Data Science: A Primer for Economists

Gomez-Ruano, Gerardo

2020

DATA SCIENCE
A PRIMER FOR ECONOMISTS

GERARDO GOMEZ-RUANO

Abstract. The last years have seen an explosion in the demand for data science skills. In this paper, I introduce the reader to the term, point out the technological jumps that allowed the rise of its methods, and give an overview of the most common ones. I close by pointing out the strengths and weaknesses of the corresponding tools as well as their complementarities with economic analysis.

1. Data Science: what is it?

The term *data science* was coined by professor Chikio Hayashi in 1996 in the *Proceedings of the Fifth Conference of the International Federation of Classification Societies* [1]; this volume was edited by Hayashi himself together with four more STATISTICIANS. In it, Hayashi's article "What is Data Science? Fundamental Concepts and a Heuristic Example" reads:

> *Data Science is not only a synthetic concept to unify statistics, data analysis and their related methods but also comprises its results. It includes three phases, design for data, collection of data, and analysis on data.*

It should come as no surprise that Hayashi proposed a more appealing umbrella-term for the growing work that was being done by many researchers from a variety of backgrounds.

Today data science is more associated with businesses than with academia. So a more up-to-date definition of data science would be:

> *Data Science is the set of steps necessary–from the design of data collection to the preparation of analytical and visual content–for the provision of actionable insights to stakeholders.*

The use of the word "Science" may seem as an abuse of language, since this word has been reserved for studies that follow the Scientific Method, and not for quantitative

 Email: ggomezr@bu.edu .

methods used in science. Why not use "Applied Statistics" or something similar? Probably because it is far more appealing. For good or for bad, the overwhelming growth of quantitative methods in an age of exponentially-growing computational and information-gathering possibilities, has established *data science* as a reference term for a set of skills and tasks.

Many things have changed in the last twenty years since Hayashi's definition, and different terms have appeared now and then. Some of these terms are: *Data Mining*, *Data Analytics*, *Big Data*, *Machine Learning*, *Deep Learning*, and even *Artificial Intelligence*. In a fast-paced race to offer new, profitable technologies there is no limit for innovative, ambitious terms. However *data science* remains the umbrella term, due to its enormous generality.

2. The rise of Data Science in discontinuous jumps

Massive computational resources and amounts of data are not new at all. Super-computers have been a staple of academic research ever since the appearance of computers themselves. Not many people are aware that the conception of a Universal Computer, as opposed to a simple calculator or electronic device, was a theoretical construct meant to solve a mathematical problem (Turing Machine). By the same token, the use of massive, artificially generated or observationally gathered data is not new at all either. Meteorology, Quantum Physics, Astronomy and other disciplines / sciences have always provided plenty of data for the existing computational resources. University super-computers have always had a waiting line.

What then, happened in the last twenty years? Two discontinuities.

2.1. **Discontinuity 1: the Internet.**

Although AMAZON–arguably one of the first and most successful online companies–was founded in 1994, the internet exploded fully thanks to the appearance of GOOGLE in 1998, an incredibly powerful–and still the most popular–search engine. It is not by chance that the so-called dot-com[1] bubble began just after Google's appearance: The Nasdaq Composite Index increased threefold in two years: from August 1998 (1,499.25 points) to February 2000 (4,696.69 points). This is an annual growth of about 77%.[2] only to fall to its lowest point in October 2002 (1,172.06 points). The Internet, the network comprised of all the connected computers around the world, is ultimately responsible for the supply of an incredible amount of information for almost free.

[1]This name is due to the ".com" ending of website names for commercial entities.

[2]Amazon's share price increased from $13 to $68 during the same period. GOOGLE became public until 2004.

The following is a perfectly timed quote from Hal Varian, an Economics scholar known for his Textbook "Intermediate Microeconomics", who later became Google's Chief Economist:

> *The ability to take data—to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it—that's going to be a huge skill in the next decades... Because now we do really have essentially free and ubiquitous data.*

This quote is from 2008, twelve years after Hayashi's definition; and these words were from no one less than Google's Chief Economist.

The internet was thus a necessary condition for the widespread popularity of Data Science.

