Preface

Table of Contents

A textbook for managers

The target reader for this book is a manager who needs to clearly understand the business stakes of "data science", "big data", "APIs" and "artificial intelligence" so that they can:

- **leverage** these technologies to improve the efficiency of their existing business,
- **innovate** with new products and services

The promise of this book is to bring you from a starting point with no knowledge of these technical concepts, to a point where you understand the concepts **and** you can develop "data centric" business projects: when "data" contributes to creating value for customers and all stakeholders.

Is this too technical or too easy for me?

This book does not assume any pre-requisite. It uses simple terms and a progressive learning curve to lead you to a comprehensive understanding of the topics.

If you are unsure, try this simple test: http://bit.ly/essentials-1-test

→ There are 20 topics. See how you score. If the score is below 12, this introductory volume is for you.

1. Multiple layers of definition

1. Definition of data

The English term "data" (1654) originates from "datum", a Latin word for "a given"[1]. "Data" is a single factual, a single entity, a single point of matter.

The word "data" to mean "transmittable and storable computer information" was first used in this sense in 1946. The expression "data processing" was first used in 1954.

Thoughts: the etymology suggests that data is "a given". Can you question this?

Data represents either a single entity, or a collection of such entities ("data points"). We can speak also of datasets instead of data (so a dataset is a collection of data points).

2. The variety of data sets.

A date	A color	A grade
A relation of friendship	A sound	A heartbeat
A user input	A duration	A curriculum vitae

A picture	A longitude and latitude	A price
A number of friends	A temperature	A list of favorite movies
etc…	etc…	etc…

These examples are chosen on purpose to be varied and from unexpected places. They illustrate three principles:

a. Think about data in a broad sense

Data is not just numerical, neither is it "what sits in my spreadsheets". You should train in thinking about data in a broader sense:

- pictures are data
- language is data (including slang, lip movements, etc.)
- relations are data: individual A is known, individual B is known, **but the relationship between A and B is data as well**
- preferences, emotional states… are data
- etc. There is no definitive list, you should train yourself looking at business situations and think: "where is the data?"

b. metadata is data, too

Metadata is a piece of data describing another data.

Example:

```
The bibliographical reference ①
describing
a book ②
```

① the metadata

② the data

→ Data without metadata can be worthless (imagine a library without a library catalogue)

→ Metadata can be informative in its own right, as shown with the NSA scandal (read this article from the New Yorker about NSA and metadata[2]).

c. zoom in, zoom out

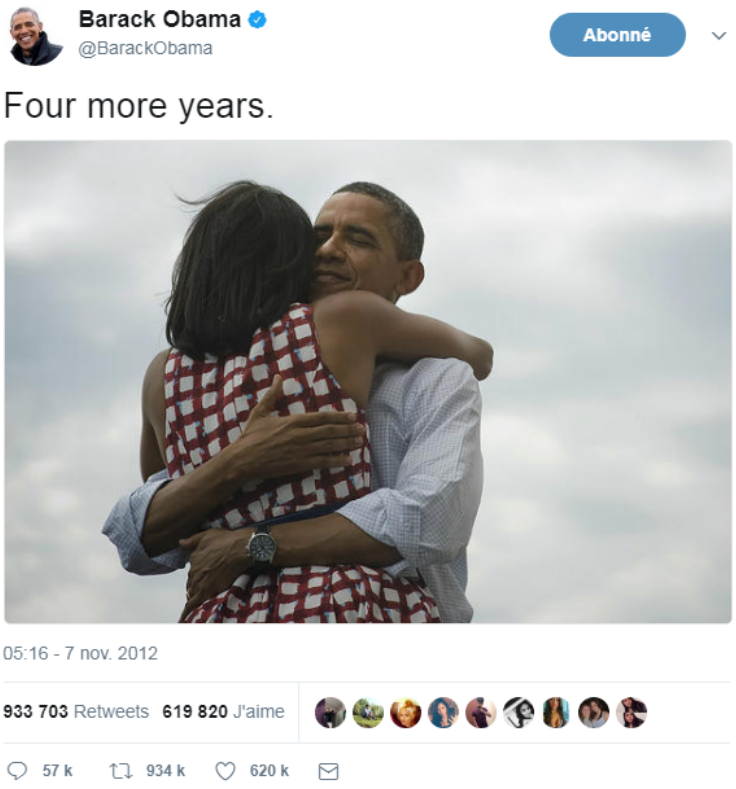
We should remember considering that a data point can be itself a collection of data points:

- a person walking into a building is a data point.
- however this person is itself a collection of data points: location data + network relations + subscriber status to services + etc.

So it is a good habit to wonder whether a data point can in fact be "unbundled" (spread into smaller data points / measurements)

3. How to describe datasets

a. Formats, types, encoding



- This is a digital **medium** (because it's on screen as opposed to analogic, if we had printed the pic on paper)
- The **type** of the data is textual + image
- The text is **formatted** in plain text (meaning, no special formatting), as opposed to more structured data-interchange formats (check json or xml[3]).
- The **encoding** of the text is UTF-8. Encoding has to do with the issue: how to represent alphabets and signs from different languages in text? (not even mentioning emojis?). UTF-8 is an encoding which is one of the most universal.
- The tweet is part of a list of tweets. The list represents the **data structure** of my dataset, it is the way my data is organized. There are many alternative data structures: arrays, sets, dics, maps...
- The tweet is stored as a picture (png file) on my hard disk. "png" is the **file format**. The data is **persisted** as a file on disk (could have been stored in a database instead).

b. Tabular data

Tabular data is a common way to handle datasets, by organizing it in lines and columns:

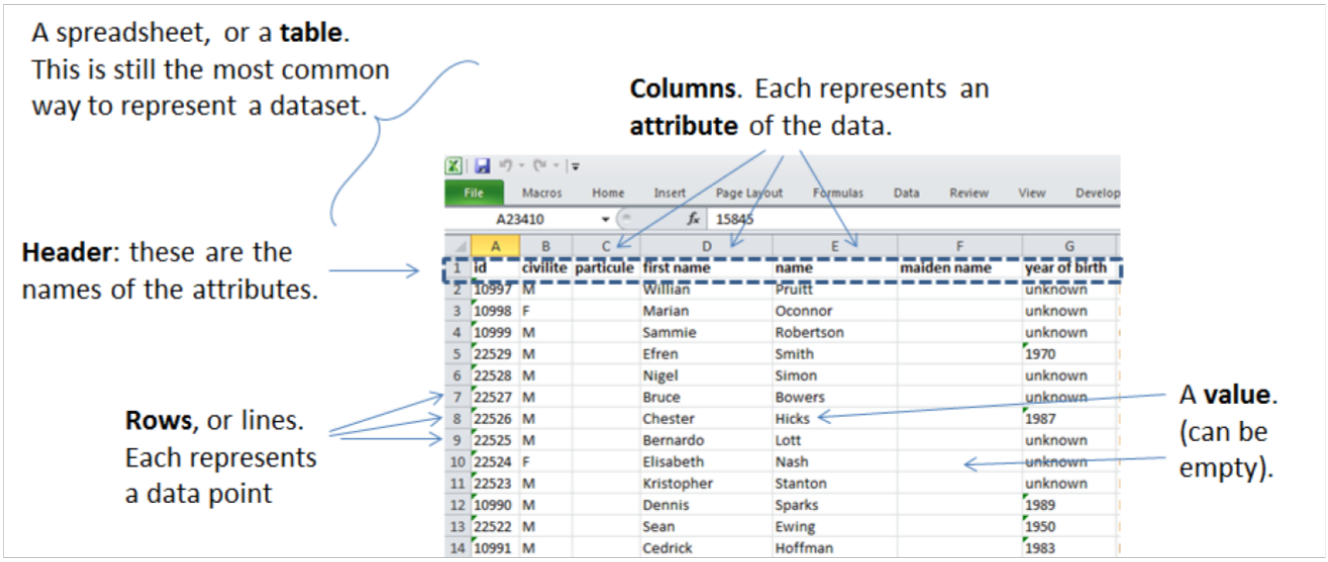


Figure 1. tabular data

c. First party, second party and third party data

- **First party data** : the data generated through the activities of your own organization. Your organization own it, which does not mean that consent from users is not required, when it comes to personal data.
- **Second party data** : the data accessed through partnerships. Without being the generator nor the owner of this data, partners make it available to you through an agreement.
- **Third party data** : the data acquired via purchase. This data is acquired through a market transaction. Its uses still comes with conditions, especially for personal data.

d. Sociodemo data vs behavior data

- Sociodemogaphic or **sociodemio** data refers to information about individuals, describing fundamental attributes of their social identity: age, gender, place of residence, occupation, marital status and number of kids.
- **Behavior data** refers to any digital trace left by the individual in the course of it life: clicks on web pages, likes on Facebook, purchase transactions, comments posted on Tripadvisor…

Sociodemo data is typically well structured or easy to structure. It has a long history of collection and analysis, basically since census exists.

Behavior data allows to go further than sociodemo data: each individual can be characterized by its acts and tastes, well beyond what an age or marital status could define.

But behavior data is typically not well structured and harder to collect.

4. Data and size

1 bit		can store a binary value (yes / no, true / false…)
8 bits	1 byte (or octet)	can store a single character
~ 1,000 bytes	1 kilobyte (kb)	Can store a paragraph of text
~ 1 million bytes	1 megabyte (Mb)	Can store a low res picture.

~ 1 billion bytes	1 gigabyte (Gb)	Can store a movie
~ 1 trillion bytes	1 terabyte (Tb)	Can store 1,000 movies. Size of commercial hard drives in 2017 is 2 Tb.
~ 1,000 trillion bytes	1 petabyte (Pb)	20 Pb = Google Maps in 2013

2. A clarification of big data

1. Big data is a mess

Figure 2. Facebook post by Dan Ariely in 2013

Jokes aside, defining big data and what it covers needs a bit of precision. Let's bring some clarity.

