SOUTHWORKS
MAKE EVERYTHING RIGHT

The Data Journey

Everything you need to know to learn from **your data**

Mauro Krikorian, Head of R&D, SOUTHWORKS

ABOUT THE AUTHOR

Mauro Krikorian is native from Buenos Aires, Argentina – fan of River Plate football club ("el millo"), of course. He loves complex problems and challenges that can be solved through technology.

As Head of Research and Development at SOUTHWORKS, Mauro brings his vast experience to bear for the many projects he leads for customers such as 7-Eleven, Axioma, City Football Group, Discovery, GlaxoSmithKline, Microsoft, NEC, Swisscom, and many others. Mauro also oversees several research workstreams of new and emerging technologies in order to bring emerging tech expertise to SOUTHWORKS clients for their competitive advantage.

During his academic life his interests were focused on language theory, combinatorial optimization and meta-heuristics, cryptography, low level programming, data structures and languages, probability and statistics tied to signal analysis for data compression and forecasting, and distributed systems.

During his time at University of Buenos Aires his essay on statistical and machine learning techniques to identify genes and proteins through their relationships, was presented at the "RPIC 2011 - XIV Reunión de Trabajo en Procesamiento de la Información y Control" congress for data and signal processing. During his time at University of Buenos Aires, he also researched hand-written recognition, leveraging an interesting machine learning approach from the supervised learning paradigm and modern image processing techniques.

During his professional career, Mauro has channeled those theoretical interests into many different practical areas such as applying cryptography related algorithms and techniques into different security protocols for secure data transmission and digital identity management, language theory in the realm of cognitive services, but also machine learning and AI, data structures within Big Data solutions and modern databases, and data compression techniques and concepts (one of his biggest passions) on everything where appliable – back when memory and storage capacities were much more limited, constrained and costly than today.

Throughout his career, Mauro has designed and built a wide range of tools, from low level coding for system automation and such, to front-line customer facing applications (internal to organizations, as well as for external or final consumers) - among his most interesting challenges at both extremes of the spectrum, he has worked on a series of low level, highly-efficient, algorithms and routines written directly in assembler to perform fast parsing and analysis of big amounts of data for insurance companies, to designing and building from scratch a semantic engine based on a type-1 grammar (defined by himself) that allowed non-technical users from financial institutions to build different set of rules addressed to assess credit risk.

Mauro has worked in many different contracting modes along the years, from being in-house, to freelance with different sets of responsibilities and challenges. In his freelance capacity, Mauro has worked for companies such as Microsoft, where he was part of the team building the new CI/CD platform for its Dynamic suite of products (starting with CRM back then); MuleSoft, where he led the team that developed the suite of connectors to integrate the MuleSoft platform with Microsoft's tech stack.

FOLLOW MAURO ON

Table of Contents

1. INTRODUCTION

With tons of data being generated constantly, and faster than ever, you know there is a huge opportunity to leverage data to improve your business with the right set of tools. But with so many platforms, frameworks and tools out there, you probably don't know where to start.

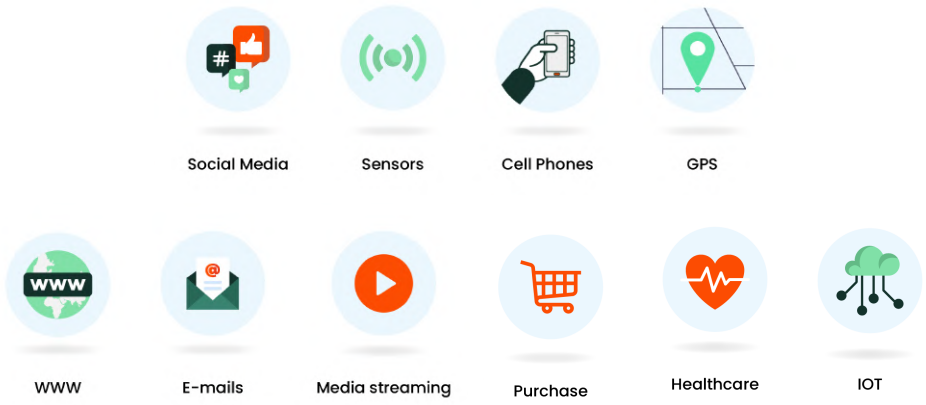
Whether you are looking to monitor your business performance through specific KPIs, better understand what your users want, take your business automation to the next level, or discover interesting insights, there is always an opportunity to improve and to get it right. (Our mantra is to Make Everything Right™, so we're always on the lookout for ways our clients and partners can leverage technology for better solutions).

Today, most companies, no matter how big or small they are, can generate enormous quantities of data. But very often, all of these data sources are stored within different data silos from different departments and teams - And the complexity starts to grow as these groups rely on different storage platforms, have different data capture patterns, transmission mechanisms, user permissions etc.

Depending on the type of business, it might also be the case that companies need to consume vast amounts of data generated by external actors and systems. Throughout this paper I tackle these scenarios, outlining solutions that will help you to walk up the DIKW pyramid from the very bottom level.

You will get to know about the different data pipelines that you need to build and put into operation to transform raw data into pieces of information, with the final goal of increasing your business knowledge by leveraging business intelligence platforms. I'll focus on everything you need to take care of to learn from your data, from information you want to extract from it aimed to improve specific business areas and KPIs, to new insights that you might not even be aware of – like identifying patterns and relationships within your data to build new pipelines on-top and to constantly keep evolving how your business operates.

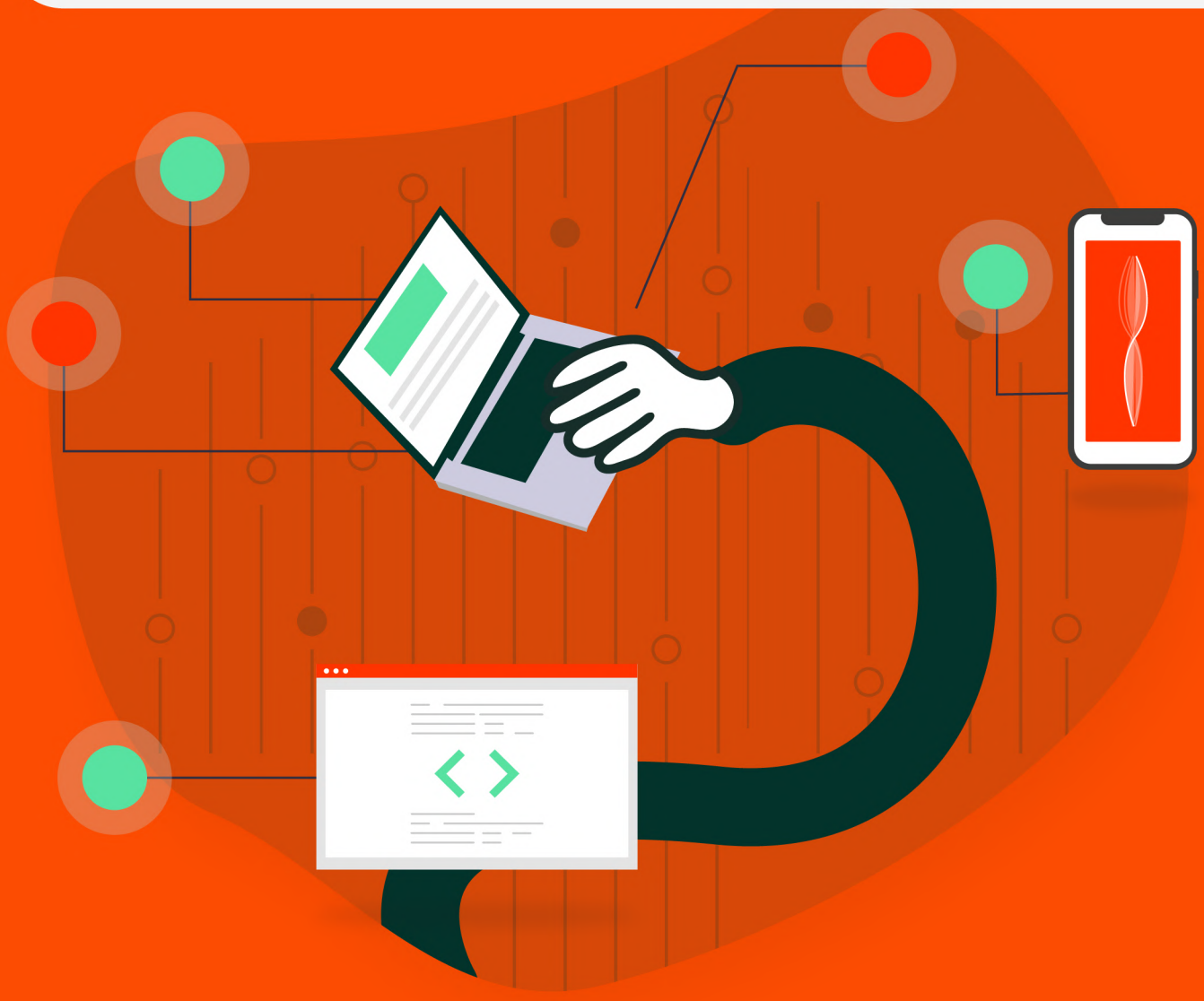
At the end of the journey, you will see how AI and machine learning techniques applied to your data, after applying different transformations and processes, will provide you valuable insights and great predictions that you can use to gain competitive advantage.