2.2. **Discontinuity 2: Big Scale Parallelism.** Every computer has a main brain, which is an electronic chip. This chip is known as the CPU or Central Processing Unit.

For a very long time (about forty years), an important part of the improvement in computational power came from increasing more and more the speed (measured in frequency) of the CPU. Old CPUs, like Intel's popular *386* launched in 1985, ran at frequencies in the Mega Hertz (MHz) territory whereas CPUs in the year 2020 run at frequencies in the Giga Hertz (GHz) territory.[3]

With the appearance of the internet, it became clear that users wanted and needed to perform multiple tasks at the same time: browsing the internet, listening to music, writing a document, watching a video, and more. Multi-tasking used to be done–and still is done when necessary–by splitting the processor's time over all tasks. But this solution was not good enough for the post-internet era.

In 2006, Intel introduced CPUs with multiple *cores* (multiple computation units in the same CPU). Multiple cores allowed the simultaneous processing of *many independent tasks*. As a result, two different applications like a spreadsheet and a word processor could run simultaneously and without compromising the stability of the other. Nevertheless, the time that processing *a single long task* takes is not reduced by simply introducing multiple cores; new algorithms and programming principles are needed to spread the computation of a single long task across different cores.

[3]The pace at which the CPU's performance increased is oftentimes called "Moore's Law". This "law", enunciated in 1965 by Intel's co-founder, Gordon Moore, states–in gross terms–that the CPU's computational power doubles every one and a half years. This "law" served as a coordination device whereby the whole computer industry could plan in advance despite the rapidly changing conditions.

Such "refactoring" of code can be very hard and expensive; it is called *parallel programming*.

To give an example of parallel programming, suppose you need to compress many files. One option is to have a single core process every file sequentially, the other option is to have multiple cores process one file each simultaneously; the latter will clearly finish faster. This is a crude illustration of parallel computing, of course.

Despite efforts by big players–like Google–to popularize parallel computing, the cost of rewriting code often outweighs the benefits for developers. Even today (year 2020), with multiple cores being ubiquitous, one can find many computer applications that take a long time because they only use a single core. At the time of writing, one of the most computationally intensive built-in applications of MS Windows, the file compression utility WinZip, still runs on a single core.[4]

Few niche applications have taken advantage of parallel programming, naturally those that are most computationally intensive. Linear algebra is perfectly suited for parallelization since matrix multiplications are literally multiple operations that can be processed independently and simultaneously. So numerical and statistical packages like Matlab and Stata developed versions especially suited for multiple cores. Still, for the enormous amount of information that big companies and researchers deal with, parallel processing with a handful of cores is sometimes not enough. More computation units are necessary for those cases. So big-scale parallelism had to come from two other places: cloud computing and graphics cards.

Cloud Computing allows the use of immense computing power as a service. As previously said, it used to be that one had to choose between long waiting lines or acquiring prohibitively-expensive high-performance computers. But the rise of computing power as a service allows the rental of many computers (a cluster of computers) at the same time to perform intensive computations (big data and complex algorithms). Something that, of course, is only possible thanks to the internet. In the words of the pioneer and most important cloud computing provider at the time of writing, Amazon Web Services:

> *Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider.*

[4]Other file compression applications like 7zip do take advantage of multiple cores, and perform the same task considerably faster.

Amazon's share prices increased seventy-fold in twelve years: from the beginning of its cloud computing unit in July 2006 ($26) to August 2018 ($2012). That is an annual growth of about 43% . As of the time of writing, more than half of Amazon's profits come from its cloud computing unit (shocking for a company that is best known for selling goods online). Other important players like Microsoft and Google have followed Amazon's footsteps and are now serious competitors in the market for cloud computing.

Graphics Cards are–accidentally–also responsible for big-scale parallelism. Graphics cards have been available since the late 80's for computer aided design (CAD) and playing three-dimensional games in computers. Both activities (arguably more the latter) are very computationally intensive. Because every ray of light is independent of each the other, most computations can be done independently and with little if any *branching* (these are *if* statements and similar). They are typically *affine transformations* that can be performed incredibly fast and at very high precision in these graphics cards.

