple, it is a poor predictor, but if it is too complex, "over-learning" arises: it starts to model noise.

Figure A
Generalisation and Over-Learning



Reading note: The light curve represents the in-sample empirical risk on the learning sample, while the dark curve represents the out-of-sample risk on the test sample.

structures are very popular in signal analysis (image, text, sound) because they are capable, based on a very large quantity of observations, of extracting information which humans are incapable of perceiving and to deal with non-linear problems (LeCun *et al.,* 2015).

Information extraction can, for example, be performed through convolution. As an unsupervised procedure, it has produced excellent results in image analysis. In technical terms, this may be seen as a kernel-based transformation (as used in SVM techniques; see next section). While an image may be viewed as a matrix, with each coordinate representing a pixel, a convolution amounts to applying a transformation to a point (or area) of this matrix, thereby producing a new datum. The procedure can thus be repeated by applying different transformations (hence the notion of convolutional layers). The final vector obtained can then be fed into a neural model. More generally, a convolutional layer may be seen as a filter allowing the initial datum to be transformed.

One intuitive explanation for deep learning, and particularly deep networks, being so powerful for describing complex relationships in data is their construction around a simple functional approximation and the use of a form of hierarchy (Lin *et al.,* 2016). Nevertheless, deep learning models are more difficult to use since they require a significant degree of empirical judgement. While open-source libraries (Keras, Torch, etc.) currently allow more

readily for parallel computations by using, for example, GPUs (Graphical Processor Units), the user is still required to determine the structure of the most appropriate neural network.

Support Vector Machines

As noted above, in machine learning classification problems (as in signal processing), observations in the set $\{-1,+1\}$ are preferable (rather than $\{0,1\}$ in econometrics). With this notation, Cortes & Vapnik (1995) laid the theoretical foundations of support vector machine (SVM) models, an alternative to the then very popular neural networks. The initial idea of SVM methods involves finding a separating hyperplane dividing space into two sets of points as homogeneously as possible (i.e. containing identical labels). In dimension two, the algorithm involves determining a line separating the space into two areas that are as homogeneous as possible. Since it is a problem which may sometimes have an infinite number of solutions (there may be an infinity of lines separating the space into two distinct and homogeneous areas), an additional constraint is generally added: the separating hyperplane must be located as far as possible from the two homogeneous subsets which it generates (Diagram 2). In such cases, we speak of margin. The algorithm thus described is a soft- or hard-margin linear SVM.

If a plane can be entirely characterised by a directional vector w orthogonal to the latter and a constant b, applying an SVM algorithm to a set of $n \in \mathbb{N}^*$ points x_i of $\mathbb{R}^p$ labelled by $y_i \in \{-1,1\}$ thus amounts to solving a constrained optimisation programme similar to a lasso problem (quadratic deviation under linear constraint; see Online complements – link at the end of the article). In particular, we are led to solving the following:

$$\left(w^\star, b^\star\right) = \underset{w,b}{\operatorname{argmin}}\left\{\| w \|^2\right\} = \underset{w,b}{\operatorname{argmin}}\left\{w^T w\right\}$$

under constraints

$$\forall i \in \{1,\cdots,n\}, \begin{cases} \omega^T x_i + b \geq +1 \text{ when } y_i = +1 \\ \omega^T x_i + b \leq -1 \text{ when } y_i = -1 \end{cases}$$

The constraint can be loosened by allowing a point in a subset not to have the same label as the majority of the points in the subset provided it is not too far from the boundary. These are known as soft-margin linear SVMs. Heuristically, and indeed in practice, we cannot have $y_i\left(w^T x_i + b\right) - 1 \geq 0$ for any $i \in \{1,\cdots,n\}$; we loosen by introducing positive variables ξ such that:

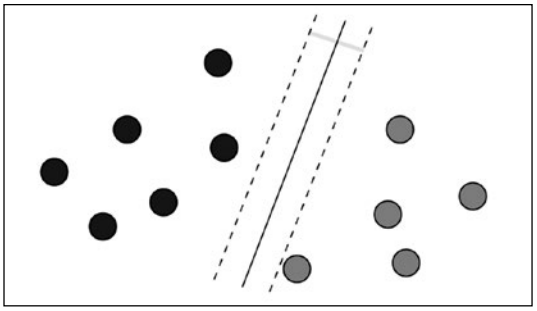
$$\begin{cases} \omega^T x_i + b \geq +1 - \xi_i \text{ lorsque } y_i = +1 \\ \acute{\text{E}}^{\text{T}} \text{x}_{\text{i}} + b \leq -1 + \xi_i \text{ lorsque } y_i = -1 \end{cases} \quad (1)$$

with $\xi_i \geq 0$. A misclassification occurs if $\xi_i > 1$, and a penalty is then applied as a price to pay for each error. The aim then is to solve a quadratic problem:

$$\min\left\{\frac{1}{2}\omega^T \omega + C 1^T 1_{\xi>1}\right\}$$

under constraint (1), which can be efficiently solved numerically by coordinate descent.

Diagram 2
Illustration of a Margin SVM



Sources: Vert (2017).

If the points cannot be separated, another possibility is to transfer them into a higher dimension in such a way that the data become linearly separable. Finding the right transformation separating the data is, however, very difficult. One mathematical trick for elegantly solving this problem involves defining the transformations $T(\cdot)$ and the scalar products using a kernel $K(x_1, x_2) = \langle T(x_1), T(x_2) \rangle$. One of the most common choices for a kernel function is the radial basis function (Gaussian kernel) $K(x_1, x_2) = \exp\left(-\|x_1 - x_2\|^2\right)$. However, no rules have so far been devised for choosing the "best" kernel. This technique is based on distance minimisation and does not predict the probability of being positive or negative, although a probabilistic interpretation is nonetheless possible (Grandvalet *et al.*, 2005).

Trees, Bagging and Random Forests

Classification trees were introduced by Breiman *et al.* (1984) and then by Quinlan (1986). We speak of CART, or Classification

and Regression Tree. The idea is to divide (based on the notion of branching) the input data consecutively until an allocation criterion (in relation to the target variable) is reached, based on a pre-defined rule.

The intuition: entropy $H(x)$ is associated with the amount of disorder in the data x in relation to the modalities of the classification variable y, and each partition aims to reduce this disorder. The probabilistic interpretation is to create groups that are as homogeneous as possible by reducing the variance of each group (intra-group variance), or in an equivalent manner by creating two groups that are as different as possible by increasing the variance between the groups (inter-group variance). At each stage, the partition providing the most significant reduction of disorder (or of variance) is chosen. The complete decision tree is developed by repeating this procedure across all the sub-groups, where each step results in a new partition into 2 branches, which subdivides the dataset into 2. Lastly, a decision about when to put an end to the creation of new branches is made by carrying out the final allocations (leaf nodes). There are several options. One option is to build a tree until all leaves are pure, i.e. composed of a single observation. Another option is to define a stopping rule linked to the size or decomposition of the leaves. Examples of stopping rules can be of minimum size (at least 5 elements per leaf) or minimum entropy. We speak of the pruning of the tree: the tree is allowed to grow, and then certain branches are cut *a posteriori* (which is different from introducing a stopping criterion *a priori* to the growth process of the tree – for example by imposing a minimum size on the leaves, or other criteria discussed in Breiman *et al.*, 1984).