2. The 3 V

Big data is usually described with the "3 Vs":

V for Volume

The size of datasets available today is staggering (ex: Facebook had 250 billion pics in 2016).

We should also note that the volumes of data are increasing at an **accelerating rate**. According to sources, "90% of all the data in the world has been generated over the last two years"[4] (statement from 2013) or said differently, "More data will be created in 2017 than the previous 5,000 years of humanity"[5]

V for Variety

This is a bit less intuitive. "Variety" means here that data is increasingly unstructured and messy, and this is an important characteristic of the "big data" phenomenon. To caricature a bit, try to picture a shift from A to B:

A - Structured data

Structured data refers to data which is formatted and organized according to a well defined set of rules, which makes it **machine readable**. For example, zip codes are a structured dataset because they follow a precise convention regarding the number of letters and digits composing them, making it easy for an optical reader and software to identify and "read" them. Same with license plates, social

security numbers…

But these are simple examples.

What about, for instance, a tax form? If each field of the form is well defined, then the data collected through the form can be said to be "structured". By contrast, a form where the user can write free text (think of a comment on a blog post, or a blank space where users can write a feedback) produces unstructured data: data which does not follow a special convention for its size and content. This is typically much harder for software to process, hence to analyze.

To summarize, think of structured data as as anything that can be represented as well organized tables of numbers and short pieces of text with the expected format, size, and conventions of writing: phonebooks, accounting books, governmental statistics…

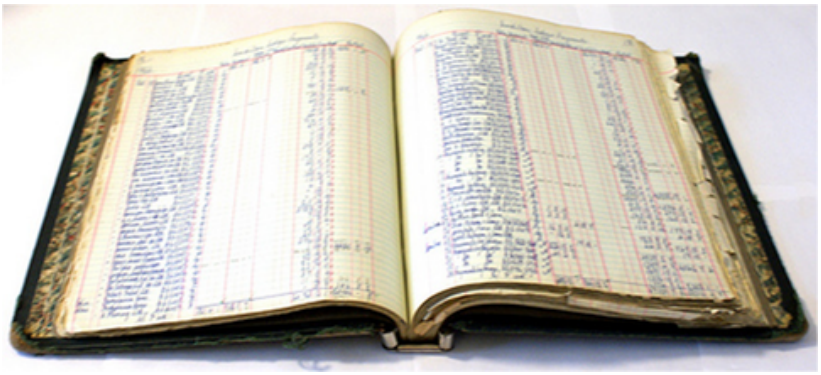


Figure 3. A book of accounts showing structured data

B - Unstructured data

Unstructured data refers to datasets made of "unruly" items: text of any length, without proper categorization, encoded in different formats, including possibly pictures, sound, geographical coordinates and what not…

These datasets are much harder to process and analyze, since they are full of exceptions and differences. But they are carry typically rich information: free text, information recorded "in the wild"…

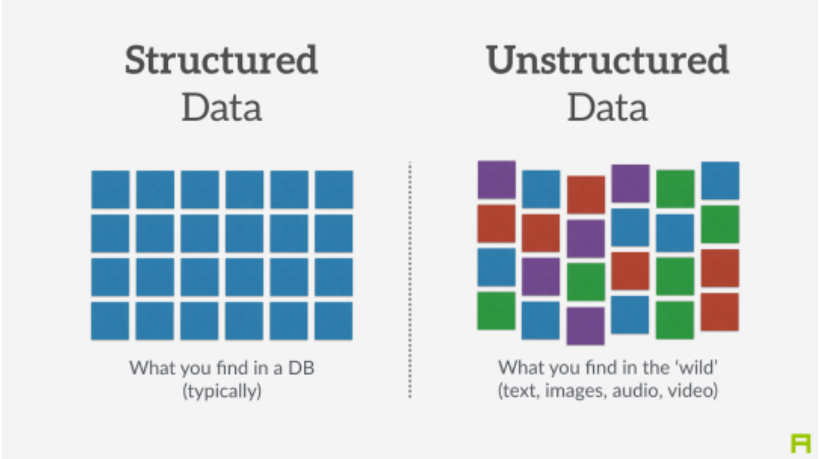


Figure 4. Structured vs unstructured data

V for Velocity

In a nutshell, the speed of creation and communication of data is accelerating (examples taken from here[6]):

- Facebook hosts 250 billion pics? It receives 900 million more pictures **per day**
- Examining tweets can be done automatically (with computers). If you want to connect to Twitter to receive tweets in real time as they are tweeted, be prepared to receive in excess of 500 million tweets **per day**. Twitter calls this service the "firehose"[7], which reflects the velocity of the stream of tweets.
- **Sensor data** is bound to increase speed as well. While pictures, tweets, individual records… are single item data sent at intervals, more and more sensors can send data **in a continuous stream** (measures of movement, sound, etc.)

So, velocity poses challenges of its own: while a system can handle (store, analyze) say 100Gb of data in a given time (day or month), it might not be able to do it in say, a single second. Big data refers to the problems and solutions raised by the velocity of data.

A 4th V can be added, for Veracity

Veracity relates to trustworthiness and compliance: is the data authentic? Has it been corrupted at any step of its processing?

We will devote a session of this course to data compliance, which is a broad topic covering data privacy, cybersecurity, and the societal impacts of data.

You can start reading the documents for this course here[8]

3. What is the minimum size to count as "big data"? It’s all relative

There is no "threshold" or "minimum size" of a dataset where "data" would turn from "small data" to "big data".

It is more of a **relative** notion: it is big data if current IT systems struggle to cope with the datasets.

(see Wikipedia definition[9] developing on this.)

"Big data" is a relative notion… how so?

a. relative to time

- what was considered "big data" in the early 2000s would be considered "small data" today, because we have better storage and computing power today.

- this is a never ending race: as IT systems improve to deal with "current big data", data gets generated in still larger volumes, which calls for new progress / innovations to handle it.

b. relative to the industry

- what is considered "big data" by non tech SMEs (small and medium-sized entreprises) can be considered trivial to handle by tech companies.

c. not just about size

- the difficulty for an IT system to cope with a dataset can be related to the size (try analyzing 2 Tb of data on your laptop…), **but also** related to the content of the data.
- For example the analysis of customer reviews in dozens of languages is harder than the analysis of the same number of reviews in just one language.
- So the general rule is: the less the data is structured, the harder it is to use it, even if it's small in size (this relates to the "V" of variety seen above).

d. no correlation between size and value

- Big data is often called "the new oil"[10], as if it would flow like oil and would power engines "on demand".
- Actually, big data is **created**: it needs work, conception and design choices to even exist (what do I collect? how do I store it? what structure do I give to it?). The human intervention in creating data determines largely whether data will be of value later.
- Example: Imagine customers can write online reviews of your products. These reviews are data. But if you store these reviews without an indication of who has authored the review (maybe because reviews can be posted without login oneself), then the reviews become much less valuable. Simple design decisions about how the data is collected, stored and structured have a huge impact on the value of the data.

So, in reaction to large, unstructured and badly curated datasets with low value at the end, a notion of "smart data" is sometimes put forward: data which can be small in size but which is well curated and annotated, enhancing its value (see also here[11]).

e. as an expression, "big data" is evolving

- It is interesting to note that "hot" expressions, like "big data", tend to wear out fast. They are too hyped, used in all circumstances, become vague and over sold. For big data, we observe that it is peaking in 2017, while new terms appear:

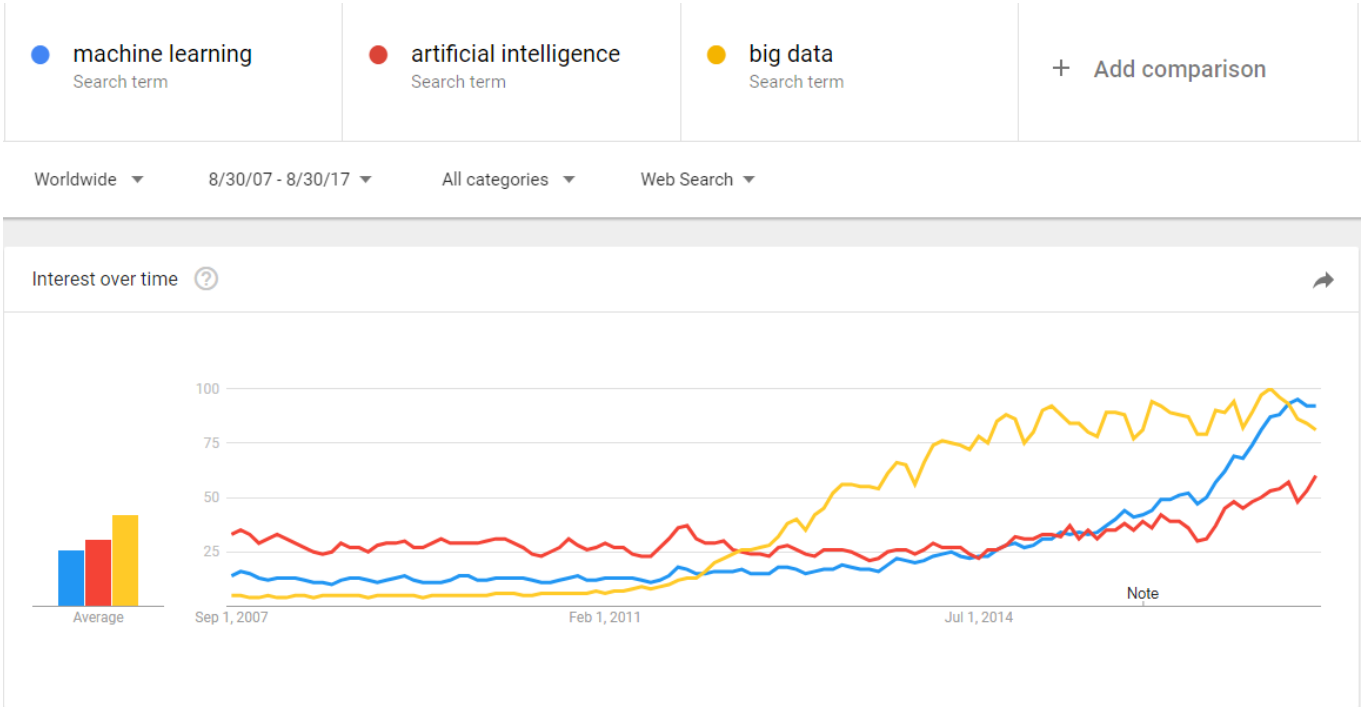


Figure 5. Google searches for big data, machine learning and AI

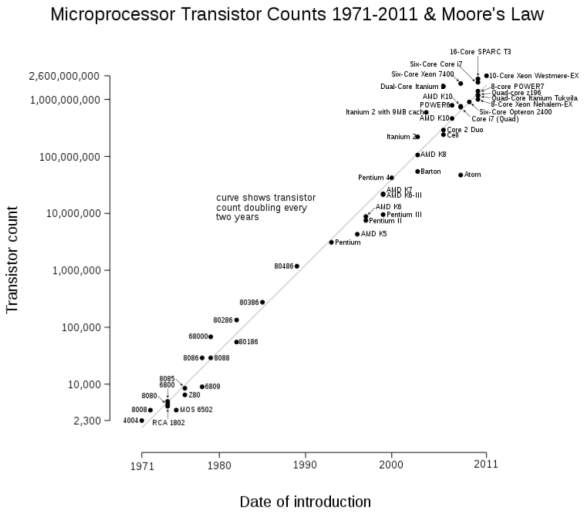
What are the differences between these terms?

- "Big data" is by now a generic term
- **Machine learning** puts the focus on the scientific and software engineering capabilities enabling to do something useful with the data (predict, categorize, score...)
- **Artificial intelligence** puts the emphasis on human-like possibilities afforded by machine learning. Often used interchangeably with machine learning.
- And **data science** ? This is a broad term encompassing machine learning, statistics, and many analytical methods to work with data and interpret it. Often used interchangeably with machine learning. **Data scientist** is a common job description in the field.

4. Where did big data come from?

a. Data got generated in bigger volumes because of the digitalization of the economy

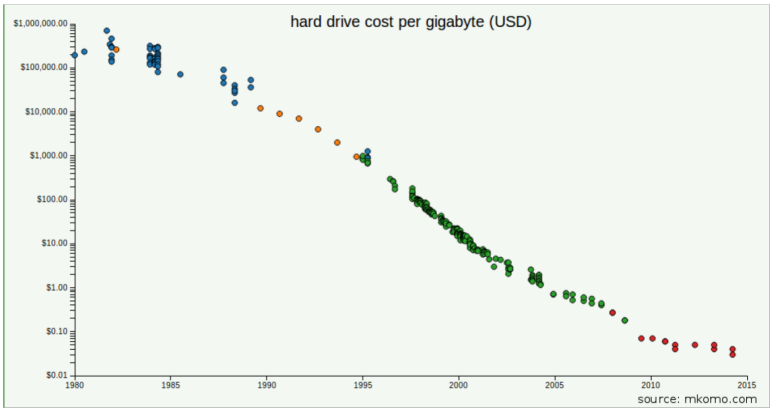
b. Computers became more powerful



source: https://en.wikipedia.org/wiki/Moore%27s_law

Figure 6. Moore's law

c. Storing data became cheaper every year



source: http://www.mkomo.com/cost-per-gigabyte

Figure 7. Decreasing costs of data storage

d. The mindset changed as to what "counts" as data

- Unstructured data (see above for definition of "unstructured") was usually not stored: it takes a lot space, and software to query it was not sufficiently developped.
- Network data (also known as graphs) (who is friend with whom, who likes the same things as whom, etc.) was usually neglected as "not true observation", and hard to query. Social networks like Facebook made a lot to make businesses aware of the value of graphs (especially social

graphs[12]).

- Geographical data has democratized: specific (and expensive) databases existed for a long time to store and query "place data" (regions, distances, proximity info…) but easy-to-use solutions have multiplied recently.

e. With open source software, the rate of innovation accelerated

In the late 1990s, a rapid shift in the habits of software developers kicked in: they tended to use more and more open source software, and to release their software as open source. Until then, most of the software was "closed source": you buy a software **without the possibility** to reuse / modify / augment its source code. Just use it as is.

Open source software made it easy to get access to software built by others and use it to develop new things. Today, all the most popular software in machine learning are free and open source.

See the Wikipedia article for a developed history of open source software: https://en.wikipedia.org/wiki/History_of_free_and_open-source_software

f. Hype kicked in

The Gartner hype cycle[13] is a tool measuring the maturity of a technology, differentiating expectations from actual returns:

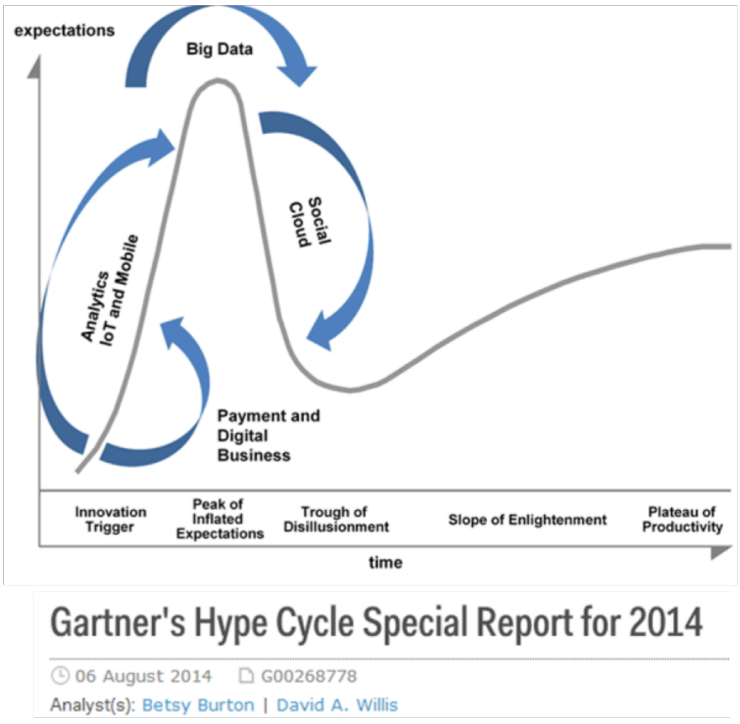


Figure 8. Gartner Hype Cycle for 2014

This graph shows the pattern that all technologies follow along their lifetime:

- at the beginning (left of the graph), an invention or discovery is made in a research lab, somewhere. Some news reporting is done about it, but with not much noise.
- then, the technology starts picking the interest of journalists, consultant, professors, industries… expectations grow about the possibilities and promises of the tech. "With it we will be able to [insert amazing thing here]"
- the top of the bump is the "peak of inflated expectations". All techs tend to be hyped and even over hyped. This means the tech is expected to deliver more than it surely will, in actuality. People get overdrawn.
- then follows the "Trough of Disillusionment". Doubt sets in. People realize the tech is not as powerful, easy, cheap or quick to implement as it first seemed. Newspapers start reporting depressing news about the tech, some bad buzz spreads.
- then: slope of Enlightenment. Heads get colder, expectations get in line with what the tech can actually deliver. Markets stabilize and consolidate: some firms close and key actors continue to grow.
- then: plateau of productivity. The tech is now mainstream.

(all technology can "die" - fall into disuse - before reaching the right side of the graph of course).

In 2014, big data was near the top of the curve: it was getting a lot of attention but its practical use in 5 to 10 years were still uncertain. There were "great expectations" about its future, and these expectations drive investment, research and business in big data.

In 2017, "big data" is still on top of hyped technologies, but is broken down in "deep learning" and "machine learning". Note also the "Artificial General Intelligence" category:

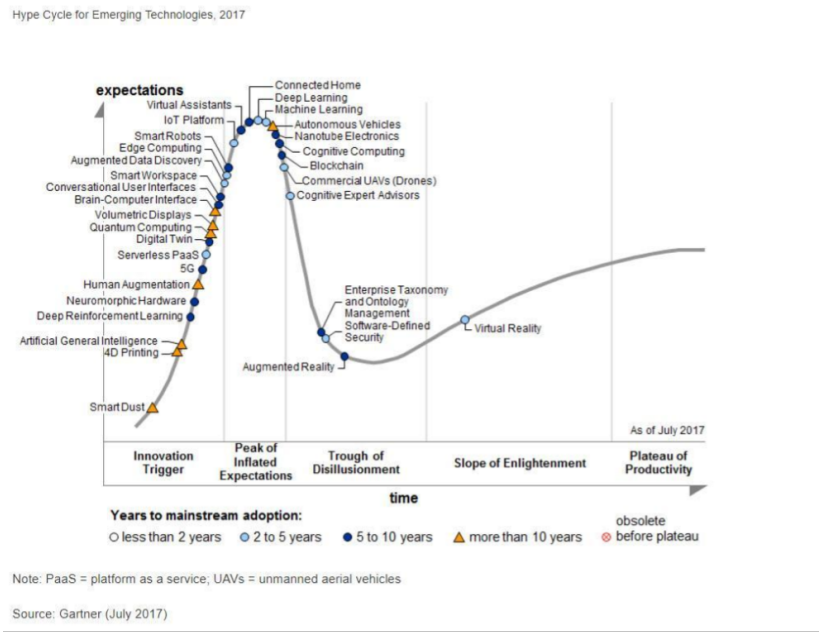


Figure 9. Gartner Hype Cycle for 2017

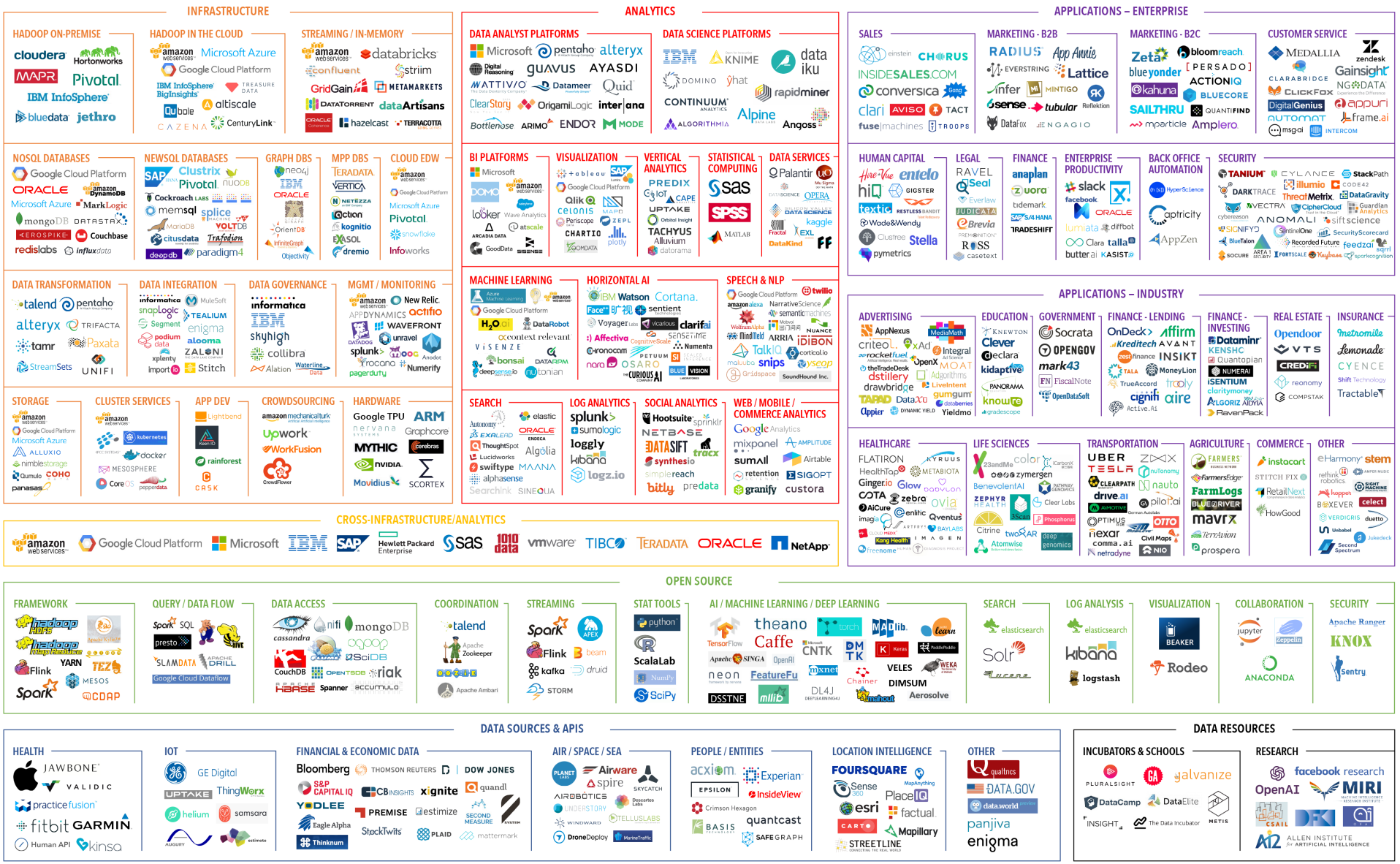
g. Big data transforms industries, and has become an industry in itself

Firms active in "Big data" divide in many sub-domains: the industry to manage the IT infrastructure for big data, the consulting firms, software providers, industry-specific applications, etc...

Matt Turck, VC at FirstMarkCap[14], creates every year a sheet to visualize the main firms active in these subdomains.

This is the 2017 version:

BIG DATA LANDSCAPE 2017



V2 – Last updated 5/3/2017

 mattturck.com/bigdata2017

FIRSTMARK
EARLY STAGE VENTURE CAPITAL

Figure 10. Big data landscape for 2017

You can find a high res version of this pic, an Excel sheet version, and a very interesting comment all here[15].

5. What is the future of big data?

a. More data is coming

The **Internet of things** designates the extension of Internet to objects, not just web pages and emails (see here for details[16]).

The **IoT** is used to **do** things (display information on screen, pilote robots, etc.) but also very much to **collect data** in their environments, through sensors).

The development of **connected objects** will lead to a tremendous increase in the volume of data collected.

b. Discussions about big data will fuse with AI

Enthusiasm, disappointment, bad buzz, worries, debates, promises… the discourse about AI will grow. AI is fed on data, so the future of big data will intersect with what AI becomes.

c. Regulatory frameworks will grow in complexity

Societal impacts of big data and AI are not trivial, ranging from racial, financial and medical discrimination to giant data leaks, or economic (un)stability in the age of robots and AI in the workplace.

Public regulations at the national and international levels are trying to catch up with these challenges. As technology evolves quickly, we can anticipate that societal impacts of big data will take center stage.

3. What is "the cloud"?

1. Note on the terminology: what is a server?

To understand the cloud precisely, We need first to have a clear vision of what a server is. A server is simply a computer stripped of everything unessential (screen, mouse, graphic card, sound card, keyboard)…

To illustrate: when Google is calculating what the best results are for your search "what are cheap and delicious restaurants in Lyon", this calculation must be done on a computer, right?

The kind of computer used to do this calculation is **not** this one:

Figure 11. A desktop computer

The reason is, we don’t need a screen, mouse, desktop, and not even the big box containing the computer itself. They take too much space, consume energy, and there is no need for them.

So when all the unnecessary parts are removed, the computer looks like this, and is called a "server":

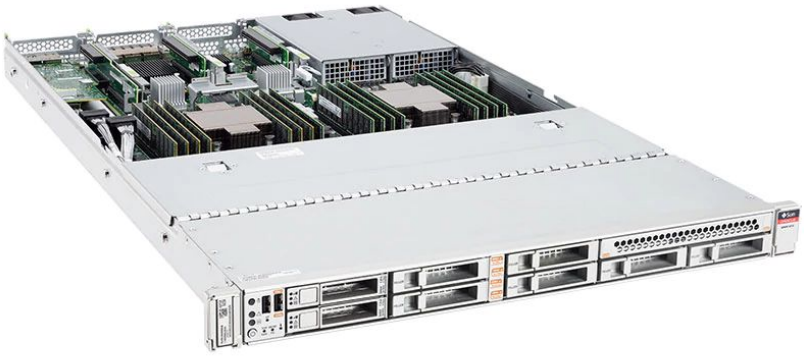


Figure 12. A server

source: https://www.oracle.com/servers/sparc/s7-2/index.html

Take a look at the shape: rectangular and very slim. This makes it easy to stack up servers one on the other. Because for Google and other companies crunching data for their business, a lot of servers are needed, so gaining on space is a real issue.

When many servers are piled up together and put in a big tall box, this is called a **rack** of servers, and look like this:

Figure 13. A rack of servers

When all the racks of servers are put in the same room, this is called a **data center** and looks like this:

Figure 14. A data center

To get a sens of how "the cloud" is actually a physical space, watch this video showing a tour of a data center at Google:

▶ https://www.youtube.com/watch?v=XZmGGAbHqa0 *(YouTube video)*

Usually, until 2005 roughly, companies had two options:

- buying their own servers and using them on their premises (at their location).
- paying the services of companies specializing in managing data centers.

Then the "cloud" changed this.

2. The cloud

The term **cloud** was made popular by Amazon with their service “Amazon Elastic Compute Cloud”: **Amazon EC2** launched in 2006.

This service was new in many ways:

- you can rent servers owned by Amazon, at a distance, when you need them, for a duration that you choose.
- there is an emphasis on ease of use: no need to know the technical details of these servers (how they are plugged, how they are configured…)
- you are just given a login + password and you can start using these servers for your needs.
- it’s "elastic": if you need more servers, or more powerful servers, it’s just possible. No need for signing a new contract or to evaluate whether Amazon has the capacity… it’s dimensioned to be possible.

Let’s compare a situation with or without the cloud:

Without the cloud	With the cloud
You make a market study for which server to buy Get the approval by your finance department to buy it (that's a fixed asset!) Wait for the server to ship Install it and configure it Maintain it (security, etc.) When the job is over: what do you do with your server? That's a sunk cost. If the job happens to need more computing capacity than your server offers: you are stuck with your too-small-server!	On Amazon' EC2 website[17], you click to choose a server among those on offer: it is **on demand** You run your job on it. Costs are metered precisely. When your job is over, you stop the server with a click and pay the bill. If the job happens to need more computing capacity, you switch to a bigger server with a single click, or it can be done for you automatically: it is **elastic**.
It is a capex	It is an opex

The cloud can fit in the budget as an operational expense instead of a capital expenditure. Opex are not inherently a better or cheaper option than a capex, but they are easier for a project team or business unit to fit in their budget. See this blog post[18], especially the critical comments below the post, to continue this discussion.

3. IaaS, PaaS, Saas

What is the use of the cloud?

Companies can use it to run elementary operations, up to more complex ones:

Infrastructure as a service (IaaS)

The cloud is used to replace the company's local IT infrastructure needs such storing data, or computing operations.

Platform as a Service (Paas)

The cloud is used to run the building blocks of a service: to manage a messaging system, to host apps, …

Software as a Service (Saas)

The cloud is used to host a full software accessible "on demand" through the browser: like Google Drive, Brightspace[19] or SalesForce[20].

4. Private or public cloud? Hybrid cloud?

- Amazon EC2 is an example of a **public cloud**: it is publicly accessible to any customer. Of course,

this does not mean that every customer can see what the others are doing on the cloud! Each customer have their private spaces on the cloud.