Since all the hardware was readily available, GPUs (GPU stands for Graphics Processing Unit, as opposed to CPU) became THE disruptive force for the most parallel-intensive method of machine learning: *neural networks* (later versions called *deep neural networks*, *deep learning*, or *artificial intelligence*). These methods require the solution of a big number of small problems, just like neurons in our brains are simple yet far many units. The main player was the company NVIDIA, whose stock increased more than tenfold in three years: from August 2015 (about $21.47) to January 2018 (about $233). This is an annual growth of about 120% .

3. The Methods and Tools that comprise Data Science

The methods that constitute data science now are not too different from those of twenty years ago. But the aforementioned factors have made their use more popular.

Choice and method for data collection, collection of data itself, and analysis of data remain all at the core of data science.

The majority of the methods–as before–come from Statistics.

3.1. **Machine Learning.**

While data science is usually identified with the whole process from designing information collection to the provision of actionable insights to stakeholders," the mathematical analysis itself is commonly called *machine learning*. Machine learning is divided into so-called *supervised* and *unsupervised learning*. The former aims to predict a defined target, while the latter aims to find unknown relationships.

Type of Learning		Method/Model	Description
Supervised	Classification	Decision Trees	classification based on a step-wise data-dependent rule
		Naive Bayes	classification based on simple probability
		Nearest Neighbour	classification based on distance from representative values
		Support Vector Machines	classification based on separating hyperplane(s)
		Discriminant Analysis	classification based on normal distributions for each category
		Logistic Regression	see section 4
		Neural Networks	see section 5
	Regression	Linear	see section 4
		Generalized Linear	see section 4
		Non-parametric	see section 4
Unsupervised	Clustering	K-Means	based on mean values of sets with cardinality K
		Hierarchical	based on binary tree hierarchies
	Matrix Factorization	Principal Components	decomposing the covariance matrix into its eigen-vectors
		Factor Analysis	like Principal Components but assuming a constant ratio between the rows of the correlation matrix

TABLE 1. Selection of popular machine learning methods/models

The obtention of parameters is called *training* and the comparison with data is called *testing*. For people with applied quantitative backgrounds, "learning" and "training" may seem as another abuse of language because the machines are not learning or training: the researcher is simply using a computer as a tool to obtain the parameters that best fit the model. Unlike econometric models, where the interest often lies in the value that the parameters take and the theoretical relationships,

Tool	Description
Unix-based Shell	command-line tools for manipulation of files and text
Python	open-source programming language for usage in interactive mode
Jupyter Notebooks	lightweight, Matlab-like interface for interacting with Python
Numpy Pandas Scikit-learn Matplotlib	open-source complementary Python libraries for machine learning and visualization
R Matlab Stata	powerful, stand-alone applications for mathematics/statistics
SQL	popular language specification for database management; MySQL is the most popular implementation
PyTorch Keras Tensorflow	open-source Python libraries for High Performance Computing (HPC) and Neural Networks (NNs)
Spark Hadoop	open-source software for distributed, big data analysis

TABLE 2. Popular machine learning tools as of 2020

Machine learning is typically uninterested in the parameter values. Another key difference from econometrics is that the theory for evaluating the robustness of the parameters is completely absent: the concept of statistical significance is rarely used and the model is usually "trained" with a subsample and then "tested" against the rest of the sample. The models may be procedural; that is, they may involve multiple steps as in *decision trees*.

Table 1 shows a non-exhaustive list of the most popular models/methods in machine learning with a brief description, and table 2 lists some of the currently popular tools (software) used in machine learning.

4. Regression

Regression methods in machine learning deserve a section on their own because they are a subset of econometrics and also the most important methods for predicting continuous variables.

The use of regression in machine learning is very primitive with respect to econometrics. Regression is exclusively cross-sectional, and the analysis of the chosen model and the fitted parameters is practically absent. Models like the Tobit are missing. Resulting models are compared mainly in terms of goodness of fit. To some, this may make sense given the abundance of information and the potential amount of regressors. But the same pitfalls that plague econometrics, plague machine learning as well.