At a given node, formed of n_0 observations (x_i, y_i) with $i \in \mathcal{I}_0$, we cut into two branches (one on the left and one on the right), thus partitioning $\mathcal{I}_0$ into $\mathcal{I}_g$ and $\mathcal{I}_d$. Let I be the criterion of interest, such as the entropy of the node:

$$I(y_0) = -n_0 p_0 \log p_0 \text{ where } p_0 = \frac{1}{n_0} \sum_{i \in \mathcal{I}_0} y_i$$

or the variance of the node:

$$I(y_0) = n_0 p_0 (1 - p_0) \text{ where } p_0 = \frac{1}{n_0} \sum_{i \in \mathcal{I}_0} y_i$$

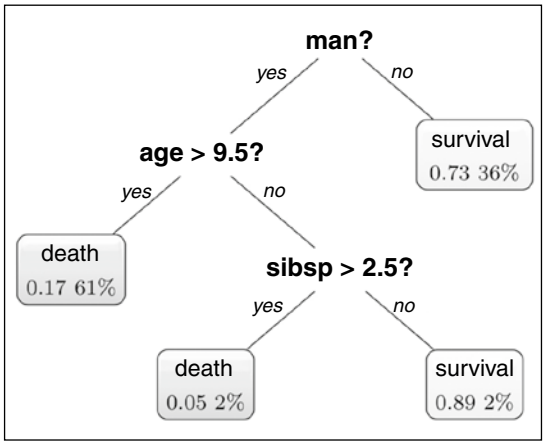
the latter also being the Gini impurity index.

The left and right branches are partitioned if the gain $I(y_0) - \left[I(y_g) + I(y_d) \right]$ is sufficiently significant. In the construction of the trees, the aim is to determine the partition that provides the greatest possible gain. This combinatorial problem being complex, Breiman *et al.* (1984) proposed a partition according to one of the variables, with $\mathcal{I}_g = \{i \in \mathcal{I}_0 : x_{k,i} < s\}$ and $\mathcal{I}_d = \{i \in \mathcal{I}_0 : x_{k,i} > s\}$, for a variable k and a threshold s (if the variable is continuous; otherwise, groupings of modalities are considered for qualitative variables).

The decision trees thus described are simple to obtain and easy to interpret (as shown by Diagram 3 on the data of the Titanic[4]), although they are not robust, and their predictive power is often very limited, particularly if the tree is very deep. One obvious idea is to develop a set of more or less independent tree models which, together, predict better than a single-tree model. The bootstrap method will be used, by sampling (with replacement) n observations among $\{(x_i, y_i)\}$. Each sample thus generated can be used to estimate a new classification tree, thus forming a forest of trees. It is the aggregation of all these trees that gives the prediction. The overall result is less sensitive than the initial sample and often gives better prediction results. These techniques, known as bagging (short for bootstrap aggregating), are similar to bootstrap techniques in regression (for example to construct confidence tubes in a functional regression).

4. This dataset, which contains information on all the passengers and crew members on the Titanic, with the variable indicating whether the person survived, has been widely used to illustrate classification techniques, see https://www.kaggle.com/c/titanic/data.

Diagram 3
Illustration of a Decision Tree Used to Predict the Survival Rate of a Passenger on the Titanic



Reading note: A woman (man: no) had a 73% chance of survival, with women representing 36% of the population.

Bagging involves generating random samples by sampling with replacement from the original sample, as with the bootstrap method. Random forests are based on the same principle as bagging, but during the construction of a classification tree, at each branch, a subset of m covariates is drawn randomly. In other words, each branch of a tree is not based on the same set of covariates. This helps to increase the variability between the trees and, ultimately, to obtain a forest composed of less correlated trees.

Choice of Classification Model

Given a model $m(\cdot)$ approximating $\mathbb{E}[Y \mid X = x]$, and a threshold $s \in [0,1]$, let us posit:

$$\hat{y}^{(s)} = \mathbb{1}[m(x) > s] = \begin{cases} 1 \text{ si } m(x) > s \\ 0 \text{ si } m(x) \leq s \end{cases}$$

The confusion matrix is then the contingency table associated with the countings $N = \left[N_{u,v}\right]$ with:

$$N_{u,v}^{(s)} = \sum_{i=1}^{n} \mathbb{1}\left(\hat{y}^{(s)} = u, y_j = v\right)$$

for $(u,v) \in \{0,1\}$. Table 1 presents such a matrix, with the name of each of the components: TP for true positives, corresponding to the 1 predicted in 1, TN for true negatives, corresponding to the 0 predicted in 0, FP for false positives, corresponding to 0 predicted in 1, and FN for false negatives, corresponding to 1 predicted in 0.

Several quantities are derived from this table. Sensitivity is the probability of predicting 1 in the population of 1, or the true positive rate. Specificity is the probability of predicting 0 in the population of 0 or the true negative rate. However, the true negative rate will be of greater interest, i.e. the probability of predicting 1 in the population of 0. The representation of these two values when s varies gives the ROC curve (receiver operating characteristic):

$$ROC_s = \left(\frac{FP_s}{FP_s + VN_s}, \frac{VP_s}{VP_s + FN_s}\right) = \left(sensitivity_s, 1 - specificity_s\right) \quad \text{pour } s \in [0,1]$$

This curve is presented in the next section, based on real data. The two values widely used in machine learning are the index κ, which compares observed and expected accuracy using a random model (Landis & Koch, 1977), and the AUC, corresponding to the area under the ROC curve. For the first index, once s is chosen, let $N^{\perp}$ be the contingency table corresponding to independent cases (defined based on N in the chi-square independence test. We then posit:

Table 1
Confusion Matrix, or Contingency Table for a Given Threshold *s*

	$y = 0$	$y = 1$	
$\hat{y}_s = 0$	VN_s	FN_s	$VN_s + FN_s$
$\hat{y}_s = 1$	FP_s	VP_s	$FP_s + VP_s$
	$VN_s + FP_s$	$FN_s + VP_s$	n

$$total\ precision = \frac{TP + TN}{n}$$

whereas:

$$random\ precision = \frac{[TN + FP] \cdot [TP + FN] + [TP + FP] \cdot [TN + FN]}{n^2}$$

We may then define:

$$\kappa = \frac{total\ precision - random\ precision}{1 - random\ precision}$$

Traditionally, s will be set at 0.5, as in naive Bayesian classification, although other values may be retained, in particular if the two errors are not symmetrical. There are compromises between simple and complex models measured by the number of parameters (or degrees of freedom more generally) in terms of performance and cost. Simple models are generally easier to compute, but can also lead to poorer goodness-of-fit (with high bias, for example). By contrast, complex models can provide a more accurate goodness-of-fit, but also risk being more costly in terms of computation. Furthermore, they go beyond the data or have greater variance and, just as with overly simple models, present significant test errors. As noted above, in machine learning, the optimal model complexity is determined using the bias-variance compromise.

From Classification to Regression

Historically, machine learning methods have focused on classification problems (with possibly more than 2 modalities[5]), with relatively little interest being shown in cases

5. *For example, in the case of letter or number recognition.*

where the variable of interest y is continuous. Nevertheless, a number of techniques can be adapted, such as trees and random forests, boosting and neural networks.

In the case of regression trees, Morgan & Sonquist (1963) proposed the AID method, based on the variance decomposition formula with an algorithm similar to the algorithm of the CART method described above. In a classification context, we would calculate, at each node (in the case of the Gini impurity index by adding on the left leaf $\{x_{k,i} < s\}$ and the right leaf $\{x_{k,i} > s\}$:

$$I = \sum_{i:x_{k,i}<s} \bar{y}_g\left(1-\bar{y}_g\right) + \sum_{i:x_{k,i}>s} \bar{y}_d\left(1-\bar{y}_d\right)$$

where $\bar{y}_g$ and $\bar{y}_d$ denote the frequencies of 1 in the left and right leaf, respectively. In the case of a regression tree, we use:

$$I = \sum_{i:x_{k,i}<s} (y_i - \bar{y}_g)^2 + \sum_{i:x_{k,i}>s} (y_i - \bar{y}_d)^2$$

corresponding to the (weighted) sum of intra-group variance. The optimal distribution is the distribution with the highest intra-group variance (the aim is for the leaves to be as homogeneous as possible).