Enterprise data silos – per department or teams

03.

HOW DOES EVERYTHING START?



Information has always been a great source of power, ultimately related to knowledge - depending on the ability you have to understand it and apply it.

From the simplest piece of data to huge volumes of it, there is always an opportunity to learn from it. Years ago, this was very easy to achieve as there was not as much data out there being generated and harvested as is readily available today. Computer systems and algorithms that processed old amounts of data back then were more related to data mining topics. There were big analytics platforms that first required I/O intense operations to normalize and centralize your data, and later these platforms - online analytical processing (OLAP) tools - were able to perform a multi-dimensional analysis of that data. They had the ability to create reports for departments such as sales, marketing, finance, management, etc. and companies also used them for budgeting and forecasting.

But during recent years, data volumes increased significantly. They grew from within your organization and with the ability to also consume external sources, and allowed decision makers to obtain the best insights that enable you drive your business with greater evidence to back decision making. A new set of platforms and approaches to traditional data extraction, transformation and ingestion were born and most organizations benefit from them today – these are complementary to the old set of tools, but focus on other scenarios, other amounts of data, a variety set of data sources, and provides the ability to generate value in almost real-time.

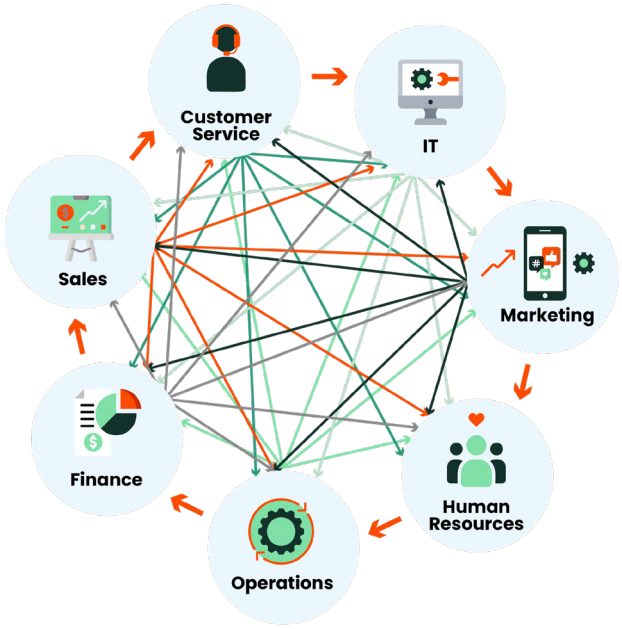
In this chapter, I will review the history of the problem, how it started and the rise of Big Data.

EXPLORATORY ANALYSIS

Let's start by recognizing that most stored data today, and data from legacy systems, is only available to some parts of your organization (and this is not due to permissions alone). It is available to departments, teams, or even individual employees, but it is not available to the entire organization the tap of a button.

These data silos rely on different platforms, have different sizes, access patterns, permissions, etc. This is not a bad thing in of itself, as they are sealed off from the rest of the network, increasing their security in the face of internal or external attackers. Nevertheless, there are times when these silos are created with no planning at all, accumulating all sorts of data as required by the individuals or groups that own them - and they keep growing in a big, unplanned, disorganized way.

The latter becomes a problem when your organization is accustomed to creating many of these 'ad-hoc' data silos. This exponentially leads to requiring more and more storage space, duplicating information in most cases, and it is a 'mission impossible' challenge for anyone within your organization to use them.



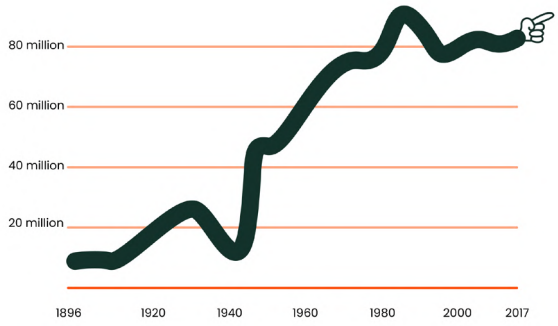
Sources of Date Generation:

And so far, I have only discussed everything that different departments and teams can generate within your organization; all the data your employees and systems can create from internal and/or external interactions.

But… what else is out there?

According to United Nations population statistics, the world population grew by 30%, or 1.6 billion humans, between 1990 and 2010. In number of people the increase was highest in India (350 million) and China (196 million) – countries that also have a high adoption of Internet access, computers, and mobile devices.

If we study population growth charts, we can see that during the last decades the net increase to the global population is steady, with an average of 80 million added every year. Absolute increase in global population per year

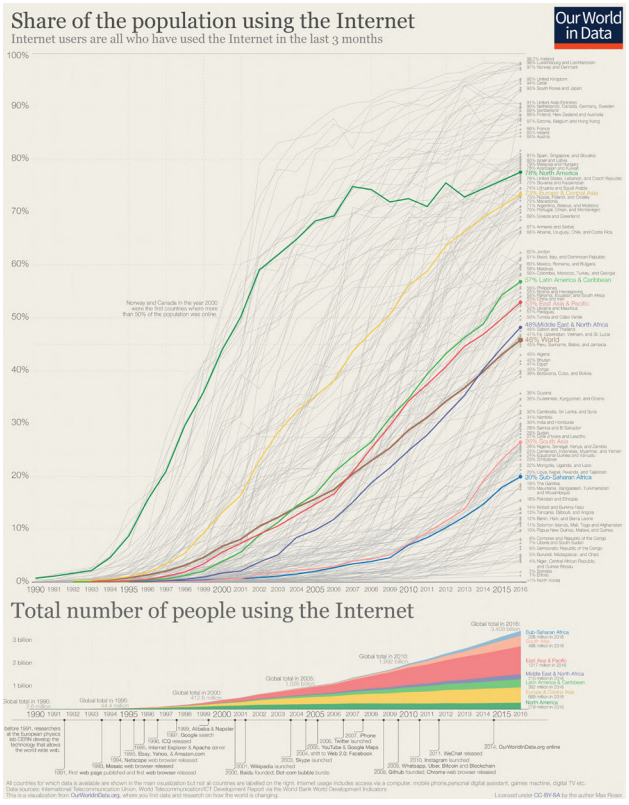


based on OWID, based on HYDE & UN

Absolute increase in global population per year

There are more and more people that, more frequently every day and at early ages, have access to a computer, a mobile device, and the Internet – generating more and more data.