Over-adjustment is a very common problem. The ad-hoc solution used in Machine Learning is testing the model against some previously separated set of the data. Instead of relying on significance and stability of regressors, the user of machine learning methods relies on a mixture of trial-and-error and model-comparisons to achieve his goal.

Other pitfalls include spurious correlations and logical fallacies. But we will say more on this in section 6.

5. Neural Networks

The holly grail of data science / machine learning is neural networks. The astonishing power of neural networks finally crystalized thanks to the use of massive hardware. Different variations of neural networks have appeared and–understandably–they differentiate themselves from their original version by calling themselves *deep* neural networks (DNN) and *deep* learning. This method has brought with it what is arguably the most important technological advancement in recent years: Pattern Recognition. Pattern Recognition is so important for human beings, that it has been–unfortunately–called artificial intelligence.[5] Among other things, DNN allow:

- Speech recognition
- Facial recognition
- Text/Character recognition
- Visual recognition of arbitrary objects (vehicles, pedestrians, traffic lights, driving lanes, groceries, etc.)

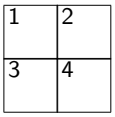
[5]As a consequence, true artificial intelligence now has to be called Artificial *General* Intelligence

- voice command recognition

The amount and extent of the possibilities derived from Pattern Recognition is appalling. With the help of multiple cameras, this technology allows recognition and classification of practically anything. Available information and computational power are the only constraints.

DNNs have shown us just how much Pattern Recognition makes us human. It is thanks to Pattern Recognition that we are able to group objects (semantics), allowing us to abstract from certain qualities and focus on others that we consider fundamental for belonging to a group: what makes a chair? a table? a car? a truck? a notebook? a skyscraper? a house? a street? a pedestrian? a hat? We can simply label the pictures of these objects and "train" the machine to identify the presence of these objects (do not be fooled, the objects themselves are not identified, only the presence of the pattern! See below) Thus, physical objects are perfectly amenable to recognition.

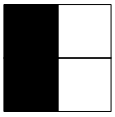
Neural networks are so important that they deserve a concrete–if only toy–example. Consider a very simple black-and-white image with four pixels:

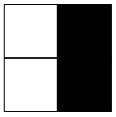


Let p_i denote the ith pixel, and let

$$p_i = \begin{cases} 1 & \text{if the } i\text{th pixel is black,} \\ 0 & \text{otherwise.} \end{cases}$$

Suppose we wish the machine to identify when the image displays columns of width one. In other words, we want the machine to be able to tell the following two cases from the rest:





We then label these two cases as "column" and the rest as "other". This is a very simple example, not just because of the dimensions, but because we can perfectly label the cases, and all the possible cases are available for "training".

For a machine, the fact that these four pixels are organized into a square is superfluous. What matters is that we give these four pixels some order, which we already

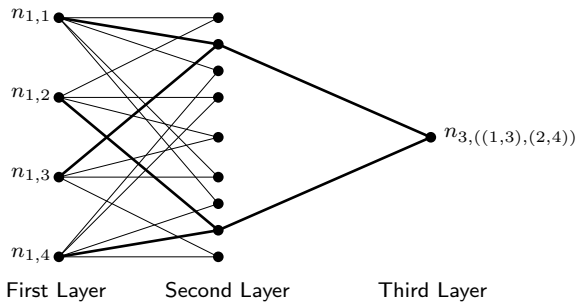


FIGURE 1. Example Neural Network

have by enumerating them. So the machine can be thought of as only seeing sequences of four *bits*: p_1, p_2, p_3, p_4. And out of the $2^4 = 16$ possible binary sequences, only two represent a column of width one, namely $1, 0, 1, 0$ and $0, 1, 0, 1$.

We can consider this machine as one that has four "optical" neurons that become "excited" when they see a black square, and send an electrical signal to the other neurons connected to them. Let us use the notation $n_{1,i}$ instead of p_i to emphasize this conceptual change and the fact that these will form the "first" layer of neurons.

Behind the first layer of four neurons, there may be plenty more neurons which may have arbitrarily many connections. However we will restrict the structure of these by assigning them to layers and assuming that layers are directly connected only to adjacent layers.