In the context of random forests, a majority criterion is often used in classification (the predicted class is the majority class in a leaf), whereas for regression the predictions across all the trees are averaged. In a regression context (y continuous variable), the idea is to create a succession of models based on the boosting method (Box 2), which, in this case, takes the form:

$$m^{(k)}(x) = m^{(k-1)}(x) + \alpha_k \underset{h\in\mathcal{H}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - m^{(k-1)}(x) + h(x))^2 \right\}$$

where α_k is a shrinkage parameter and where the second term corresponds to a regression tree, on the residuals, $y_i - m^{(k-1)}(x_i)$. However, there are other techniques which allow for sequential learning. In an additive model (GAM), the aim is look for a notation in the form:

$$m(x) = \sum_{j=1}^{p} m_j\left(x_j\right) = m_1\left(x_1\right) + \cdots + m_p\left(x_p\right)$$

The idea of projection pursuit is based on a decomposition of the linear combinations and not of the explanatory variables. Let us consider a model:

$$m(x) = \sum_{j=1}^{k} g_j\left(\omega_j{}^T x\right) = g_1\left(\omega_1{}^T x\right) + \cdots + g_k\left(\omega_k{}^T x\right)$$

As with additive models, the functions $g_1, \cdots, g_k$ are to be estimated, as are the directions $\omega_1, \cdots, \omega_k$. This notation is relatively general and allows for interactions and cross effects to be considered (which is something that could not be done with additive models, which do not take into account nonlinearities). For example, in dimension 2, a multiplicative effect $m\left(x_1, x_2\right) = x_1 \cdot x_2$ is expressed as follows:

$$m\left(x_1, x_2\right) = x_1 \cdot x_2 = \frac{(x_1 + x_2)^2}{4} - \frac{(x_1 - x_2)^2}{4}$$

in other words $g_1(x) = x^2/4$, $g_2(x) = -x^2/4$, $\omega_1 = (1,1)^T$ and $\omega_2 = (1,-1)^T$. In the simple version, with $k = 1$, with a quadratic loss function, we may use a Taylor expansion to approximate $[y_i - g\left(\omega^T x_i\right)]^2$, and construct an

Box 2 – **Slow Learning by Boosting**

The idea of boosting, introduced by Shapire & Freund (2012), is to learn slowly from the errors of the model, in an iterative manner. In the first stage, a model m_1 is estimated for y, based on X, giving error ε_1. In the second stage, a model m_2 is estimated for ε_1, based on X, giving error ε_2, etc. After k iterations, the model is then selected:

$$\begin{aligned} m^{(k)}(\cdot) &= \underset{\sim y}{m_1(\cdot)} + \underset{\sim \varepsilon_1}{m_2(\cdot)} + \underset{\sim \varepsilon_2}{m_3(\cdot)} + \cdots + \underset{\sim \varepsilon_{k-1}}{m_k(\cdot)} \\ &= m^{(k-1)}(\cdot) + m_k(\cdot) \end{aligned} \quad (2)$$

Here, the error ε is seen as the difference between y and model $m(x)$, but it may also be seen as the gradient associated with the quadratic loss function. Equation (2) may be seen as a gradient descent, but expressed dualistically. The problem will then be recast as an optimisation problem:

$$m^{(k)} = m^{(k-1)} + \underset{h\in\mathcal{H}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \ell\left(y_i - m^{(k-1)}(x_i), h(x_i)\right) \right\} \quad (3)$$

where the space $\mathcal{H}$ is relatively simple (in such cases we speak of a weak learner). Traditionally, the functions $\mathcal{H}$ are staircase functions (found in classification and regression trees) known as stumps. To ensure that learning is slow, it is not uncommon for a shrinkage parameter to be used, and rather than positing, for example, $\varepsilon_1 = y - m_1(x)$, $\varepsilon_1 = y - \alpha \cdot m_1(x)$ is posited, with $\alpha \in [0,1]$.

iterative algorithm in the standard way. If we have an initial value ω_0, let us note that:

$$\sum_{i=1}^{n}[y_i - g\left(\omega^T x_i\right)]^2 \approx \sum_{i=1}^{n} g'(\omega_0{}^T x_i)^2 \left[\omega^T x_i + \frac{y_i - g\left(\omega_0{}^T x_i\right)}{g'\left(\omega_0{}^T x_i\right)} - \omega^T x_i\right]^2$$

corresponding to approximation in the generalised linear models on the function $g(\cdot)$ which was the link function (assumed to be known). We recognise a weighted least squares problem. The difficulty here is that the functions $g_j(\cdot)$ are unknown.

Applications

Big Data have required the development of estimation techniques capable of overcoming the limitations of parametric models, which are seen as too restrictive, and of traditional nonparametric models, whose estimation can be difficult in the presence of a large number of variables. Statistical learning, or machine learning, provides new nonparametric estimation methods, which perform well in a general context and in the presence of a large number of variables.[6] However, greater flexibility comes at the cost of a sometimes significant lack of interpretation.

In practice, one important issue is to determine the best model. The answer to this question depends on the underlying problem. If the relationship between the variables is approximated by a linear model, a correctly specified parametric model should perform well. By contrast, if the parametric model is not correctly specified, since the relationship is highly nonlinear and/or involves significant cross effects, then the statistical methods derived from machine learning should perform better.

The correct specification of a regression model is a common hypothesis, but one that is seldom verified and justified. In the following applications, we show how statistical methods derived from machine learning can be used to justify the correct specification of a parametric regression model or to detect a misspecification.

Sales of Child Car Seats (Classification)

Here, we will be drawing on an example used in James *et al.* (2013). The dataset contains the sales of child car seats at 400 stores (*sales*), as well as several variables, including the quality of the shelving location (*shelveloc*, equal to "poor", "average" and "good") and price (*price*).[7] A binary dependent variable is artificially created to describe high or low sales (*high* = "yes" if *sales* > 8 and "no" if not). In this application, the aim is to identify the determinants of a good volume of sales. We begin by considering a latent linear regression model:

$$y^\star = \gamma + x^T\beta + \varepsilon, \quad \varepsilon \sim G(0,1), \tag{4}$$

where x is composed of k explanatory variables, β is a vector of k unknown parameters and ε is an *i.i.d.* error term with a distribution function G with zero expectation and unit variance. The dependent variable y^* is not observed, with only y, with:

$$y = \begin{cases} 1 & \text{si } y^\star > \xi \\ 0 & \text{si } y^\star \leq \xi \end{cases} \tag{5}$$

The probability of y being equal to 1 may then be expressed as follows:

$$\mathbb{P}(Y=1) = G\left(\beta_0 + x^T\beta\right) \tag{6}$$

where $\beta_0 = \gamma - \xi$.[8] This model is estimated by maximum likelihood by selecting a parametric distribution G. If it is assumed that G is the normal distribution, it is a probit model; if it is assumed that G is the logistic distribution, it is a logit model. In a logit/probit model, there are two possible sources of misspecification:

- The linear relationship $\beta_0 + x^T\beta$ is misspecified;

- The parametric distribution used G is incorrect.