- Many companies have security requirements which prevent them from accessing public clouds. They need to have their servers on premises. In this case, they can build their own **private cloud**: it is a cloud just like Amazon EC2, except that it is owned, managed and used by the company exclusively - it is not accessible to third parties. But even private, it keeps the basic characteristics of a cloud: on-demand and elastic in particular.
- **Hybrid clouds** are a variety of private clouds: it is a private cloud where some forms of operations can be delegated to a public cloud. For example, operations which are not security sensitive and which need a capacity of computing in excess of what the private cloud of the company can provide.

4. The headache of data integration

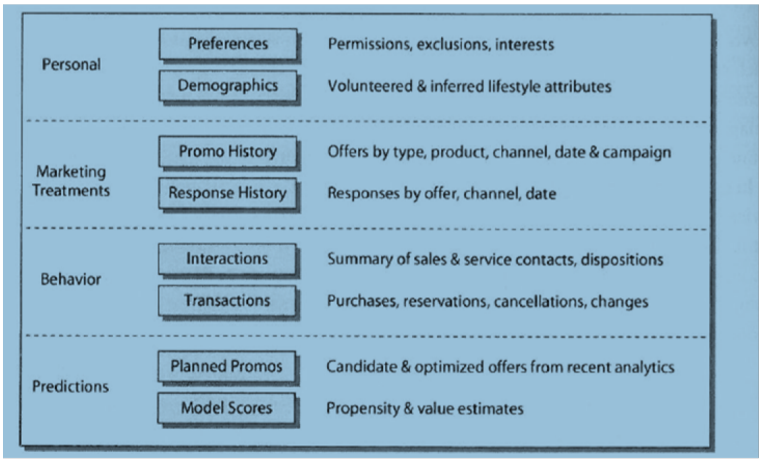
1. Data: you don’t get in on tap

A naive vision of data would get that it’s all fluid. Don’t we talk about "data streams"? Working with data would be as simple as opening a tap and "getting customer data", for instance.

Actually, data is more like a complex patchwork: many different pieces which must be stitched together - and this is hard.

Take customer data.

It is not a given. Instead, this is a design made of multiple primary data sources:



Source: UNICA Corporation, in *Multichannel Marketing*, by A. Arikan (2008).

Figure 15. Multiple sources of customer data

> Analysts often spend 50-80% of their time preparing and transforming data sets before they begin more formal analysis work.

Garrett Grolemund, Data Scientist and Master Instructor at RStudio[21]

Take away: Data is fragmented by nature. It comes from different sources and presented in different formats. **You** (the marketer in collaboration with data scientists) wrangle to *construct* customer profiles by joining and assembling different sources of data into a meaningful synthesis.

Data scientists actually have entire books devoted to the subject of wrangling with the complexity of integrating different data sources.

2. Sources of fragmentation

a. Channels keep diversifying

Point of sale, print, TV, radio, outdoor posters, mobile apps, mobile sites, emails, SMS, APIs, social networks, search engines, e-commerce platforms, e-commerce websites, blogs, content channels, ...

→ all these channels can provide relevant data.

b. Connections between these channels intensify and complexify

- Social TV is TV delivered with Internet services,
- User profiles created on one platform are imported on another
- Orders taken online can be picked up on a variety of point of sales
- Ads circulating through one channel replicate on other channels, ...

→ It is very complex to trace the "customer journey" on all these channels and to keep an updated view of a customer profile.

It is even more difficult to explore causality (which action on which channel caused which subsequent action by the user?)

c. Underlying technologies fragment and keep evolving, across channels

Browsers, Cookies, APIs, mobile OS (Android or iOS?), etc... All these different techs evolve and need continuous effort and expertise to integrate.

Example: **did you notice** that on a mobile device, the url of the pages you visit can now start with an AMP url[22]?

Like, to visit the page of the New York Times the url should be http://www.nyt.com but it looks like: **http://google.com/amp/www.nyt.com**

This **http://google.com/amp** prefix is a new tech by Google to accelerate the display of web pages on mobiles. Fine. But then, as a marketing data analyst, how to count visits to:

http://google.com/amp/www.nyt.com

and

http://www.nyt.com

→ It is important to count visits to these two urls as a visit **to the same page**.

In Sept 2017 major services of web data analytics were still struggling with this issue[23].

This illustrates that to just count visits to a web page (something which should be classic and robust)

and integrate this data to a larger analysis, big issues can arise and be hard to fix even in 2017, because of the evolution of techs and standards.

d. In the meantime, customers have growing expectations about the quality of service

Difficulties posed by data integration do not slow or decrease customers expectations. To the contrary, we see an elevation of expectations. Customers increasingly expect:

- realtime contact
- two-ways interaction (they want to be able to voice their opinion, and get a response)
- seamless experience (no glitch, modern UI, consistence of the UX across channels)
- personalized experience (customization of the message they receive)

e. Example: A French bank going through the 2010s

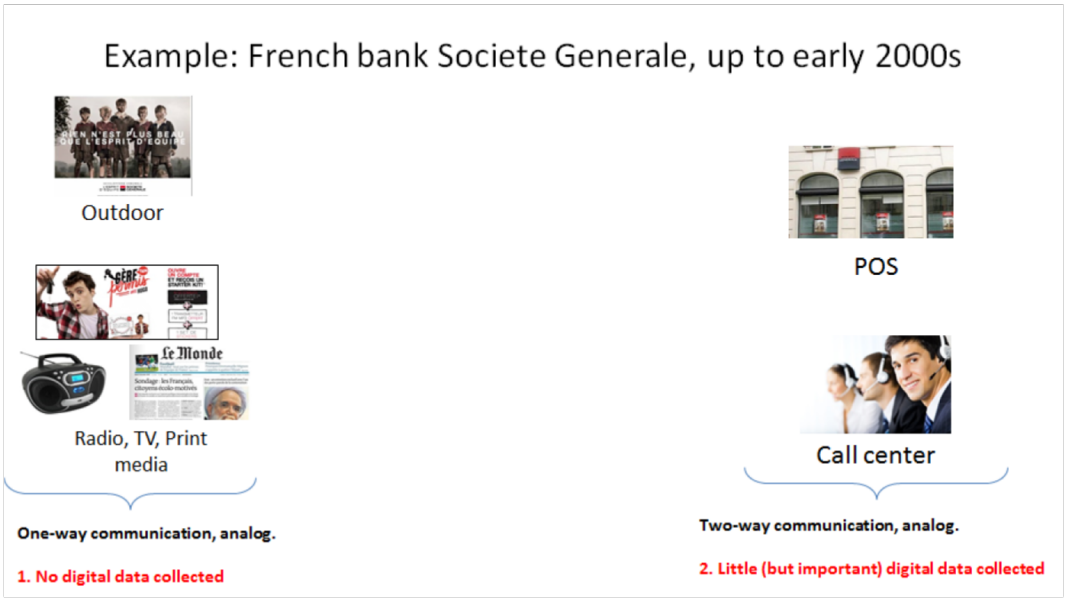


Figure 16. Before - a couple of data sources across a few channels

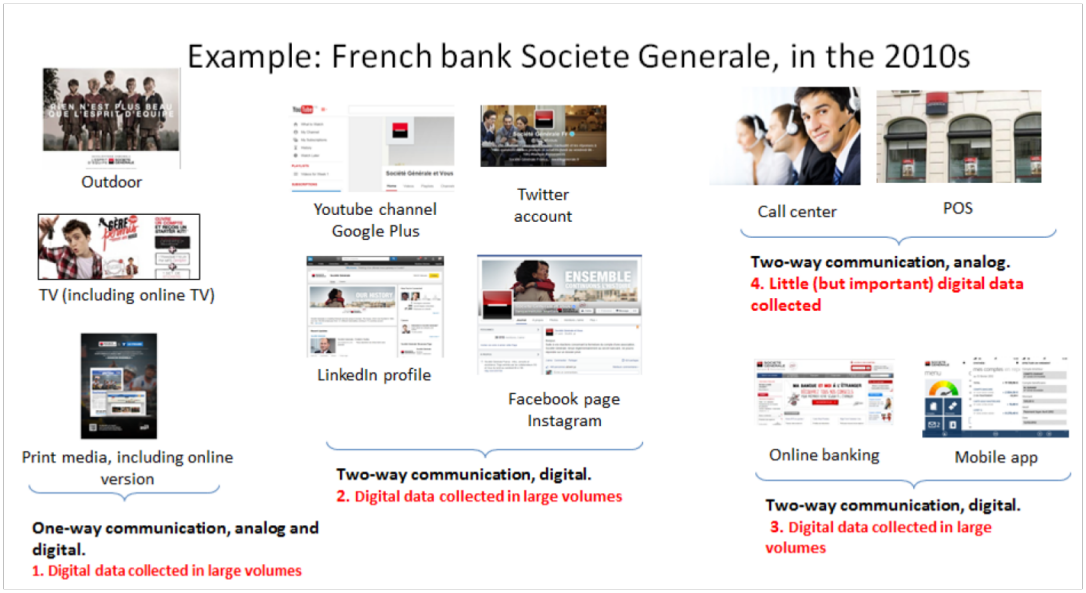


Figure 17. Now - many data sources across a variety of channels

3. Tools for data integration: DMPs and more

a. Data Management Platform (DMP)

In 2015/2016 a new acronym started to trend: "DMP", standing for **"Data Management Platform"**.

Basically a **DMP** is an information system dedicated to solving the issues of data integration:

- it can store a large amount of data
- it can receive data from a variety of sources, in a variety of formats
- it offers functions to reconcile records from different data sources and generate a unique identifier for each reconciled entry.
- it offers segmentation / classification functions
- it provides security and analytics capabilities on the data
- it makes this data available for execution by other software.

b. DMP in relation to other components of the information system

DMPs are relatively new. They integrate with 3 other information systems in the firm:

- CRM
  - This is the software **gathering** data related to customers and sales. It is a major source of **input data** for a DMP.