In addition, if we look at internet adoption statistics, we also see that there is a big increment in the share of the global population since 1990 - as shown in the below chart from Our World in Data.



The share of the population and the total number of people using the Internet

Max Roser, Hannah Ritchie and Esteban Ortiz-Ospina (2015) - "Internet". Published online at OurWorldInData.org. Retrieved from: 'https://ourworldindata.org/internet' [Online Resource]

Being that the human population is not growing nearly as fast as Moore's law, data generation is not driven mainly by the pace at which population grows, but mainly by the way machines evolve - computers, IoT devices, interconnected systems, smart systems and gadgets, etc. Everything is connected in some way nowadays, generating all kinds of data constantly and at very high rates.

Here is where the previous problem grew exponentially, and the industry brought in a simple and clear concept for it: '**Big Data**'.

People and Machines constantly generating data with increased interconnectivity

Depending on the type of business you operate, you will be more interested in niche, big, or custom subsets of "Big Data" - but you cannot ignore it, this concept has come to stay. With computers being more powerful every year, their capabilities for analysis increase significantly as do their capabilities to generate more and more data at greater pace – so the definition of 'big' will require adjustment over time.

It is funny though, that this thing that we call Big Data is, in fact, mainly composed of very small chunks of data. And this brings to light another problem that I will address through this paper, which is the ability for storage systems to read and write them at the same high rates they are created – not to mention how to store them in terms of data structure, indexing for easy and fast access, and normalization to formats that are more accessible for humans or machines depending on what use you want to give them.

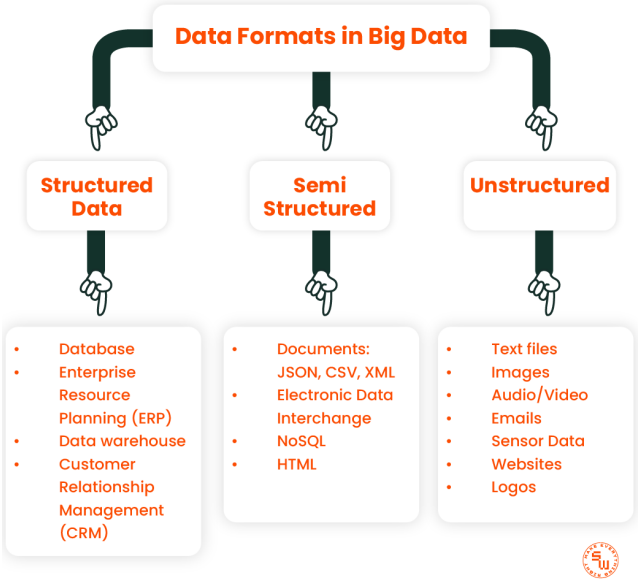
As I mentioned, there are different ways, rates, and more important, formats on how human and machines generate data. These raw formats are divided mainly into structured and unstructured, although another category which is semi-structured is sometimes applied to it.

There is not a specific category that can be tied to humans or machines, as both generate all types of it – structured and unstructured. Sometimes, when humans and machines interact closely, formats tied to the semi-structured category are used as these formats ease the readability for humans and simplify parsing for machines – these are, to mention a few, JSON, CSV, XML, HTML documents, as well as different early structured messages like UN/EDIFACT that were developed back in the 70's to fulfill Electronic Data Interchange (EDI). The latter focused on transferring structured data from one computer system to another without human intervention, but defined in a way that one or more data elements of an EDI message were intended for human interpretation.

If we discuss structured data today, we can see that most of it relies upon systems that enforce this structure one way or another. Here Relational Database Management Systems (RDBMS) and Dara Warehouses (DW), Enterprise Resource Planning (ERP) Platforms, Customer Relationship Management (CRM) Systems, enforce the actor using them to

follow a structure by having specific interfaces and constraints; from columns, tuples and relations defined in a DB schema, to constraints like unique keys, foreign keys, allowed and mandatory values, etc.; specific properties and attributes for ERP or CRM entities, and so forth and so on. Humans interacting with these kinds of systems must follow specific rules imposed by the underlying structure defined there, as must machines updating information automatically.

The last category, unstructured data, is what we can consider today as the heart of Big Data. As aforementioned, this is most of the time small chunks of data being generated automatically or manually, by machines and humans. From millions of Twitter messages begin written on a daily basis by millions of users, to millions of entries within log files that are captured by systems running 24x7. Humans using these platforms generate a lot of this kind of data daily, and not only text messages, but images, audio, and video. Machines today, with the rise of IoT, not only generate entries within logs (that sometimes depending on the platform might be structured), but also sensor data from thousands of devices connected to manage a smart-factory, to millions of them to build an entire smart-city.

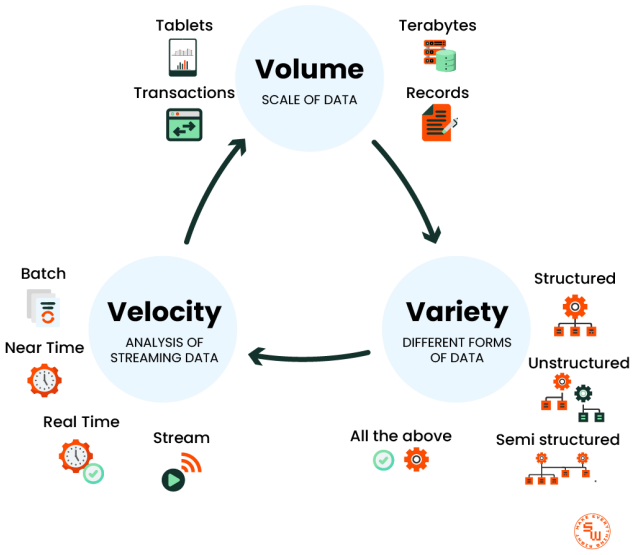


Different types of data formats

As I briefly mentioned there, some cases might have some kind of structure, but typically, where the problem arises is how to correlate all of these small chunks to obtain interesting insights and understand the big picture. It is not interesting just to know that within some intervals a stop-light changes from green to red and its bulbs are not burned out (which you get it in a structured format).

Of course you can have an elaborate control system built on top of it to monitor that all of them are in constant sync in the full grid, but, if you can correlate that to pedestrian messages complaining on Twitter about malfunctioning locations you will get more valuable information, not just the very basics pre-processed info from the raw, but greater insights that you can leverage in a much smarter way to prioritize and take proper actions – I will discuss this in the following sections.

To understand how very different Big Data is from old fashioned data, we can break down the analysis into different dimensions. More likely you have already heard about the three more commonly used: Volume, Velocity and Variety.



The 3 Vs of Big Data

Why Volume? This is a heritage term from the Internet and Mobile era. Global access to information for all people around the world, more systems inter-connected, faster and bigger, scalable platforms that can serve millions of clients at the same time – the opportunity to grab, grab, and grab pieces of information to better understand user behavior is huge.

If you have attended a probability theory course, you might remember the Law of Large Numbers , and one that you might have heard about, is the Law of Truly Large Numbers . This is the underlying reason of why massive amounts of data is interesting: for better analytics, accurate insights, event forecasting, and so on, with the help of mathematics.

Would you get a better prediction if you could analyze millions of records instead of hundreds? Would you get a better prediction if you had 100 attributes on those entries rather than just 10? Bottom line, having more data by far exceeds having better models.