In the event of misspecification, of whatever kind, the estimation is no longer valid. The most flexible model is the following:

$$\mathbb{P}\left[Y=1|X=x\right] = G\left(h(x)\right) \tag{7}$$

where h is an unknown function and G an unknown distribution function. The bagging, random forest and boosting methods can be

6. *See, among others, Hastie* et al. *(2009) and James* et al. *(2013).*
7. *It is the Carseats dataset from the ISLR library.*
8. $\mathbb{P}[Y=1]=\mathbb{P}[Y^\star>\xi]=\mathbb{P}[\gamma+x^T\beta+\varepsilon>\xi]=\mathbb{P}[\varepsilon>\xi-\gamma-x^T\beta]$ *which can ultimately be written as* $\mathbb{P}[\varepsilon<\gamma-\xi+x^T\beta]$. *Given* $\gamma-\xi=\beta_0$, *we obtain* $\mathbb{P}[Y=1]=G(\beta_0+x^T\beta)$. *In general, it is assumed that the variance of the error term is equal to* σ^2, *in which case the parameters of model (6) are* β_0/σ *and* β/σ, *which means that the parameters of latent model (4) are not identifiable and are estimated to within one scale parameter.*

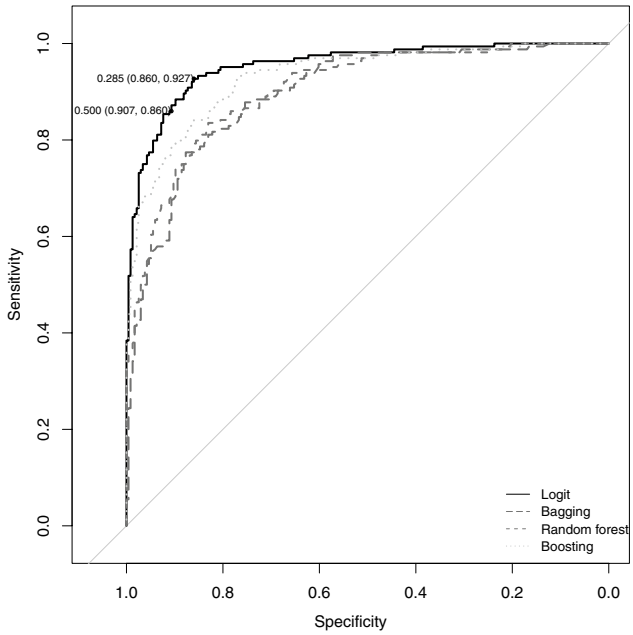
used to estimate this general model without making a preliminary choice about the function h and the distribution G. The estimation of the logit/probit model nevertheless performs better if h and G are correctly specified.

Model (6) is estimated using the logistic distribution for G, while model (7) is estimated with the bagging, random forest and boosting methods. A 10-fold cross-validation analysis is performed (Box 3). The individual probabilities of the out-of-sample data, i.e. of each of the folds not used for the estimation, are used to assess the quality of the classification.

Figure I shows the ROC curve and the area under the curve (AUC) for the logit, bagging, random forest and boosting estimations. The ROC curve is a graph that simultaneously represents the quality of the prediction in the two classes, for different values of the threshold used to classify the individuals (the term is "cutoff"). One obvious way of classifying individuals is to assign them to the class for which they have the highest estimated probability. In the case of a binary variable, this amounts to predicting the class for which the estimated probability is higher than 0.5. However, a different threshold could be used. For example, in Figure I, a point on the ROC curve of the logit model indicates that by using a threshold of 0.5, the correct prediction rate for the answer "no" is 90.7% (specificity), while the correct prediction rate for the answer "yes" is 86% (sensitivity). Another point indicates that by using 0.285, the correct prediction rate for the answer "no" is 86% (specificity), while the correct prediction rate for the answer "yes" is 92.7% (sensitivity). As described above, an ideal classification model would have an ROC curve of the form Γ. The best model is the model whose curve is above the others. One criterion commonly used to select the best model is the criterion with the largest area under the ROC curve (AUC). The advantage of such a criterion is that it is easy to compare and does not depend on the choice of classification threshold.

In our example, the ROC curve of the logit model is above the other curves and has the largest area under the curve (AUC = 0.9544). These results indicate that this model provides the best classification predictions. Since it no other model performs better, this finding suggests that the linear logit model is correctly specified and that there is no need to use a more general and more complex model.

Figure I
Sales of Car Seats: ROC Curves and Areas Under the Curve (AUC)



AUC	Logit	Bagging	Random Forest	Boosting
	0.9544	0.8973	0.9050	0.9313

Sources: Simulated data on 400 points of sale of baby car seats with the data set "Carseats" from James *et al.* (2013), https://CRAN.R-project.org/package=ISLR

Box 3 – **K-Fold Cross Validation**

Cross-validation is based on the idea of building an estimator by removing an observation. Since the aim is to build a predictive model, the prediction obtained from the estimated model will be compared with the missing observation:

$$\widehat{\mathcal{R}}^{\text{CV}} = \frac{1}{n}\sum_{i=1}^{n}\ell\left(y_i, \widehat{m}_{(i)}(x_i)\right)$$

The main problem of this method (known as the leave-one-out method) is that it requires calibrating n models, which can be problematic in high dimension. An alternative method is k-fold cross-validation, which involves using a partition of $\{1,\cdots,n\}$ in k groups (or folds) of the same size, $\mathcal{I}_1,\cdots,\mathcal{I}_k$ (let $\mathcal{I}_{\bar{j}} = \{1,\cdots,n\} \setminus \mathcal{I}_j$). With $\widehat{m}_{(j)}$ built on the sample $\mathcal{I}_{\bar{j}}$, we then posit:

$$\widehat{\mathcal{R}}^{k-\text{CV}} = \frac{1}{k}\sum_{j=1}^{k}\mathcal{R}_j \text{ or } \mathcal{R}_j = \frac{k}{n}\sum_{i\in\mathcal{I}_j}\ell\left(y_i, \widehat{m}_{(j)}(x_i)\right)$$

Using $k = 5,10$ presents two advantages compared to $k = n$ (corresponding to the leave-one-out method): (1) the number of estimations to be performed is far too low, i.e. 5 or 10 rather than n; (2) the samples used for the estimation are less similar and, therefore, less correlated with each other, which tends to avoid excessive variance (James *et al.,* 2013).

Purchase of Caravan Insurance (Classification)

Here, we will be drawing on an example used in James *et al.* (2013). The dataset contains 85 variables on the demographic characteristics of 5,822 individuals.[9] The dependent variable (*purchase*) indicates whether the individual has purchased caravan insurance; it is a binary variable, corresponding to "yes" or "no". In the dataset, only 6% of the individuals took out such insurance. The classes are therefore highly imbalanced.

Model (6) is estimated using the logistic distribution function, while model (7) is estimated by the bagging, random forest and boosting methods (the tuning parameters are those used by James *et al.* (2013), n.trees = 1,000 and shrinkage = 0.01). A 10-fold cross-validation analysis is performed. The individual probabilities of the out-of-sample data, i.e. of each of the pieces not used for the estimation, are used to assess the quality of the classification.

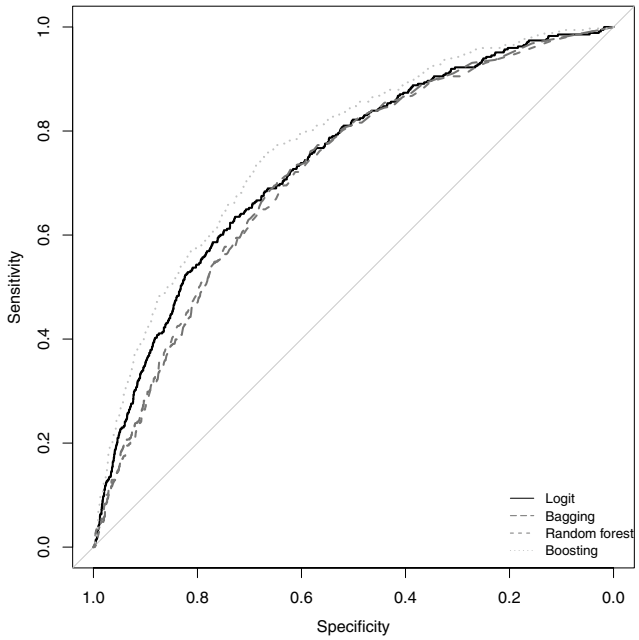
Figure II shows the ROC curve and the area under the curve (AUC) for the logit, bagging, random forest and boosting estimations. The curve of the boosting model is above the other curves and has the largest area under the curve (AUC = 0.7691). These results indicate that boosting provides the best classification predictions. Compared to the previous example, the curves are relatively far from the L shape, which suggests that the classification will not be as good.