Let us assume that there is a second layer of neurons with one neuron $n_{2,(i,j)}$ for every pair $(n_{1,i}, n_{1,j})$ with $i \neq j$, and that both $n_{1,i}$ and $n_{1,j}$ are connected to it. There are 12 such neurons since there are $4 \cdot 3$ such pairs (see figure 1). Suppose that these neurons $n_{2,(i,j)}$ with $i \neq j$ become excited if and only if both $n_{1,i}$ and $n_{1,j}$ are excited and send them an electrical signal. Mathematically, we can define the electrical input of neuron $n_{2,(i,j)}$ by the linear combination $n_{1,i} + n_{1,j}$. And then we can assume that neuron $n_{2,(i,j)}$ gets excited if and only if $n_{1,i} + n_{1,j} > 1$, in which case $n_{2,(i,j)}$ takes the value one, while otherwise it takes the value zero.

Let there be a third layer of neurons, which is analogous to the second. That is, there is a neuron $n_{3,(k,l)}$ for every pair $(n_{2,k}, n_{2,l})$, where $k \neq l$ and $k, l = (i, j)$ for some $i \neq j$. At this stage we are done because there is one and only one neuron $n_{3,(k,l)}$ in the third layer that gets excited ($n_{2,k} + n_{2,l} > 1$) when there is a column of width one in the image, namely neuron $n_{3,((1,3),(2,4))}$. It is unnecessary to show the other neurons of the third layer in figure 1.

This toy example illustrates how neural networks can be used to recognize patterns of any sort; all that is needed is to translate the perception into zeros and ones. It follows immediately that sound, video, and their stereo-scopical versions can be used since they are easily captured into files or streams of zeros and ones. Once the right parameters are found, they can be hard-wired into chips for fast and lightweight use.

The same example above also illustrates how pattern recognition is not the same as object recognition: the machine is never aware of the fact that a chair "has legs" because it does not even know what is meant by legs of a chair or even by chair itself. All the machine does is recognize if something that looks like a chair is in the image. Sure, one could endlessly train the machine, but the fundamental difference (as of now) is that one needs millions of images to train a machine to recognize a "three-legged chair", while it only takes a sentence to "train" a human being into recognizing one such. Having said this, it is still evident that no other class of learning has the power of neural networks/a.k.a. deep learning/a.k.a. artificial intelligence.

6. Limits of Data Science and complementarities with Economics

So far we have seen how data science has the potential to translate data into insights. The reader is probably already aware of the most obvious limits of data science: there is no underlying theory about the behaviour of the data and there seems to be a lack of rigour in the method of research. Everything works, but precisely because of this one has that much of what "works" can be of little use. In the end, domain knowledge is not just recommended, but necessary. Before diving into the data, it is important to have a logical structure for it.

The complementarities with Economics are also obvious whenever the subject of analysis is of economic nature. In fact, it is not unusual that companies start hiring "data scientists" only to find out that what they really need is economists with a solid quantitative background. Amazon and Facebook are two technological firms that fit this description.

Many economists lack the "hacking" expertise that computer scientists or self-taught programmers have. They are dispensable in situations where the object of study is not of economic nature and expertise in econometrics is useless. Image recognition is a perfect example. By the same token, whenever the gathering of data involves processing of raw `html` and other computer code, people with experience in this kind of data-gathering techniques are crucial for success.

In the great majority of cases though, economists with a solid quantitative background and computer skills are likely to be of interest for firms that demand analysis of data for business decisions.

References

[1] Hayashi Chikio. 1998. What is Data Science? Fundamental Concepts and a Heuristic Example. In: Hayashi Chikio, Yajima K., Bock HH., Ohsumi N., Tanaka Y., Baba Y. (eds) Data Science, Classification, and Related Methods. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Tokyo

[2] Manyika James. 2008. Interview with Hal Varian. Transcript in
McKinsey & Company. 2009. Hal Varian on how the Web challenges managers. `https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/hal-varian-on-how-the-web-challenges-managers` (accessed September 14, 2020)