What about Velocity? As with Volume, this is a heritage term from the Internet and Mobile era. Products and services are being delivered, and consumed, faster than ever, and today it is possible to track every interaction your users have with them – millions of global users accessing your products at every second not just increment the data volume massively, but the rate at which data from each of those events flows into an organization is constantly increasing as more people are able to reach your offerings.

If you think about it, today it is not the velocity of the incoming data and how to capture it that is the issue, as we have fast SSD devices and scalable streaming platforms that can drop millions chunks of data for later batch analysis into blob storage. The key point here is the feedback loop... How to inspect, and understand, these constant flows of data and react to them faster than competitors. How to come out with the best decision in near-real time to retain and grow your users.

And Variety? I already mentioned a lot of it before starting to discuss the 3 Vs. There are many formats out there, and very rarely data comes to you perfectly ordered and ready for consumption. The origins of the data sources are diverse and most of it does not fit into classical relational structures. Chunks of data can be text from social networks, image data, a raw feed directly from a sensor source. Here is where a very common use of Big Data processing is taking unstructured data and extracting meaning for consumption either by humans or as a structured input to other systems – I will dig deep into this in the next chapters.

So far, I discussed the 3 most common (or classical) dimensions people think about when talking about Big Data – but are there any others we can bring into discussion? Yes of course, and they represent other challenges and opportunities as well.

Let's start with a very important characteristic: Veracity. What is it and how is it related to the classic three? Well, as you might easily figure out, it is inversely proportional to the other three. When one or all of the above properties increase, the confidence and trust you can have in those data sources decreases. This represents a hidden problem that covers biases, misinformation and noise which you should be constantly aware of. At the end of the day, you need to understand and measure the risk associated with business decisions, no matter how big or small are, based on processing them.

What about the quality of your data? This is Validity. It is similar to Veracity, but not quite the same. Here you usually put into practice different cleansing and governance mechanisms to normalize and correct your data for your intended use, with the sole purpose of ensuring data quality. The validity of your data sources must be accurate before giving them as input to different analysis and decision-making processes. This is an important topic as well, and it is why data scientists spend around 60% of their time to make it right.

If you do a Google search, you will find people talking about many other Vs - Humans tend to do this (and I will talk about patterns later), when we find a nice pattern to describe or identify something we try to find as many cases as possible to validate we are right (in this case, keep finding Big Data attributes, or dimensions, that start with V – closely tied together but with little discrepancies to justify their existence).

Nevertheless, there is one more that got my attention, and it is: Volatility. When does your data expire? Until which point in time you should keep storing it, and most importantly, use it for your business intelligence analytics? This is another important attribute that you should always keep in mind as you won't get accurate, or up-to-date, results if you are playing with stale or old data.

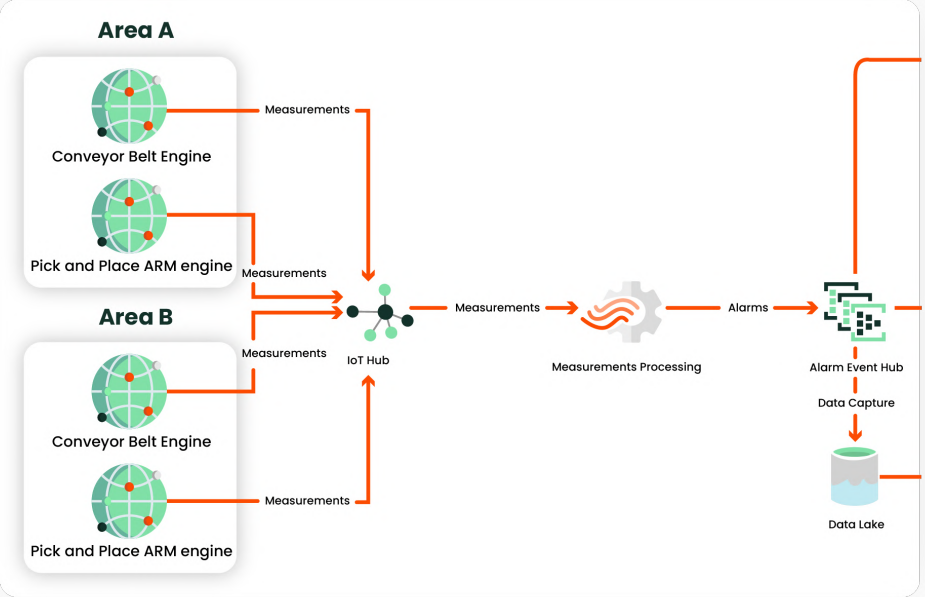
As a result of your Google search, you might also find people talking about Value, Variability, Visualization, Vulnerability, and many more - as long as it starts with V and it can describe a characteristic related with Big Data, it's probably being discussed out there.

A TYPICAL SCENARIO – THE SMART FACTORY

SOUTHWORKS published a scenario that showcases the smart-factory sample mentioned before. The solution simulates and monitors a printed circuit board manufacturing system, and it leverages the Azure stack as a framework to bring this to life.

To execute the simulation, SOUTHWORKS created a console application that emulates the behavior of the sensors and controllers of the printed circuit board manufacturing IoT devices. The solution saves the event stream coming from IoT devices into a data lake and raises alerts when out of bounds conditions are detected.

It is the first part of this solution what you should focus on, understanding how continuous data flows from IoT sensors in a smart factory are handled and stored (depicted below).



Read the complete post here

CUSTOMER STORY – VIRTUAL REALITY IOT SENSORS

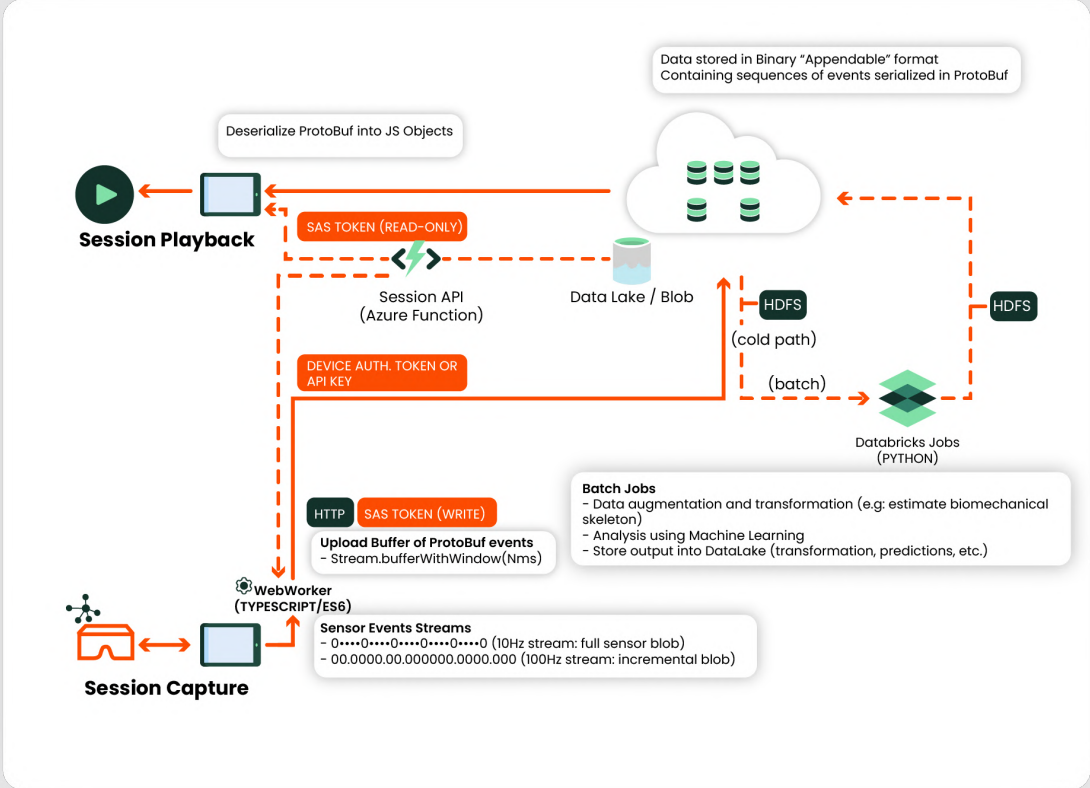
SOUTHWORKS worked with a company that takes physical rehabilitation to the next level by using VR hardware and software to meet the specific therapeutic needs of a wide range of patients and survivors. They combine science-based healthcare expertise and decades of world-class software development experience into a unique combination of cutting-edge healthcare and technology.