It is important to consider the results of a standard classification, i.e. with a classification threshold of 0.5, which is often used by default in software (the prediction of the answer of individual i is "no" if the estimated probability of the individual answering "no" is higher than 0.5; if not, it is "yes"). The left side of Table 2 shows the correct classifications with this threshold (threshold of 0.5) for the different methods. With the best model and the standard threshold (boosting and threshold of 0.5), the "no" answers are 99.87% correct while the "yes" answers are all wrong. This equates to using a model which predicts that no one buys caravan insurance. For analysts, choosing such a model is absurd since their main focus is precisely the 6% of individuals who purchased such insurance. This result is explained by the presence of highly imbalanced classes. Indeed, by predicting that no one buys insurance, the error rate is "only" 6%. However, these are errors which result in not explaining anything.

Several methods can be used to overcome this problem, linked to highly imbalanced classes (Kuhn & Johnson, 2013, Chapter 16). One simple solution is to use a different classification threshold. The ROC curve presents the results according to several classification thresholds, where the perfect classification is illustrated by the couple (specificity, sensitivity) = (1,1), i.e. by the upper-left corner of the graph. The classification threshold corresponding to the point on the ROC curve closest to this corner is selected as the optimal classification threshold. The right side of Table 2 shows the correct classification rates with the optimal thresholds for the different methods (the optimal thresholds

9. *It is the Caravan dataset from the ISLR library under R.*

Figure II
Purchase of Insurance: ROC Curves and Areas Under the Curve (AUC)



AUC	Logit	Bagging	Random Forest	Boosting
	0.7372	0.7198	0.7154	0.7691

Sources: Experimental dataset "Caravan" on the consumption of caravan insurance, James *et al.* (2013). https://CRAN.R-project.org/package=ISLR

of the logit, bagging, random forest and boosting methods are 0.0655, 0.0365, 0.0395 and 0.0596, respectively). With boosting and an optimal threshold, the "no" answers are 68.6% correct, while the "yes" answers are 73.85% correct. The aim of the analysis being to correctly predict the individuals likely to buy caravan insurance ("yes" class) and to distinguish them sufficiently from the others ("no" class), the optimal threshold performs far better than the standard threshold (0.5). With a logit model and an optimal threshold, the correct classification rate for the "no" class is 72.78%, while the rate for the "yes" class is 63.51%. Compared to boosting, the logit model is slightly better at predicting the "no" class, but is significantly worse at predicting the "yes" class.

Personal Loan Defaults (Classification)

Consider the German database of personal loans, used in Nisbet *et al*. (2001) and Tufféry (2001), with 1,000 observations and 19 explanatory variables, including 12 qualitative variables: by disjuncting them (by creating an indicator variable for each modality), we obtain 48 potential explanatory variables. A recurring question in modelling is to determine which variables merit being used. The

Table 2
Purchase of Insurance: Sensitivity to the Choice of Classification Threshold

	Threshold of 0.5		Optimal Thresholds	
	Specificity	Sensitivity	Specificity	Sensitivity
Logit	0.9967	0.0057	0.7278	0.6351
Bagging	0.9779	0.0661	0.6443	0.7069
Random Forest	0.9892	0.0316	0.6345	0.6954
Boosting	0.9987	0.0000	0.6860	0.7385

Sources: Experimental dataset "Caravan" on the consumption of caravan insurance, James *et al.* (2013). https://CRAN.R-project.org/package=ISLR

most obvious solution for an econometrician may be a stepwise method (with running through all possible combinations of variables being, on the face of it, too complex in high, forward or backward dimension). The set of variables in a backward approach is shown in the first column of Table 3 (see Box 4 for the principles governing penalisation and the choice of explanatory variables). The table provides a comparison with two other approaches: first, the lasso method, by suitably penalising the norm ℓ_1 of the vector of parameters β (last column). We note that the first two variables considered as null (for a sufficiently large λ) are the first two to emerge from a backward procedure. One last method has been proposed by Breiman (2001b), using all of the trees created when building a random tree: the importance of the variable x_k in a forest of T trees is given by:

$$Importance(x_k) = \frac{1}{T}\sum_{t=1}^{n}\sum_{j\in N_{t,k}} p_t(j)\Delta\mathcal{I}(j)$$

where $N_{t,k}$ denotes the set of nodes of the tree t using the variable x_k as a separation variable, $p_t(j)$ denotes the proportion of observations in a node j, and $\Delta(j)$ is the index variation at the node j (between the preceding node, the left leaf and the right leaf). The central column of Table 3 shows the variables by decreasing order of importance when the index used is the Gini impurity index.

With the stepwise approach and the lasso method, we remain with linear logistic models. In the case of random forests (and trees), interactions between variables can be taken into account when 2 variables are present. For example, the variable *residence_since* ranks very high among the predictive variables (third most important variable).

Wage Determinants (Regression)

The Mincer wage equation (Mincer, 1974; Lemieux, 2006) has traditionally been used

Table 3
Credit: Choice of Variables, Sequential Sorting, Based on a Stepwise Approach, by Importance Function in a Random Forest and by Lasso

Stepwise	AIC	Random Forest	Gini	Lasso
checking_statusA14	1112.1730	checking_statusA14	30.818197	checking_statusA14
credit_amount(4e+03,Inf]	1090.3467	installment_rate	20.786313	credit_amount(4e+03,Inf]
credit_historyA34	1071.8062	residence_since	19.853029	credit_historyA34
installment_rate	1056.3428	duration(15,36]	11.377471	duration(36,Inf]
purposeA41	1044.1580	credit_historyA34	10.966407	credit_historyA31
savingsA65	1033.7521	credit_amount	10.964186	savingsA65
purposeA43	1023.4673	existing_credits	10.482961	housingA152
housingA152	1015.3619	other_payment_plansA143	10.469886	duration(15,36]
other_payment_plansA143	1008.8532	telephoneA192	10.217750	purposeA41
personal_statusA93	1001.6574	Age	10.071736	installment_rate
savingsA64	996.0108	savingsA65	9.547362	property_magnitudeA124
other_partiesA103	991.0377	checking_statusA12	9.502445	age(25,Inf]
checking_statusA13	985.9720	housingA152	8.757095	checking_statusA13
checking_statusA12	982.9530	jobA173	8.734460	purposeA43
employmentA74	980.2228	personal_statusA93	8.715932	other_partiesA103
age(25,Inf]	977.9145	property_magnitudeA123	8.634527	employmentA72
purposeA42	975.2365	personal_statusA92	8.438480	savingsA64
duration(15,36]	972.5094	purposeA43	8.362432	employmentA74
duration(36,Inf]	966.7004	employmentA73	8.225416	purposeA46
purposeA49	965.1470	employmentA75	8.089682	personal_statusA93
purposeA410	963.2713	duration(36,Inf]	8.029945	personal_statusA92
credit_historyA31	962.1370	purposeA42	8.025749	savingsA63
purposeA48	961.1567	property_magnitudeA122	7.908813	telephoneA192

Sources: Dataset "Credit" of the casdataset library of R, loans to households in Germany (Nisbet *et al.*, 2001; Tufféry, 2001). http://cas.uqam.ca/

Box 4 – **Penalisation and Methods for the Choice of Explanatory Variables**

To select relevant explanatory variables in econometrics, we may use criteria *ex post* relating to the quality of the model penalising the complexity, in practice the number of explanatory variables (such as R^2 adjusted or the Akaike criterion – AIC – see the online complement). In the forward method, we start with a regression on the constant before adding one variable at a time, retaining the variable that most improves the model according to the chosen criterion, until adding a variable reduces the quality of the model. In the backward method, we start with a regression on all the variables before adding one variable at a time, removing the variable that most improves the quality of the model, until removing a variable reduces the quality of the model. Stepwise methods introduce ensemble methods to limit the number of tests.