- ERP
    - Large software synchronizing information systems from finance, sales, logistics and more. The CRM can be independent or part of the ERP.
- DSP
    - Piece of software automatizing ad buying[24]. The audiences identified in the DMP could be served corresponding ads with a DSP.
- Data lake
    - Data lakes are databases specializing in storing large amounts of unstructured data. They respond to the need of "storing today, for future use". A data lake is not meant to be optimized for queries. They are meant to store everything we collect today, in the case this data will serve future usages. When this happens, data can be extracted from the data lake and put in a database, in a form which makes it convenient to query and analyze.

How can data circulate across these software and with the external world? The next chapter is devoted to APIs, another essential concept.

5. APIs and their business relevance

1. Definition of API

API: acronym for **Application Programming Interface**

An API is the way to make software programs “easy to plug and share” with other programs.

An API is simply a group of rules (you can also call it a convention, or an agreement…) which programmers follow when writing the part of their code which is in charge of communicating with other software.

These rules are then published (on a webpage for example), so that anyone who needs to connect to the program can learn what rules to follow.

So, an API is simply a way to write code to make it easy to interface with other programs? Yes. Why the fuss then?

Having conventions on how to write a software so that somebody can plug it to its own software is one thing. APIs of this sort are a classic topic in computer science[25] but we are not concerned with this here.

APIs we are going to discuss are about communication between distant computers, in a business context.

Let’s do a bit of history:

2. The origin of APIs

Companies which need to exchange data is nothing new. Manufacturers, retailers, banks, … they need to exchange information at regular interval.

Sending invoices, receiving receipts for merchandise, and many other administrative records generated in the course of business.

These receipts, invoices… can be printed and mailed (this solution still exists of course).

With informatics developing in the 1970s and 1980s, a new system emerged: the exchange of information via computers: Electronic Data Interchange[26] (EDI)

a. EDI: Electronic Data Interchange

EDI is not an exchange of file attachments in emails or via a file transfer on a website, because emails and websites did not exist yet! (emails and the Web were adopted by firms in the late 1990s).

Instead, exchanging data via EDI consisted in using complex electronic tools (like the fax but even more complicated) because:

- each industry has its own protocol to exchange data (one protocol for logistics, one for payments, one for this or that retailer chain, etc.)
- you need a dedicated device or software for each EDI protocol, and these are not given for free
- EDI protocols can vary from one country to another
- EDI protocols are controlled by industry associations which do not adopt innovation quickly

and finally, EDI protocols created "closed systems": a company A can connect to company B via an EDI only if the two have a pre-agreement to use this EDI.

So EDIs are fragmented, complicated to implement, slow to evolve, not cheap and restricts the communication to a "club" of partners who agreed to use it.

EDIs still exist, especially in large B2B industries like transportation (check here[27]), but it lost in popularity in the wider economy because… APIs have arrived.

b. The emergence of web APIs

In the late 1990s and early 2000s, Internet and the World Wide Web expanded dramatically.

More and more servers in different parts of the world needed to be interfaced with each other to exchange data.

It became increasingly convenient to define simple and universal conventions that everyone could learn and follow to standardize these exchanges, for free and easily.

That's what **web APIs** do. They are also often called:

- **API** for short
- **web services**
- **REST** API (see below for this last one).

A **web API** extends the logic of the APIs we have seen in the beginning of this document, to software communicating via the web.

To recall, an API is a convention followed when writing a software, making this software available to other software.

NOTE

Example: the API of Microsoft PowerPoint enables the import of Excel tables in pptx documents, because the API of Powerpoint plugs to the API of Excel. In this example Excel and Powerpoint are suposed to be installed on the same computer of course!

A Web API is an API which enables two pieces of software to communicate, via Internet. **They don't need to be installed on the same computer.**

c. The benefits of a web API compared to an EDI

Unlike an **EDI** , a web API drops any industry-specific concern. Web APIs are just a convention to send and receive data over the Internet, without any saying on the content of the data.

The data sent and received can be invoices, webpages, train schedules, audio, video… whatever.

Contrary to an EDI, a company creating a web API can choose to leave its access open (remember that EDIs need the two parties to have a pre-established agreement).

So that a potential client interested in using the web API of a company can set it up in a couple of clicks, instead of waiting weeks or months before a contract is signed and the EDI is setup.

Saying that APIs are open does not mean an absence of security : communication through APIs can easily be identified and encrypted, as needed.

d. REST API?

Two popular web API conventions emerged in the 1990s and competed for popularity:

- SOAP (Simple Object Access Protocol[28])
- REST (Representational State Transfer[29])

REST APIs became ultimately the most widely adopted, because it uses the same simple principles that webpages use to be transferred over the Internet (the "http" protocol that you see in web page addresses). This is why APIs are often called "REST APIs"[30].

In 2000-2010, it became increasingly easy and natural to adopt the REST convention to make one's software and data available to another computer. This simple evolution to ease interoperability had **immense effects**:

3. Business consequences of APIs

a. APIs opened software to the world

An API transforms a closed software into something that can be plugged to anything other computer or object, as long as it is connected to the Internet.

For instance, APIs were a key factor of success for SalesForce[31] in the early 2000s. SalesForce, created in 1999, has a revenue of US$8.39 billion in 2017:

- SalesForce developed a CRM as a SaaS where features of the CRM were **exposed as APIs** (meaning, these features could be plugged to external apps via the REST protocol).
- SalesForce created a PaaS to host apps that could plug to the SalesForce CRM via the APIs developed by SalesForce.

This platform is called Force.com[32] and external developers can put their apps there, as long as they

are compatible with the SalesForce API.

SalesForce takes a commission on the sales made by these third party apps hosted on Force.com, but more importantly, the platform creates an **ecosystem** of apps and developers around the SalesForce products which makes it hard for a customer company to switch to a different product.

b. APIs accelerated software innovation

Thanks to API it became easy to add software blocks together and create new apps, even if the app developers where from different countries, industries, or big and small. Check this amazing story[33].

c. APIs opened data

Companies and public organization own many datasets of great business interest. The use of these datasets can be free (for small projects and NGOs) or monetized if the user is an enterprise.

Without APIs, datasets can be made publicly available as docs (eg, Excel spreadsheets) to download but this is not practical (try downloading something like `all_train_schedules_2000_to_2017.xls` !).

So, imagine a transportation company like French SNCF which finds it interesting to publish station names, train schedules, etc. because it could be used by other companies to build new services : how can it do it?

The data is on a server of SNCF. Then SNCF adds an API and its documentation[34], making the data available to anyone who knows about REST APIs (and this is trivial[35]).

Entrepreneurs and programmers in general will be able to access the data via the API and use it, possibly to create new services based on this train information. **Open data** designates this movement to make datasets available to a broad audience, and web APIs have been a key technological ingredient in this movement.

4. The ecosystem of APIs

a. A wealth of APIs

To discover new APIs, or to make your APIs easier to discover, the most well known place is the website "Programmable Web": https://www.programmableweb.com/

Searching on this website, you will find APIs ranging from the most business-y[36] use case, to APIs of a more fun and odd sort[37].

Still, many APIs are not listed on this website, and a google search for "info I need + API" is also a good way to find if the API you'd need exists. Interested in whale sightings? There is an API for that[38].

b. APIs: a business world of its own

APIs have become central to the economy. As a result, a large number of services associated to APIs

have developed to cater for all the needs of companies that use them:

- how to create an API
- how to manage the documentation of a large number of APIs
- how to connect a wide variety of APIs
- how to control and audit the security of APIs
- how to monetize and API…

→ Many large firms and startups now specialize in all these different issues. Here is the 2017 landscape of the main companies active in the API industry:

DESIGNED BY
Mehdi Medjaoui

The API Landscape

Last Update: March, 2017

POWERED BY
spoke intelligence

Business Processes as an API/API-as a Product/Transactional APIs (151)

SMS / Voice APIs (22)

Fax / Mail (5)

Email APIs (20)

Search Engine APIs (5)

Payments APIs (25)

Accounting API (10)

eCommerce APIs (4)

AI APIs (4)

Data-as-a-service/Messaging (50)

Online Forms(2)

Humans as a Service APIs (2)

HR APIs (2)

API Lifecycle platform (78)

API Documentation Platform (6)

API Testing (6)

API Management (16)

API Analytics / Monitoring (5)

Streaming APIs (9)

Developer portal (5)

IOT Platforms/Networks/Devices API (7)

Access level and Identity management (20)

API Oriented IT Consulting Firms (8)

Backend Building Tools/MBaaS (14)

MBaaS (7)

GraphQL BaaS (3)

DBaaS (4)

API Abstraction / Integration Platform (32)

Horizontal API abstractions (5)

Automation/ Orchestration (8)

Vertical API Abstractions (13)

Domotics (6)

Figure 18. The API landscape in 2017

6. Essential notions on privacy and data protection

1. Privacy: just one aspect of data protection

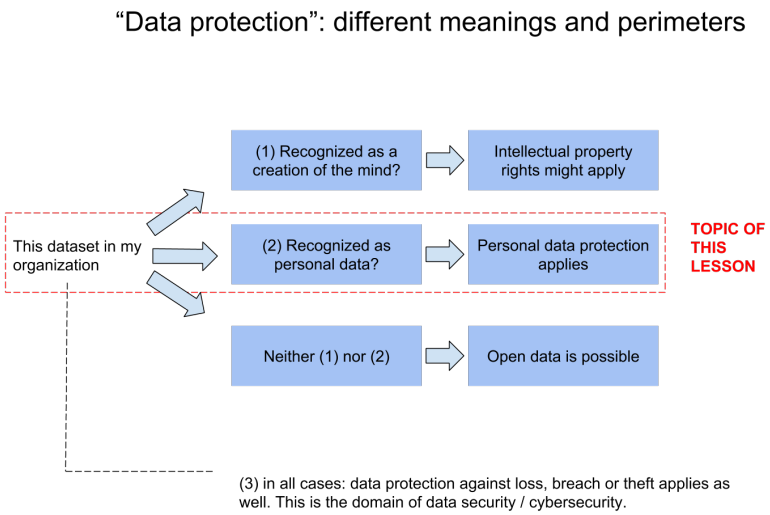


Figure 19. Defining data protection

2. When is personal information considered "data"?

At the most basic level, anything could count as "data" with possibly a personal character to it, including comments written about somebody in a personal notebook.