This company achieves this amazing UX by placing several sensors within the patient's body to support rehabilitation of the upper body with a focus on strengthening, range of motion, and postural control. All the data generated is lost once used to display patient's avatar into the VR world rendered on the head-set device.

They wanted to improve UX and gain more insights on patients' physical recovery by bringing new capabilities into their platform, including functionality such as letting a doctor playback and analyze a user session, or leveraging complex machine learning and analytics cloud models applied to all data being generated, while also addressing cognitive functions like visuospatial awareness and command-response.

SOUTHWORKS developed a reliable high-performance solution to write, and easily access, huge amounts of data being captured by thousands of sessions, for thousands of patients, and hundreds of hospitals and clinics in a data lake specifically designed to address all different data-access scenarios (from session playback to big data analytics of several sessions to improve their VR-UX).

The solution leveraged the latest breed of Microsoft Azure Data and Serverless Computing solutions such as Azure Functions, Databricks, Data Lake and Analytics services. All these pieces glued together, in addition to specific transmission protocols, data lake schemas, file formats etc, helped provide an architecture that can cope with continuous reads and writes with the fewest possible penalties.



Now, our customer has huge datasets to play with. It brought into the picture a data-science team and, as we speak, they are designing and running new and exciting machine learning models in the cloud.

Its product team is happy to provide new tools to doctors to be able to reproduce patient's sessions - alongside interesting analytics - that helps them to tailor patient's rehabilitation more accurately. Finally, patients got an improved experience as new and more interesting games and interactions can be brought into the VR world to help them recover sooner.

04.

From data to information

Understanding and finding meaning in data

From this point on in your data journey, you will need to engage more actively to continue progressing through the stages from data to information, to knowledge and eventually wisdom. As I will show you during this section, there are many mechanisms and techniques useful to start processing your data at scale. Not just capture it and store it in case you need to review a transaction log, or troubleshoot a defective system, but to start connecting the different pieces of the puzzle to see the whole picture.

Computers 'think' as 1s or 0s – there is no other way, and everything they can handle should follow this axiom. On top of this, we build more complex systems, but everything has a structure, a schema, a set of rules. So, in order to address problems by using computing power we need first to map them into their world.

Humans also think in a structured way – for example, when you study for a test, you might highlight pieces of text to later build a mind map that helps you to remember and associate the concepts that you extract trough text analysis and interpretation; the same thing happens when you deal with something new, you try to find a pattern that brought you good results in the past and apply it. With all these, at the end of the day, you are giving structure to problems.

When we (humans) face something that does not have an order (or structure) we start a matching process trying to identify these patterns I mentioned before. This is the best way we know to understand and find meaning and make decisions. In addition, when we just have a small piece of information, our brains are really fast to perform this process, but when the information becomes massive we need to sit down, grab a glass of water (or beer if you prefer), a piece of paper and start breaking down ideas, concepts, drafting a connected graph with all of them, which at the end will guide you on how to approach and analyze all of that information. It will provide the structure needed to understand the whole thing.

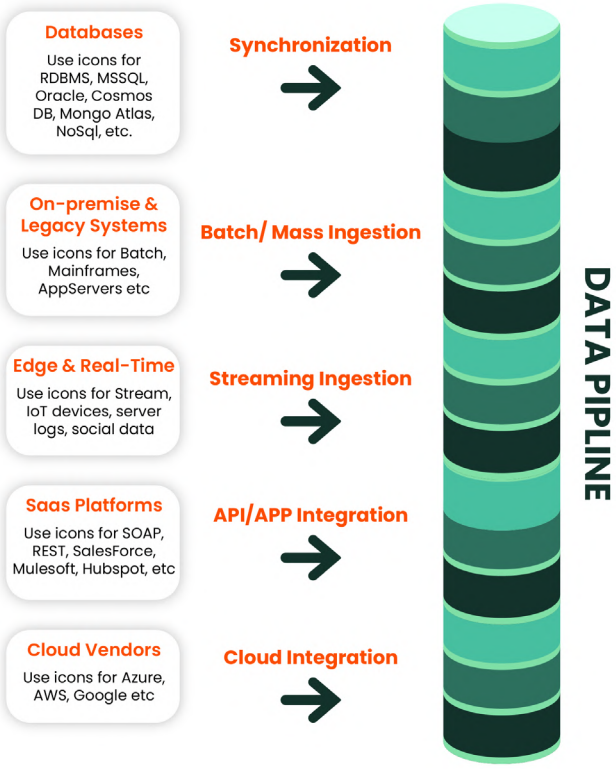
For computers, at least for the time being, to deal with and to understand something without a previous structure is impossible. They need to have it. Of course, at this point, you might be wondering what about unsupervised machine learning techniques? Those are no more than mathematical algorithms that have just one goal: to give structure to something unstructured, and later on, do something else with that outcome. In a very summarized way, you are leveraging computer's power to help you take the problem into its world of 0s and 1s – your data, or subsets of it, has certain attributes or not, it belongs to one set or another, etc.

During this chapter, I will explore several techniques and concepts that apply to Big Data. From the different approaches you can use to extract it from its source, to how you store it in the smarter way to fulfill different use cases with efficiency and performance.

EXPLORATORY ANALYSIS

As I mentioned in the previous section, today your organization (and different departments within it) consume data in several ways whether it may come from the same source or not. This is process that you might constantly execute to extract raw data and convert it to some meaningful piece of information.

Today you probably hear a lot about Data Pipelines and ETL (Extract, Transform and Load), which most of the time are used indistinctly but are not the same. A data pipeline basically describes a series of processes you use to move data from one place to another, from storage to storage, to a system, or from system to system. There are many types of data pipelines, and ETL might be (if required) just a subset or piece of it. But whether you are putting in place a set of processes (or steps) to manipulate your data in order to perform quality checks and/or normalize it, to match and merge it creating a master record or profile, to mask it, to anonymize it and/or to encrypt it for security reasons, or even to integrate data from multiple sources before correlating them, you are building a data pipeline.



Data Pipeline Patterns

Talking about patterns once more, we do have Data Pipeline Patterns here as well. So, which are they? They are not fixed to a specific number, as you can always break them down into smaller categories, but avoiding doing a fine category drilldown here, I'll group all possible types of data pipeline patterns in the following 5 categories:

1. **Replication and synchronization**
2. **Batch or mass ingestion**
3. **Stream ingestion**
4. **API and application integration**
5. **Cloud data integration**

So, let's talk a little bit about each one of them and where you would use them.

1.

Replication and synchronization.

This is a process most commonly used by RDBMS (Relation Database Management Systems) to maintain consistency between two or more data sets, but it is not a process that you can only encounter within this kind of system exclusively. Some systems have one option or the other, some have both, and they are basically different depending on how often they run, if they are automated or not, if they are synchronous or asynchronous, if they have a one or two-way data flow channel, etc.