The machine learning strategy involves penalising *ex-ante* in the objective function, even at the risk of constructing a biased estimator. Typically, the following is built:

$$\left(\hat{\beta}_{0,\lambda}, \hat{\beta}_{\lambda}\right) = \operatorname{argmin}\left\{\sum_{i=1}^{n} \ell\left(y_i, \beta_0 + x^T\beta\right) + \lambda\, pénalisation(\beta)\right\} \quad (8)$$

where the penalisation function will often be a norm $\|\cdot\|$ chosen *a priori*, and a penalisation parameter λ.

to explain (individual) wages according to the individual's education, experience and gender:

$$\log(\text{wage}) = \beta_0 + \beta_1 ed + \beta_2 exp + \beta_3 exp^2 + \beta_4 fe + \varepsilon \quad (9)$$

where *ed* is the level of education, *exp* is the level of professional experience and *fe* is a dummy variable equal to 1 if the individual is a woman. According to human capital theory, the expected wage increases with experience, at an increasingly slow rate, until it reaches a threshold before decreasing. The introduction of the square of *exp* enables such a relationship to be taken into account. The presence of variable *fe* allows for any wage gap between men and women to be measured.

Model (9) establishes a linear relationship between wage and level of education and a quadratic relationship between wage and professional experience. These relationships may seem too restrictive. Several studies have shown, in particular, that wages do not fall after a certain age and that a quadratic relationship or a higher-degree polynomial is more appropriate (Murphy & Welch, 1990; Bazen & Charni, 2017).

Model (9) also establishes that the wage gap between men and women is independent of the level of education and experience. It is too restrictive if, for example, the average wage gap between men and women is low for unskilled jobs and high for skilled jobs, or low among early-career workers and high among late career workers (interaction effects). The most flexible model is the fully nonparametric model:

$$\log(wage) = m(ed, exp, fe) + \varepsilon \quad (10)$$

where $m(\cdot)$ is a random function. It has the advantage of being able to take into account any nonlinear relationships and complex interactions between the variables. However, its significant flexibility is at the cost of a more difficult interpretation of the model. Indeed, a 4-dimensional graph would be needed to represent the function m. One solution is to represent the function m in 3 dimensions by fixing the value of one of the variables, although the represented function may differ significantly with a different fixed value.

We will use data from a survey by the US Census Bureau carried out in May 1985 drawn from Berndt (1990) and available under R.[10] The two models are estimated and a 10-fold cross-validation analysis is used to select the best approach. Parametric model (9) is estimated by ordinary least squares (OLS). Fully nonparametric model (10) is estimated by the method of splines since it includes few variables and also by the bagging, random forest and boosting methods.

The results of the 10-fold cross-validation are presented in Table 4. The best model is the model that minimises the criterion $\widehat{\mathcal{R}}^{10-CV}$. The results show that model (9) is at least as effective as model (10), which suggests that parametric model (9) is correctly specified.

Determinants of House Prices in Boston (Regression)

Here, we will be drawing on one of the examples used in James *et al.* (2013), whose data

10. *It is the CPS1985 dataset from the AER library.*

are available under R. The dataset contains the median values of house prices (*medv*) in $n = 506$ neighbourhoods around Boston along with 13 other variables, including the average number of rooms per house (*rm*), the average age of houses (*age*) and the percentage of households with a low economic status (*lstat*).[11]

Consider the following linear regression model:

$$medv = \alpha + x^T \beta + \varepsilon \tag{11}$$

where x = [chas,nox,age,tax,indus,rad,dis,lstat, crim,black,rm,zn,ptratio] is a vector in dimension 13 and β is a vector of 13 parameters. This model specifies a linear relationship between the value of houses and each of the explanatory variables. The most flexible model is the fully nonparametric model:

$$medv = m(x) + \varepsilon . \tag{12}$$

The estimation of this model by the Kernel method or the method of splines can be problematic since the number of variables is relatively high (there are 13 variables here) or, at least, too high to consider estimating a surface in dimension 13. We estimate the two models and use a 10-fold cross-validation analysis to select the best approach. Parametric model (11) is estimated by ordinary least squares (OLS) and fully nonparametric model (12) is estimated using three different methods: bagging, random forest and boosting (here we use the default values used in James *et al.*, 2013, pp. 328–331).

Table 5 shows the results of the 10-fold cross-validation. Based on the in-sample results (on the learning data), the bagging and random forest methods are found to be vastly more effective than the OLS estimation of linear model (11), the criterion $\widehat{\mathcal{R}}^{10-CV}$ going from 21.782 to 1.867 and 1.849. The out-of-sample results (on data other than those used to estimate the model) tend in the same direction, although the difference is less significant, with the criterion $\widehat{\mathcal{R}}^{10-CV}$ going from 24.082 to 9.59 and 9.407. These results illustrate a common phenomenon with nonlinear methods such as bagging and random forest, which can be highly effective in predicting the data used in the estimation, but less effective at predicting out-of-sample data. This explains why the selection of the best estimation is typically based on an out-of-sample analysis.

The difference between the estimation of models (11) and (12) is significant. Such a difference suggests that the linear model is misspecified and that nonlinear relationships

11. It is the Boston dataset from the MASS library. For a complete description of the data, see: https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/Boston.html.

Table 4
Wage: Fold Cross Validation Analysis (*K* = 10): Performance of the Estimation of Linear Model (9) and Fully Nonparametric Model (10)

$\widehat{\mathcal{R}}^{10-CV}$	Model (9)	Model (10)			
	OLS	Splines	Bagging	Random forests	Boosting
Out-of-sample	0.2006	0.2004	0.2762	0.2160	0.2173

Source: Population census, USA, 1985, Berndt (1990). Dataset CPS1985 from AER Library. https://rdrr.io/cran/AER/man/CPS1985.html

Table 5
House Prices in Boston - Fold Cross Validation Analysis (*K* = 10): Performance of the Estimation of Linear Model (11) and Fully Nonparametric Model (12)

$\widehat{\mathcal{R}}^{10-CV}$	Model (11)	Model (12)		
	OLS	Splines	Random forests	Boosting
In-sample	21.782	1.867	1.849	7.012
Out-of-sample	24.082	9.590	9.407	11.789

Coverage: Districts of the Boston metropolitan area.
Sources: James *et al.* (2013), Boston data set from the MASS library. https://stat.ethz.ch/R-manual/Rdevel/library/MASS/html/Boston.html

and/or interaction effects are present in the relationship between house prices and the explanatory variables x. This requires looking for a better parametric specification. Based on parametric model (11), and in order to take into account any potential nonlinearities, the following generalised additive model (GAM) may be considered:

$$medv = m_1(x_1) + m_2(x_2) + \ldots + m_{13}(x_{13}) + \varepsilon \quad (13)$$

where $m_1, m_2, \ldots m_{13}$ are unknown functions. The advantage of this model is that it allows for any nonlinear relationship between the dependent variable and each of the explanatory variables to be considered. Furthermore, it does not suffer from the curse of dimensionality problem since each of the functions is of dimension 1 and it is easily interpretable. However, it does not take into account any potential interaction effects. The estimation of generalised additive model (13) by the method of splines, as part of a 10-fold cross-validation analysis, gives value $\widehat{\mathcal{R}}^{10-CV} = 13.643$. Compared to parametric model (11), there is a significant gain (13.643 vs. 24.082). However, the difference with the fully nonparametric model (12) remains substantial (13.643 vs 9.590, 9.407, 11.789). Such a difference suggests that taking into account individual relationships which may be highly nonlinear is not sufficient and that interaction effects between variables are present. The simplest interaction variables among the pairs of variables $(x_i \times x_j)$ could be included in the model, but that would imply adding a significant number of variables to the original model (78 in this case), which would have an impact on the quality of the estimation of the model. In any case, as things stand, what can be said is that the linear model is misspecified and that there are potentially significant interaction effects in the relationship between *medv* and X, the identification of such effects remaining a delicate matter.