In practice, "data" starts to to be considered as such when:

This is information capable of being processed **automatically**

→ Hint: data on computers, not unstructured written notes

Or information intended to be processed automatically

→ Hint: paper records to be fed in a computer (eg, via scanning), not any pile of paper on your desk.

Or **structured information** that can be used to facilitate the retrieval of specific information on specific individuals

→ Hint: paper records, filing systems, databases

3. Personal data matters because of privacy

> Personal data are any anonymous data that can be double checked to identify a specific individual (e.g. fingerprints, DNA, or information such as “the son of the doctor living at 11 Belleville St. in Montpellier does not perform well at school”).
>
> — CNIL (French Independent Administrative Authority), https://www.cnil.fr/en/personal-data-definition

Personal data is data that an individual has the right to keep private. **How and why is privacy an issue**?

Privacy is mentioned in the Article 12 of the 1948 Universal Declaration of Human Rights[39]:

> No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation. Everyone has the right to the protection of the law against such interference or attacks.
>
> — Universal Declaration of Human Rights

This article from the Declaration is found in similar forms in most of the conventions on human rights in the world.

This **right to privacy** enables individuals to define their identity in relation to the world, by giving each individual the power to control what to keep for themselves, and what to reveal / share with the world.

In 2005, a report on Privacy in the Digital Environment[40] by the Haifa Center of Law & Technology develops:

> The right to privacy is our right to keep a domain around us, which includes all those things that are part of us, such as our body, home, property, thoughts, feelings, secrets and identity.

> The right to privacy gives us the ability to choose which parts in this domain can be accessed by others, and to control the extent, manner and timing of the use of those parts we choose to disclose.

In addition to shaping an individual’s own personal sphere and identity, privacy also underpins the development of a relation between the individual and the society she is part of:

→ when an individual’s privacy is secured, they don’t have to fear that their personal opinions and activities (as simple as reading a newspaper) will endanger them as citizens.

→ this gives liberty to individuals to develop political expressions which do not necessarily conform

with the power structure in place. This would be much harder if everyone's political opinions could not be kept private.

4. Evolution of privacy

Privacy is a social norm which transforms as societies evolve. Since the 2000s, a couple of tendencies can be identified:

- increasing tracking of the digital traces left by individuals by companies which use these traces for ad targeting and data reselling
- increasing state surveillance through digital means, against security threats and unspecified goals.
- broader public acceptance of new forms of violations to privacy.

For example, TV shows[41] where participants are filmed 24/24 and where they reveal their (real or supposed) intimacy, dates back only from the late 1990s.

Figure 20. The Truman Show, 1998

5. Privacy of the consumer and privacy of citizens: the relations between the two

Thanks to whistleblowers like Edward Snowden[42], the extent of privacy breaches by governmental agencies is now better known.

This trailer for "CitizenFour" gives a sense of the dangers whistleblowers face when revealing how governmental agencies spy on their citizens:

▶ https://vimeo.com/108771171 *(Vimeo video)*

Journalists, academics, activists and NGOs such as the Electronic Frontier Foundation[43] make the case that:

- consumers are insufficiently aware and sensitive of how much information is captured in the normal conduct of their lives, just by using mobile phones and apps, web browsing, and increasingly in public places.
- citizens are insufficiently aware and sensitive of the breach of their privacy by security agencies of their own country of residence, and by other countries.

Many citizens consider that if they don't break the law, then they have "nothing to hide".

Similarly, consumers might find that bargaining their private data against a free service and some targeted ads, is a good deal.

Sociologist of technology Zeynep Tufekci[44] goes further:

Her argument is that besides "surveillance" and "lack of privacy", companies like Google and Facebook developing a business model based on ad targeting by analytics on personal data, design a **persuasion architecture** which can be used / highjacked for political purposes.

Tufekci does not argue that Google, Facebook or the likes inherently have anti-democratic purposes, but that:

- they develop of an information architecture which has the potential to shape opinions of crowds,
- they do so without transparency
- some past experiments on voting in the US, and current developments on electronic surveillance in China, show that the power of these technologies has already consequences in the real world.

link to the TEDx conference by Zeynep Tufekci[45]

6. Conclusion: data protection in business, more than an regulatory obligation

The collection and treatment of personal data by businesses has far reaching implication, and should not be considered merely from a legal standpoint by firms.

The topic engages the Corporate social responsibility[46] of the firm.

The nature of the business model itself - profiling consumers in the most specific way - has profound consequences on the design of the environment surrounding individuals.

What are the next steps? Several trends can be identified:

1. Some voices question the business model: are personalized ads based on personal data as effective as the market valuation of Facebook suggests? How much is just scam? Some voices warn against the extent of the fraud, as the video below shows (see also here[47], or here[47]):

▶ https://www.youtube.com/watch?v=oVfHeWTKjag *(YouTube video)*

2. Legislation by political authorities to protect the public interest, especially via an obligation for transparency, in the face of more personal data being collected, for a larger variety of purposes.

3. A deepening of the current model with more personal data being collected, in private spaces (homes) and behavior in public places (crowd management in streets, stadiums, etc.):

Figure 21. Echo Alexa

Echo Alexa is a home assistant with a conversational interface, providing services personalized with the data provided by the user.

7. Machine learning, data science and artificial intelligence

1. Explaining machine learning in simple terms

a. A comparison with classic statistics

Let's compare[48] machine learning to something we would call "regular statistics":

A basic method in statistics is to compute a regression line to identify a trend from a scatter plot.

To illustrate, we take some data about marketing budgets and sales figures in the corresponding period:

"Regular statistics" enables, among other things:

1. to find the numerical relation between the 2 series, based on a pre-established formal model (eg, ordinary least squares[49]).

→ we see that sales are correlated with marketing spendings. It is likely that more marketing spending causes more sales.

2. to predict, based on this model:

→ by tracing the line further (using the formal model), we can predict the effect of more marketing spending

"Regular statistics" is advanced by scientists who:

1. are highly skilled in mathematics

→ their goal is to find the exact mathematical expression defining the situation at hand, under rigorous conditions

→ a key approach is **inference**: by defining a **sample of the data** of just the correct size, we can reach conclusions which are valid for the entire dataset.

2. have no training in computer science / software engineering

→ they neglect how hard it can be to to run their models on computers, in terms of calculations to perform.

→ since they focus on **sampling** the data, they are not concerned with handling entire datasets with

related IT issues.

Machine learning does similar things to statistics, but in a slightly different way:

- there is an emphasis on getting the prediction right, not caring for identifying the underlying mathematical model
- the prediction needs to be achievable in the time available, with the computing resources available
- the data of interest is in a format / in a volume which is not commonly handled by regular statistics package (eg: images, observations with hundreds of features)

Machine learning is advanced by scientists who are typically:

1. highly skilled in statistics (the "classic" statistics we have seen above)
2. with a training or experience in computer science, familiar with working with unstructured data / big data
3. working in environments (industry, military, ...) where the operational aspects of the problem are key determinants (unstructured data, limits on computing resources)

Machine learning puts a premium on techniques which are "computationally adequate":

- which need the minimum / the simplest algebraic operations to run: the best technique is worthless if it's too long or expensive to compute.
- which can be run in such a way that multiple computers work in parallel (simultaneously) to solve it.

(footnote: so machine learning, in my opinion, shares the spirit of "getting things done" as was operations research in the early days[50])

The pursuit of improved models in traditional statistics is not immune to the notion of computational efficiency - it does count as a desirable property - but in machine learning this is largely a pre-requisite.

b. An illustration: the case of the GPU

A key illustration of the difference between statistics and machine learning can be provided with the use of **graphic cards** .

Graphic cards (or GPUs: graphics processing units) are these electronic boards full of chips found inside a computer, which are used for the display of images and videos on computer screens:

In the 1990s, video gaming developed a lot from arcades to desktop computers. Game developers created computer games showing more and more complex scenes and animations. (see an evolution of graphics[51], and advanced graphics games in 2017[52]).

These video games need powerful video cards (aka GPUs[53]) to render complex scenes in full details - with calculations on light effects and animations **made in real time**.

This pushed for the development of ever more powerful **GPUs** . Their characteristics is that they can compute simple operations to change pixel colors, **for each of the millions of pixels of the screen in parallel**, so that the next frame of the picture can be rendered in milliseconds.

Millions of simple operations run in parallel for the price of a GPU (a couple of hundreds of dollars), not the price of dozens of computers running in parallel (can be dozens of thousands of dollars)? This is interesting for computations on big data!

If a statistical problem for prediction can be broken down into simple operations which can be run on a GPU, then a large dataset can be analyzed in seconds or minutes on a laptop, instead of cluster of computers.

To illustrate the difference in speed between a mathematical operation run without / with a **GPU** :

▶ https://www.youtube.com/watch?v=-P28LKWTzrI *(YouTube video)*

The issue is: to use a GPU for calculations, you need to conceptualize the problem at hand as one that can be:

- broken into a very large series
- of very simple operations (basically, sums or multiplications, nothing complex like square roots or polynomials)
- which can run independently from each other.

Machine learning typically pays attention to this dimension of the problem right from the design phase of models and techniques, where statistics would typically not consider the issue, or only downstream: not at the design phase but at the implementation phase.

Now that we have seen how statistics and machine learning differ in their approach, we still need to understand how does machine learning get good results, if it does not rely on modelling / sampling the data like statistics does?

Machine learning can be categorized in 3 families of tricks:

2. Three families of machine learning

a. The unsupervised learning approach

Unsupervised learning designates the methods which take a fresh dataset and find interesting patterns in it, **without inferring from previous, similar datasets**.

The analogy is with a person doing a task for the first time:

→ she learns a new thing by applying clever heuristics, without having been training on the task before.