In the aforementioned RDBMS world (nowadays also within No-SQL platforms), synchronization is often a process that runs constantly (depending on your DB engine capabilities and its configuration) and ensures synchronizing data between two or more servers, updating changes automatically, and maintaining consistency between them. This, among other benefits like scalability, provides a fault-tolerance solution that can keep your business up and running even in the case one server is down or offline. Many RDBMS also have the concept of replicas – but it is closely tied to synchronization, unidirectional or not, likely having different updating frequencies, etc. But it is not only within this scenario where this kind of data pipeline is implemented - or configured as most of the times it is provided by the RDBMS (or modern DB) itself.

Either way, data replication and synchronization are crucial for integrating big systems. One way or another, different data sources need to be in sync with, and multiple available to, other systems that rely on them for report generation, business analytics or decision making. Many industries and areas like sales team productivity, invoicing accuracy, logistics and transportation, etc. cannot afford having sync data conflicts, as they will result in costly operation errors and poor data quality related to Big Data dimensions I mentioned in the previous chapter.

There are a number of ways in which synchronization might happen: when changes are applied exactly in the same order they are generated by following a log, this is the transactional approach. When replicatin or having such a log in all replicas is not possible, changes are applied as an aggregate, and in this case a snapshot does the job – keeping the whole history only at the original source. Lastly, when changes are occurring on both sides a merge is required (if neither side is declared as master), and it is the most complex and costly approach for all the instances to have an up-to-date dataset with all latest changes.

2.

Batch or mass ingestion.

This type of data pipelines has one, or several, processes that are executed to periodically collect big (or any size) data sources and store them within a storage system (its destination). Common samples of this pipeline type are closely related to ETL operations, where big

chunks of data are extracted from source and manipulated before being appended to a final destination, and its related but different brother born with the rise of Big Data and Massive Parallel Processing Platforms (MPP) which is ELT – whether you don't care, or still know, what will be the format or structure of the information you are storing at the time being, but will rely on powerful processing power to later consume or transform the data (also related with schema-on-read type of solutions that I'll be talking ahead) for your future needs.

There is not only ETL (or even ELT) where you can find this pattern. Let's say, for example, that you can put together a batch pipeline to achieve data quality and correlation which normalizes profile information for new customers every hour. This type of pipeline might run at the mentioned pace, collecting and correlating information from different sources in order to build your customer-360 database for a unified customer view that you can exploit better within different platforms of your business for cross and upselling.

3.

Stream ingestion.

This type of data pipelines come up from the need to process data sequentially, in order, as it is being dropped into the pipe – it arises from scenarios where dynamic data is generated continuously, by thousands of data sources dispatching records of very small sizes, and you cannot wait or delay taking a proper action until all the information is batch-analyzed; here you need to respond in near real-time!

Let's take the simple example of online retailers. They are able to capture every click and interaction their customers have within their platforms and create large histories of users' behaviors, not only sale purchases. If they can quickly analyze that information on the fly, instead of just storing it for later purposes, they will gain a competitive advantage in their market. As I mentioned in the previous chapter, the boom of Internet and mobile devices over the last decade means that almost every customer has a smartphone at hand streaming geolocation info, images and even audio data if you want to.
With the rise of IoT (Internet of Things), this constant data inflow grew exponentially over recent years, and more and more platforms are developed in order to cope with the demand. These platforms are able to scale up as needed with just a simple click, or better automatically, and are able to inspect on a record-by-record basis or over a sliding time window basis, the constant feedback being received for different types of insights, from filtering to aggregations and correlations to mention a few.

At the end of the day, you have valuable real-time information from customers and autonomous devices, and from it you can generate near real-time reports, or execute actions like raising an alert when indicators are outlying normal behavior. And not only from one system or application, you can also collect, mix and analyze customer information from your web facing apps, your e-commerce platform, customers' social network trends and status to apply more sophisticated algorithms like using machine learning trained models to extract better insights that you can leverage to engage the customer more and more with your products.

4.

API and application integration.

This category explores the integration of third-party tools or custom APIs within your ecosystem. Usually you will use data adapters, or data connectors, that might be provided by the platform's vendors itself or developed by another third-party.

The patterns used within this category are closely related with real-time solutions but are not quite stream ingestion. When an event occurs in one application, this one might need to inform another system about it or even ask it for information it needs. Nevertheless, in some cases these updates can happen in batches, so it is not exclusively tied to live-updating.

The challenges within this category do not rest only on your side of the equation, but most of the time on the integrations you need to achieve with your business partners. You might be prepared to digitalize and share all of your business data through an extensive set of APIs, but your partners might be a little bit behind. Additionally, different protocols and communication channels here might bring another level of complexity to it, from old data exchange protocols like the Electronic Data Interchange (EDI), passing through Simple Object Access Protocol (SOAP) some years ago, to the more recent Representational State Transfer (REST). Keeping up to date with technology trends is important, you will achieve smoother and faster integrations with your partners when everyone is surfing the same wave. Finally, and closely tied to this, a carefully thought-out API design and API Management strategy are a must!

5.

Cloud data integration.

This category explores the patterns that are used to integrate data from different vendors, between public or private clouds or hybrid approaches with cloud and on-premises networks. To efficiently access relevant data stores in a transparent way, through specific processes and applications, different mechanisms and patterns are put in place. Involved clouds might be integrated by using different approaches like SSL connections, HTTP tunnels, VPN, etc. each one of them requiring provisioning and configuring different security protocols.

With the constant increase of hybrid and multi-cloud solutions, nowadays this is required more than ever. According to a Gartner survey, 81% of public cloud users leverage more than one cloud provider. Companies are increasingly generating or consuming external data sets (by batch or streaming mechanisms) as sources of significant information and they are quickly noticing the huge value found in integrating these with their existing operational systems within hybrid cloud environments.

With the goal of having more benefits for your customers (not to create a provisioning and operational burden), more companies are choosing this approach as it helps mitigating the risks of a cloud or region outage, avoids vendor lock-ins, provides lower latency as you can deploy to regions closer to your customers, and more.

There are key aspects among these kinds of solutions that you should always keep on the radar. Within these heterogeneous environments data validation and accuracy are very important topics to bring into consideration while continuous data flows happen from one cloud to others. Having a robust implementation towards the mentioned data quality topics, along with monitoring and proper management, gives peace of mind to analysts (or anyone) consuming data from the repository where you store it (data lake, data warehouse, or any other). At the end of the day, and more importantly, this will help you build trust with your customers – who are also data consumers.

As a short summary of Data Pipelines, everyone benefits from clean, up-to-date, trustable synced data. From business analysts from within your organization that can use up-to-date information (in real-time in some cases, and multiple accessible from different regions or even worldwide), to final customers that receive product suggestions faster than ever and more compelling services that exceed their specific needs.

Within your own industry, decision makers can rely on the latest data to take important strategy decisions or gain a competitive advantage against direct competitors. In use cases in different industries like finance and retail, stockholders can stay on top of the latest trends aligned with their business interests, while manufacturers and providers in a big supply chain have the most recent updates impacting directly on production, and a distributor gets the most recent product and inventory availability to make the best marketing campaigns and offerings.