To go further, the tools developed in statistical learning may be of great use. For example, the random forest estimation technique involves measures of the significance of each of the variables in the estimation of the model. Table 6 shows these measurements in relation to model (12), estimated on the whole sample. The results suggest that the variables *rm* and *lstat* are the most significant variables to explain house price variations *medv*. This finding suggests enriching the initial relationship by adding the interaction effects linked to these two variables only, which are the most significant.

The generalised additive model including the interaction variables is estimated on the whole sample:

$$medv = m_1(x_1) + \cdots + m_{13}(x_{13}) + (rm : x)\gamma + (lstat : x)\delta + \varepsilon \quad (14)$$

where $(rm : x)$ represents the interaction variables of *rm* with all the other variables of x and $(lstat : x)$ represents the lstat interaction variables with all the other variables of x.[12] Analysis of the results of this estimation suggests that functions $\widehat{m}_i$ are linear for all variables except for the DIS variable, whose estimated relationship is shown in Figure III. This variable measures the average distance from five employment centres within the region. The effect appears to decrease more rapidly with distance when the latter is not very significant. Beyond a certain distance (beyond 2, in log), the effect is reduced and continues to decrease, albeit at a slower rate. This nonlinear relationship can be approximated by a piecewise linear regression by considering a node.

12. *We have (rm:x)=[rm×chas, rm×nox, rm×age, rm×tax, rm×indus, rm×rad, rm×dis, rm×lstat, rm×crim, rm×black, rm×zn, rm×ptratio] and (lstat:x)=[lstat×chas, lstat×nox, lstat×age, lstat×tax, lstat×indus, lstat×rad, lstat×dis, lstat×crim, lstat×black, lstat×zn, lstat×ptratio].*

Table 6
House Prices: Measures of the Importance of Each Variable in the Random Forest Estimation of Model (12), by Considering the Whole Sample

	% IncMSE	IncNodePurity
rm	61.35	18 345.41
istat	36.20	15 618.22
dis	29.37	2601.72
nox	24.91	1034.71
age	17.86	554.50
ptratio	17.43	626.58
tax	16.60	611.37
crim	16.26	1701.73
indus	9.45	237.35
black	8.72	457.58
rad	4.53	166.72
zn	3.10	35.73
chas	0.87	39.05

Coverage: Districts of the Boston metropolitan area.
Sources: James *et al.* (2013), Boston data set from the MASS library. https://stat.ethz.ch/R-manual/Rdevel/library/MASS/html/Boston.html

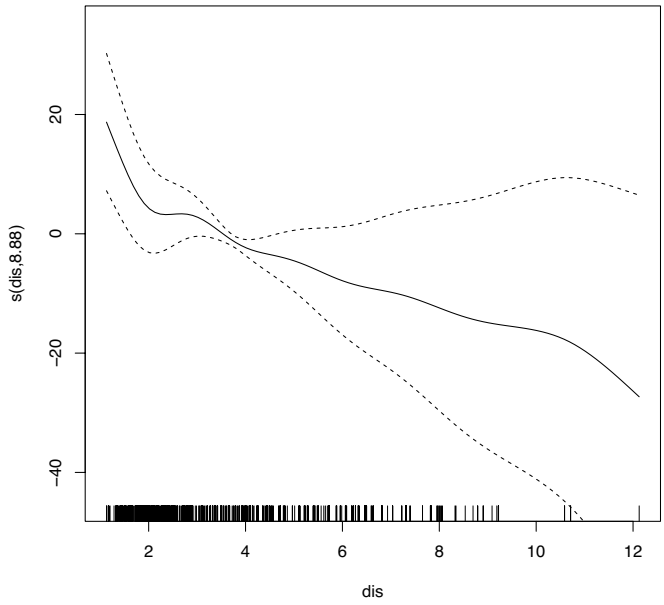
Lastly, the above analysis suggests considering the following linear model:

$$medv = \alpha + x^T\beta + (dis - 2) + \theta + (rm : x)\gamma + (lstat : x)\delta + \varepsilon \quad (15)$$

where $(dis - 2)$ is equal to the value of its argument if the latter is positive, and to 0 if it is not. Compared to the original linear model, this model includes a piecewise linear relationship with the DIS variable, as well as interaction effects between *rm*, *lstat* and each of the other variables of *x*.

Table 7 shows the results of the 10-fold cross validation of the estimation of parametric models (11) and (15), estimated by ordinary least squares (OLS), and of the generalised additive model (14) estimated by splines. It shows that the addition of interaction variables and of the piecewise linear relationship in model (15) produces far better results than the initial model (11): the criterion $\widehat{\mathcal{R}}^{10-CV}$ is divided by more than two, going from 24.082 to 11.759. By comparing these results with the results of Table 5, we also find that parametric model (15), estimated by OLS, is as effective as general model (12) estimated by boosting ($\widehat{\mathcal{R}}^{10-CV} = 11.789$). The difference with the bagging and random forest methods is not very significant ($\widehat{\mathcal{R}}^{10-CV} = 9.59, 9.407$). Lastly, the bagging, random forest and boosting methods served to highlight the misspecification of the original parametric model and then to find a far more effective parametric model by taking into

Figure III
Estimation of the Relationship $m_7(x_7)$ in the Generalised Additive Model (14), where x_7 = dis.



Note: Estimation of the m7 (x7) relationship for the dis variable in the generalized additive model; the dotted lines correspond to the 95% confidence intervals.
Coverage: Districts of the Boston metropolitan area.
Sources: James *et al.* (2013), Boston data set from the MASS library. https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/Boston.html

Table 7
House Prices in Boston - Fold Cross Validation Analysis (*K* = 10): Performance of the Estimation of Linear Models (11) and (15) and of Model (14) Including the Interaction Effects and With a Piecewise Nonlinearity

$\widehat{\mathcal{R}}^{10-CV}$	Model (11)	Model (14)	Model (15)
	OLS	Splines	OLS
Out-of-sample	24.082	13.643	11.759

Coverage: Districts of the Boston metropolitan area.
Sources: James *et al.* (2013), Boston data set from the MASS library. https://stat.ethz.ch/R-manual/Rdevel/library/MASS/html/Boston.html

account the effects of appropriate nonlinearities and interactions.

* *
*

While the two cultures (or two communities) of econometrics and machine learning have developed in parallel, the number of links between the two is constantly increasing. Whereas Varian (2014) outlined the significant contributions of econometrics to the machine learning community, our aim here was to present concepts and tools developed over time by that very community and which may be of use to econometricians, in a context of ever increasing data volumes. The probabilistic foundations of econometrics are without doubt its key asset, allowing not only for model interpretability, but also for the quantification of uncertainty. Nevertheless, the predictive performance of machine learning models is of value insofar as they allow for the identification of a misspecified econometric model. In the same way that nonparametric techniques provide a point of reference for assessing the relevance of a parametric model, machine learning tools help to improve an econometric model by detecting a nonlinear effect or an overlooked cross effect.