Example: in your wedding, how to sit people with similar interests at the same tables?

The set up:

- a list of 100 guests, and 3 tastes you know they have for each of them
- 10 tables with 10 sits each.
- a measure of similarity between 2 guests: 2 guests have similarity of 0% if they share 0 tastes, 33% if they share 1 taste, 66% with 2 tastes in common, 100% with three matching interests.
- a measure of similarity at the level of a table: the sum of similarities between all pairs of guests at the table (45 pairs possible for a table of 10).

A possible solution using an unsupervised approach:

- on a computer, assign randomly the 100 guests to the 10 tables.
- for each table:
  - measure the degree of similarity of tastes for the table
  - exchange the sit of 1 person at this table, with the sit of a person at a different table.
  - measure again the degree of similarity for the table: if it improves, keep the new sits, if not, revert to before the exchange

And repeat for all tables, many times, until no exchange of sits improves the similarity. When this stage is achieved, we say the model has "**converged**".

b. The supervised learning approach

Supervised learning is the approach consisting in calibrating a model based on the history of past experiences in order to guess / predict a new occurrence of the same experience.

Take 50,000 or more observations, or data points, like:

**an image of a cat, with the caption "cat"

**an image of a dog, with the caption "dog"

**another image of a cat, with the caption "cat"

etc....

- you need 50,000 observations of this kind, or more! It is called the **training set** .
- this is also called a **labelled dataset** , meaning that we have a label describing each of the observation.

The task is: if we give our computer a new image of a cat without a label, will it be able to guess the label "cat"?

The method:

- take a list of random coefficients (in practice, the list is a vector, or a matrix).
- for each of the 50,000 pictures of dogs and cats:
  - apply the coefficients to the picture at hand (let's say we have a dog here)

    - If the result is "dog", do nothing, it works!
    - If the result is "cat", change slightly the coefficients.
    - move to the next picture
- After looping through 50,000 pictures the parameters have hopefully adjusted and fine tuned. This was the **training of the model**.

Now, when you get new pictures (the **fresh set**), applying the trained model should output a correct prediction ("cat" or "dog").

Supervised learning is currently the most popular family of machine learning and obtains excellent results especially in image recognition, even though some cases remain hard to crack:

It is called **supervised** learning because the learning is very much constrained / supervised by the intensive training performed:

→ there is limited or no "unsupervised discovery" of novelty.

▶ https://www.youtube.com/watch?v=4HCE1P-m1l8 *(YouTube video)*

Important take away on the supervised approach:

- **collecting *large* datasets for training is key**. Without these data, no supervised learning.
- supervised learning is not good at analyzing situations entirely different from what is in the training set.

c. The reinforcement learning approach

To understand reinforcement learning in an intuitive sense, we can think of how animals can learn quickly by **ignoring** undesirable behavior and rewarding desirable behavior.

This is easy and takes just seconds. The following video shows B.F. Skinner, main figure in psychology in the 1950s-1970s:

▶ https://www.youtube.com/watch?v=TtfQlkGwE2U *(YouTube video)*

Footnote: how does this apply to learning in humans? On the topic of learning and decision making, I warmly recommend this book by Paul Glimcher[54], professor of neuroscience, psychology and economics at NYU:

(this is a very hard book to read as it covers three disciplines in depth. The biological mechanisms of decision making it describes can be inspiring to design new computanional approaches.)

Besides pigeons, reinforcement learning can be applied to any kind of "expert agents".

Take the case of a video game like Super Mario Bros:

Structure of the game / the task:

- Goal of the task: Mario should collect gold coins and complete the game by reaching the far right

of the screen.

- Negative outcome to be avoided: Mario getting killed by ennemies or falling in holes.
- Starting point: Mario Bros is standing at the beginning of the game, doing nothing.
- Possible actions: move right, jump, stand & do nothing, shoot ahead.

Reinforcement learning works by:

1. Making Mario do a new random action ("try something"), for example: "move right"
2. The game ends (Mario moved right, gets hit by a ennemy)
3. This result is stored somewhere:
    - move right = good (progress towards the goal of the game)
    - walking close to an ennemy and getting hit by it = bad
4. Game starts over (back to step 1) with a a combination of
    - continue doing actions recorded as positive
    - try something new (jump, shoot?) when close to a situation associated with a negative outcome

After looping from 1. to 4. thousands of times, Mario completes the game, without any human player:

▶ https://www.youtube.com/watch?v=qv6UVOQ0F44 *(YouTube video)*

Reinforcement learning is perceived as corresponding to an important side of human learning / human intelligence (goal oriented, "trial and error").

d. When is machine learning useful?

Using machine learning can be a waste of resource, when well known statistics could be easily applied.

Hints that "classic" statistical modeling (maybe as simple as a linear regression) should be enough:

- The dataset is not large (below 50k observations), supervised learning is not going to work
- The data is perfectly structured (tabular data)
- The data points have few features

Cases when "classic" statistics modeling is **necessary**:

- The question is about the relative contribution of independent variables to the determination of an outcome

3. Machine Learning and Data Science

Machine learning is a step in the longer chain of steps of data science.

The process was formalized as kdd[55]: "Knowledge Discovery in Databases":

More recent representations of the steps in data processing have been suggested, making room for the role of data visualization (see the lecture on the topic):

→ see the version by Ben Fry[56] (source[57]) and this one by Moritz Stefaner:

(source[58])

Machine learning is one of the techniques (along with traditional statistics) that intervenes at the step of "Data mining".

What makes data scientists important is that the steps of this kdd are highly interdependent.

You need individuals or teams who are not just versed in data mining:

→ because the shape of the data at the collection stage has a huge influence on the kind of techniques, and the kind of software, that can be used to discover knowledge.

The skills of a data scientist are often represented as the meeting of three separate domains:

source: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

4.

a. Weak vs Strong AI

Weak AI designates computer programs able to outperform humans at complex tasks with a narrow focus (playing chess)

Weak AI is typically the result of applying expert systems or machine learning techniques seen above.

Strong AI is an intelligence that would be general in scope, able to set its own goal, and conscious of itself. Nothing is close to that yet.

So AI is a synonymous with weak AI at the moment.

b. Two videos to understand AI further

Laurent Alexandre on the social and economic stakes of AI (in French):

▶ https://www.youtube.com/watch?v=rJowm24piM4 *(YouTube video)*

John Launchbury, the Director of DARPA's Information Innovation Office (I2O) in 2017:

▶ https://www.youtube.com/watch?v=-O01G3tSYpU *(YouTube video)*

8. 7 roads to data-driven value creation

7 roads to data-driven value creation

Not a closed list, not a recipe! Rather, these are essential building blocks for a strategy of value creation based on data.

1. PREDICT

Prediction: The ones doing it

1. Predictive churn / default / … (banks / telco)

2. Predicting crime 

3. Predicting deals 

4. Predictive maintenance 

Prediction: the hard part

1. Collecting data (cold start problem[59])

2. Risk missing the long tail, algorithmic discrimination, stereotyping

3. Neglect of novelty

2. SUGGEST

Suggestion: The ones doing it

1. Amazon’s product recommendation system 

2. Google’s “Related searches…” 

3. Retailer’s personalized recommendations

Suggestion: the hard part

1. The cold start problem[60], managing serendipity (see review: paying version[61], free version not available) and "filter bubble" effects (review: paying version[62], free version here[63]).

2. Finding the value proposition which goes beyond the simple “you purchased this, you’ll like that”

3. CURATE

Curation: The ones doing it

1. Clarivate Analytics curating metadata from scientific publishing 

2. Nielsen and IRI curating and selling retail data 

3. ImDB curating and selling movie data 

Curation: the hard part

1. Slow progress: curation needs human labor to insure high accuracy, it does not scale the way a computerized process would.

2. Must maintain continuity: missing a single year or month hurts the value of the overall dataset disproportionally.

3. Scaling up / right incentives for the workforce: the workforce doing the curation should be paid fairly, which is not the case yet[64].

4. Quality control

4. ENRICH

Enrichment: The ones doing it

1. Selling methods and tools to enrich datasets 

2. Selling aggregated indicators 

3. Selling credit scores

Enrichment: the hard part

1. Knowing which cocktail of data is valued by the market

2. Limit duplicability

3. Establish legitimacy

5. RANK / MATCH / COMPARE

Ranking / matching / comparing: The ones doing it

1. Search engines ranking results 

2. Yelp, Tripadvisor, etc… which rank places 

3. Any system that needs to filter out best quality entities among a crowd of candidates

Ranking / matching / comparing: the hard part

1. Finding emergent, implicit attributes (imagine: if you rank things based on just one public feature: not interesting nor valuable)

2. Insuring consistency of the ranking (many rankings are less straightforward than they appear)

3. Avoid gaming of the system by the users (for instance, companies try to play Google's ranking of search results at their advantage[65])

6. SEGMENT / CLASSIFY

Segmenting / classifying: The ones doing it

1. Tools for discovery / exploratory analysis by segmentation

2. Diagnostic tools (spam or not? buy, hold or sell? healthy or not?) 

Segmenting / classifying: the hard part

1. Evaluating the quality of the comparison

2. Dealing with boundary cases

3. Choosing between a pre-determined number of segments (like in the k-means) or letting the number of segments emerge

7. GENERATE / SYNTHETIZE (experimental!)

Generating: The ones doing it

(click on the logos to get to the relevant web page)

1. Intelligent BI with Aiden 
2. wit.ai, the chatbot by FB 
3. Virtual assistants 
4. Image generation 
5. Close-to-real-life speech synthesis 

6. Generating realistic car models from a few parameters by Autodesk: 

A video on the generation of car models by Autodesk:

▶ https://www.youtube.com/watch?v=25xQs0Hs1z0 *(YouTube video)*

Generating: the hard part

1. Should not create a failed product / false expectations
2. Both classic (think of ) and frontier science: not sure where it's going

Combos!

Figure 22. Combinations