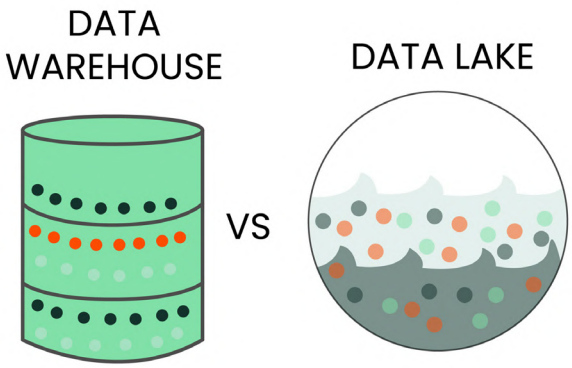
As mentioned before, the purpose of the ETL Pipelines is to find the right data by extracting it from one or many data sources, then transforming it for further processing later on, and finally load into a destination that allows easy access and analysis. Under the hood ETL Pipelines are composed by the following steps:

Extract the data from the origin (transfer)
Transform it as needed for capture (not required for ELT)
(transfer) Load it to storage destination

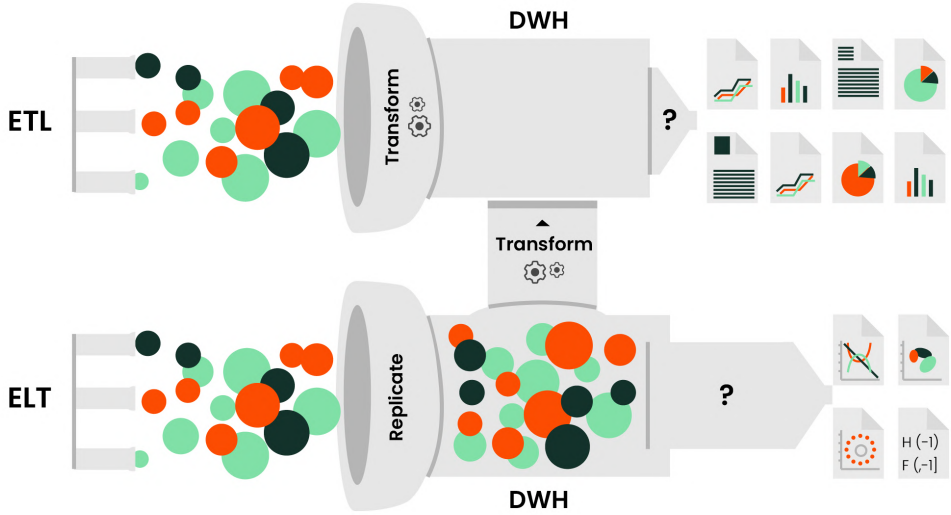
On the other side, there are ELT Pipelines where there is no need to transform the data. These types of pipelines are built to extract data from the origin and capture it raw at the destination. It is a problem (or not) of the application, process or business intelligence platform that will consume that data to deal with it.

Data origins can be, for example, business systems, APIs, marketing tools, application server logs, transactions within databases, etc. and the destination can be a database, a data warehouse, a data lake, or even particular cloud-based solutions from different cloud providers like Azure Synapse, Amazon RedShift, Google BigQuery, etc.

So, what's the difference among these repositories and platforms? What's a data warehouse? And a data lake? When to use one or another, or both?



Data Warehouses and Data Lakes



ETL and ELT Pipelines

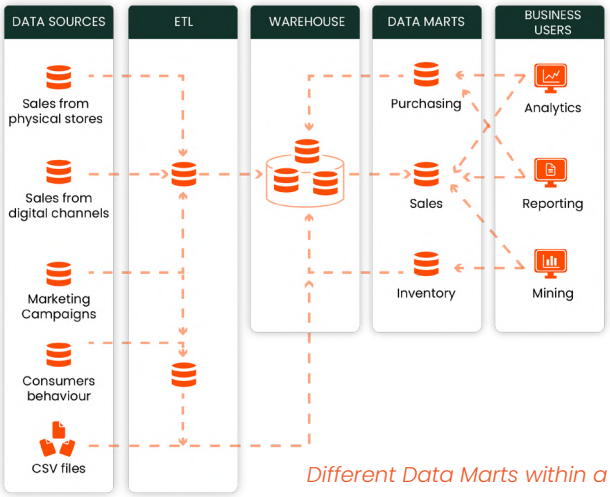
You can consider a data warehouse as the next step for databases, a storage location where you concentrate large amounts of from many different sources. In addition, they are popular within mid/large business organizations as a way to share data across teams and departments, going beyond the data silos each one of them have within their own databases.

But data warehouses have specific structures, or access patterns, as it can be the data mart. These are subsets of the data warehouse that are usually oriented to one specific business line or team, although in some cases it can be used by many. They are basically a condensed and more focused version of a data warehouse that reflects the regulations and process specifications of each business unit within an organization.

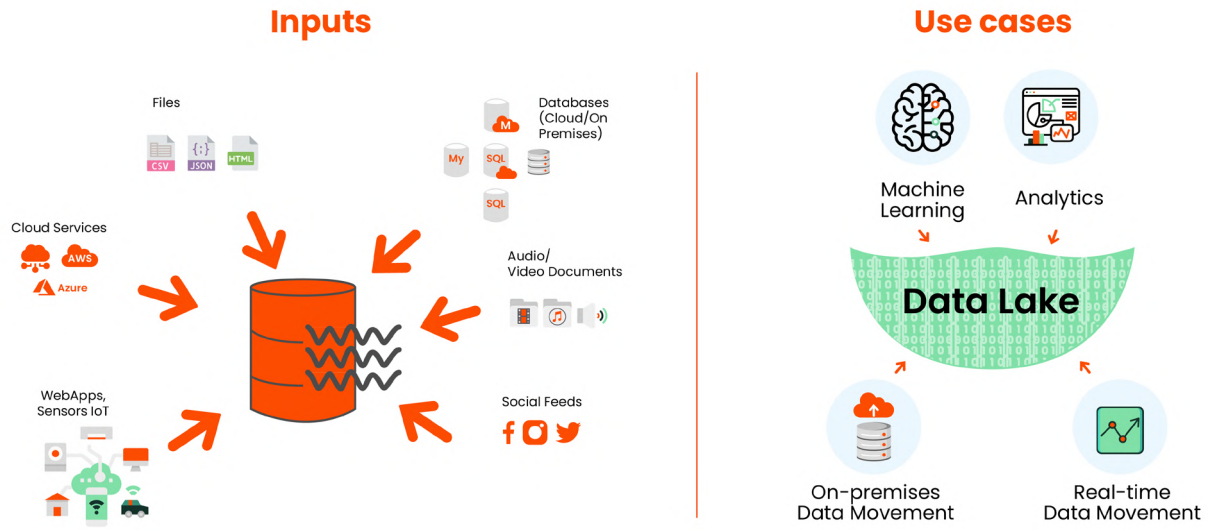
A data mart may belong to one department within your organization which univocally identifies who manipulates and develops its data, although it might be shared and consumed by many other departments in some cases. Different from transactional databases, this data marts are considered to be read only subsets designed to provide access to large groups of related records – improving this way the response time and providing to business users a performant solution for reporting, business analytics and insights.

These (data warehouses and data marts) are not new concepts at all, for decades organizations have stored their data for business intelligence purposes within these platforms, although, as they require specific data structures (schemas), they narrow the type of data analysis you can run on-top. So, you can only use them to run specific sets of data analytics processes and take business decisions leveraging these tools, but you lack the flexibility of running a whole different set of processes as your data might not already be prepared for.

Data lakes come to solve the biggest limitation I mentioned for data warehouses: their flexibility. A data lake is capable of storing huge amounts of data in their raw form, it does not need a structure or predefined schema in order to write or read from it. Same as before, you can use them to run specific sets of data analytics process, but the use cases might be a little different. You use a data lake to experiment, to bring in data scientists and research/test on hypothesis you want to validate. Later on, once you have a desired outcome, you might consider moving the data required to replicate this outcome to a data warehouse where your data would be better structured for this specific result, and you will get be able to come to the same conclusion faster.