An illustration of the potential interactions between the two communities can be found, for example, in Belloni *et al*. (2010, 2012), in the context of the choice of instrument in a regression. Using the data produced by Angrist & Krueger (1991) relating to an academic achievement problem, they show how to effectively implement instrumental econometric techniques when 1,530 instruments are available (a recurring problem with the increase in the volume of data). As we have seen throughout this paper, although the approaches adopted may differ fundamentally in the two communities, econometricians have much to gain from using many of the tools developed by the machine learning community. □

Link to the Online complements: https://www.insee.fr/en/statistiques/fichier/3706234?sommaire=3706269/505-506_Charpentier-Flachaire-Ly_complement_EN.pdf

BIBLIOGRAPHY

Aldrich, J. (2010). The Econometricians' Statisticians, 1895-1945. *History of Political Economy*, 42(1), 111–154.
https://doi.org/10.1215/00182702-2009-064

Altman, E., Marco, G. & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). *Journal of Banking & Finance*, 18(3), 505–529.
https://doi.org/10.1016/0378-4266(94)90007-8

Angrist, J. D. & Krueger, A. B. (1991). Does Compulsory School Attendance Affect Schooling and Earnings? *The Quarterly Journal of Economics*, 106(4**)**, 979–1014.
https://doi.org/10.2307/2937954

Bazen, S. & Charni, K. (2017). Do earnings really decline for older workers? *International Journal of Manpower*, 38(1), 4–24.
https://doi.org/10.1108/IJM-02-2016-0043

Bellman, R. E. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Belloni, A., Chernozhukov, V. & Hansen, C. (2010). Inference for High-Dimensional Sparse Econometric Models. *Advances in Economics and Econometrics*, *10th World Congress of Econometric Society*, 245–295
https://doi.org/10.1017/CBO9781139060035.008

Belloni, A., Chen, D., Chernozhukov, V. & Hansen, C. (2012). Sparse Models and Methods for Optimal Instruments With an Application to Eminent Domain. *Econometrica*, 80(6), 2369–2429.
https://doi.org/10.3982/ECTA9626

Blanco, A. Pino-Mejias, M., Lara, J. & Rayo, S. (2013). Credit scoring models for the microfinance industry using neural networks: Evidence from Peru. *Expert Systems with Applications*, 40(1), 356–364.
https://doi.org/10.1016/j.eswa.2012.07.051

Breiman, L. Fiedman, J., Olshen, R. A. & Stone, C. J. (1984). *Classification And Regression Trees*. Chapman & Hall/CRC Press Online. https://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf

Breiman, L. (2001a). Statistical Modeling: The Two Cultures. *Statistical Science*, 16(3), 199–231. https://doi.org/10.1214/ss/1009213726

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

Bühlmann, P. & van de Geer, S. (2011). *Statistics for High Dimensional Data: Methods, Theory and Applications*. Berlin: Springer Verlag. https://doi.org/10.1007/978-3-642-20192-9

Cortes, C. & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3), 273–297. https://doi.org/10.1023/A:1022627411411

Grandvalet, Y., Mariéthoz, J., & Bengio, S. (2005). Interpretation of SVMs with an Application to Unbalanced Classification. *Advances in Neural Information Processing Systems* N° 18. https://papers.nips.cc/paper/2763-a-probabilistic-interpretation-of-svms-with-an-application-to-unbalanced-classification.pdf

Groves, T. & Rothenberg, T. (1969). A note on the expected value of an inverse matrix. *Biometrika*, 56(3), 690–691. https://doi.org/10.1093/biomet/56.3.690

Hastie, T., Tibshirani, R. & Friedman, J. (2009). *The Elements of Statistical Learning*. New York: Springer Verlag. https://doi.org/10.1007/978-0-387-84858-7

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley. https://doi.org/10.1002/1097-4679(195007)6:3<307::AID-JCLP2270060338>3.0.CO;2-K

James, G., D. Witten, T. Hastie, & Tibshirani, R. (2013). An Introduction to Statistical Learning. *Springer Texts in Statistics* 103. https://doi.org/10.1007/978-1-4614-7138-7

Khashman, A. (2011). Credit risk evaluation using neural networks: Emotional versus conventional models. *Applied Soft Computing*, 11(8), 5477–5484. https://doi.org/10.1016/j.asoc.2011.05.011

Kolda, T. G. & Bader, B. W. (2009). Tensor Decompositions and Applications. *SIAM Review*, 51(3), 455–500. https://doi.org/10.1137/07070111X

Kuhn, M. & Johnson, K. (2013). *Applied Predictive Modeling*. New York: Springer Verlag. https://doi.org/10.1007/978-1-4614-6849-3

Landis, J. R. & Koch, G.G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1), 159–174. https://doi.org/10.2307/2529310

LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, **521,** 436–444. https://doi.org/10.1038/nature14539

Leeb, H. (2008). Evaluation and selection of models for out-of-sample prediction when the sample size is small relative to the complexity of the data-generating process. *Bernoulli*, 14(3), 661–690. https://doi.org/10.3150/08-BEJ127

Lemieux, T. (2006). The "Mincer Equation" Thirty Years After Schooling, Experience, and Earnings. In: Grossbard, S. (Ed.), *Jacob Mincer: A Pioneer of Modern Labor Economics*, pp. 127–145. Boston, MA: Springer Verlag. https://doi.org/10.1007/0-387-29175-X_11

Lin, H. W., Tegmark, M. & Rolnick, D. (2016). Why does deep and cheap learning work so well? https://arxiv.org/abs/1608.08225

Mincer, J. (1974). Schooling, Experience and Earnings. New York: NBER. https://www.nber.org/books/minc74-1

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. https://doi.org/10.1080/01621459.1963.10500855

Morgan, M. S. (1990). *The history of econometric ideas*. Cambridge, UK: Cambridge University Press.

Murphy, K. M. & Welch, F. (1990). Empirical Age-Earnings Profiles. *Journal of Labor Economics*, 8(2), 202–229. https://doi.org/10.1086/298220

Nisbet, R., Elder, J. & Miner, G. (2011). *Handbook of Statistical Analysis and Data Mining Applications*. New York: Academic Press.

Portnoy, S. (1988). Asymptotic Behavior of Likelihood Methods for Exponential Families when the Number of Parameters Tends to Infinity. *Annals of Statistics*, 16(1), 356–366. https://doi.org/10.1214/aos/1176350710

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1), 81–106. https://doi.org/10.1007/BF00116251

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408.
https://doi.org/10.1037/h0042519

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229.
https://doi.org/10.1147/rd.33.0210

Shalev-Shwartz, S. & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, UK: Cambridge University Press.

Shapire, R. E. & Freund, Y. (2012). *Boosting*. *Fondations and Algorithms*. Cambridge, A MIT Press.

Tam, K. Y. & Kiang, M. Y. (1992). Managerial applications of neural networks: The case of bank failure predictions. *Management Science*, 38, 926–947.
https://doi.org/10.1287/mnsc.38.7.926

Tufféry, S. (2001). *Data Mining and Statistics for Decision Making*. Hoboken, NJ: Wiley.

Varian, H. R. (2014). Big Data: New Tricks for Econometrics. *Journal of Economic Perspectives*, 28(2), 3–28.
https://doi.org/10.1257/jep.28.2.3

Vert, J. P. (2017). Machine learning in computational biology. Cours à l'Ensae ParisTech.
http://members.cbio.mines-paristech.fr/~jvert/teaching/

Widrow, B. & Hoff, M. E. Jr. (1960). Adaptive Switching Circuits. *IRE WESCON Convention Record*, 4, 96–104.
https://apps.dtic.mil/dtic/tr/fulltext/u2/241531.pdf

Zinkevich M. A., Weimer, M., Smola, A. & Li, L. (2010). Parallelized Stochastic Gradient Descent. *Advances in neural information processing systems* 23, 2595–2603.
https://papers.nips.cc/paper/4006-parallelized-stochastic-gradient-descent.pdf