Different Data Marts within a Warehouse

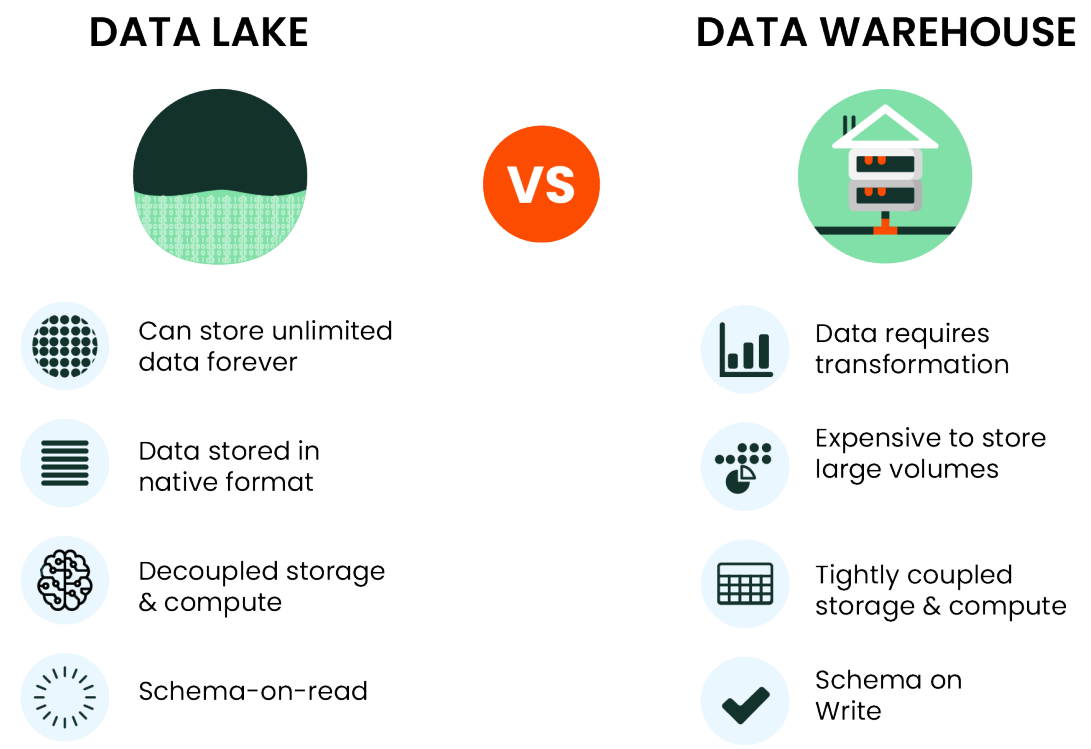


Data Lake Information & Use Cases

In addition, and when dealing with Big Data, it is important to remember sizes. Storage cost for a data lake is much cheaper than what you would pay to put everything in your data warehouse (or even DB). Whether you want to experiment on your data by keeping everything that is generated within your platforms (and external ones), or you need to keep the whole history of user transactions and logs over the years for security and compliance reasons, the amount of money you will end up paying will be significantly less than with a traditional RDBMS or data warehouse system where compute and license costs will sky-rocket - as data lakes decouple storage and compute there is no need of processing power to transform/store data using a particular structure or schema, and you can optimize query and read processes according to the frequency you need to access it. At the end of the day, you have a read/write trade-off in order to come up with the total cost, but at least you have the flexibility to manage each one of these variables independently.

What else? As data lakes do not enforce any structure, they can cope with the high-velocity at which data flows today, and due to their low cost, you can keep everything – there is no need for any type of filtering to decide what to discard. They are a good destination for streaming data from many different sources as they can store each piece of it as objects (in most cases represented by JSON files, although some can be directly binary) with no need of having a structured record.

As I mentioned before, they are used for experimental use cases through exploratory techniques and data preparation. For example, to build machine learning models that can be applied later to automate some business decisions and drive processes. Without entering too deep into how machine learning techniques and algorithms work, but to highlight something of significance - all the data in a raw format is there – a structure or schema enforced by a data warehouse might not be a good fit for applying these techniques as the imposed structure might impede a proper analysis.



Data Warehouses vs Data Lakes

I've started this chapter talking about structure – it is needed prior to every analytical process that you want to perform or to automate on a data set. In the data world this is often referred as a schema, and in order to be able to analyze data sources for whatever reasons machines need it.

Now the question is: where do you bring the schema into the picture? How are different platforms flexible towards this problem and how can you leverage them to achieve the best results?

Something that is closely related to relation database management systems is defining a schema for your tables before you can insert any type of data into the system – although we know that raw data from one or several data sources is very seldom structured. Therefore, while ingesting data into the database by running a big ETL pipeline or inserting it piece by piece, you must not only need to have defined the database schema in advance, but you must also transform the data to fit that schema – this is known as schema-on-write.

Depending on your purpose and intended outcome, this may, or may not be something useful. Either way, loading the data into the final destination is a slow process, as you need to transform it and give it the structure required by the schema where you want to store it. However, as you might perform a data cleansing step within the transformation, your data will contain fewer errors and be more reliable.

Business intelligence platforms are able to access the data very easily, knowing its 'contract', and very fast for consumption as data has structure. But what if you made design decision errors in your schema, or your data sources changed? Updating a database schema, and migrating all the data it already has, it is not a simple task per many factors - and it gets worse by volume.

Data lakes allow you to 'throw-in' raw data without taking care of the origins, nor whether the data have a common type or not, and not requiring a schema at all. Same as before, let's say you fill up for data lake by running a big ELT pipeline or you add data piece by piece, in this case you don't need (at least while ingesting the data) to care about respecting any structure or schema at all. You just capture everything for experimental, or not yet defined use cases, you want to perform later on. Although later on, when you want to use that information for any reason, you need to provide a structure to the data for any other system to deal with it – this scenario is known as schema-on-read.

Here you only apply a schema when you read the data. With the growth of computer power and storage capacity over the last decade, more and more platforms are now available to help you cope with this scenario without huge performance penalties – of course access to your data won't ever be as fast as with a platform that already has a defined structure (the schema, but you gain much more in terms of flexibility when you want to explore your data in many different ways.
In summary, when to use one or the other will depend on your use cases and the application

Schema-on-Read vs Schema-on-Write

Category	Schema On-Write	Schema On-Read	Related technologies
Adaptability	Structured	Semi-structured / Unstructured	Storing structured data requires better planning, and it is linked to SQL-like tools and systems. You need to know exactly what you are looking for prior to start ingesting big amounts of data into the destination. On the other side, you don't need to plan in advance, it is a better approach to get to know and understand your data prior to decide how you will leverage it and what decisions are going to be driven by it. It is the suggested approach when you need to explore your data with no predefined questions.
Consumption	Slower loads	Fast loads	A well-defined schema requires the use of heavy ETL tools and pipelines to collect and move data to data warehouses in order to be easily consumable by OLTP & OLAP systems. On the other side, there is no need for tailored nor complex ETL pipelines, data can be collected as it is being generated (or extracted) pretty much in raw formats.
Formatting	Fast reads	Slower reads	OLTP & OLAP systems are super-efficient and performant to deal with big amounts of data when the schema is well-defined in advance. On the other side, Big Data platforms like Hadoop, Spark and No SQL systems brought new features like data locality to perform better like HPC classic models, but data access will still have a penalty when non-structured.
Ingestion	Not flexible	Very flexible	When the schema is well-defined in advance, it enforces the way you process and store your data, but a schema change is more difficult due to the constraints set over the model, needs to migration (if any), etc. On the other side, the schema is being generated on the fly by the platform accessing your data. As mentioned before, this process is much slower but can be adapted dynamically to provide other representations of your data.
Validation	Fewer errors	More errors	An implicit data validation step is enforced by the underlying schema. You might, or not, have a well-defined ingestion pipeline where you perform data cleansing, but with a well-defined schema you have another level to enforce the data being ingested is-as expected. On the other side, client applications (or users) are allowed to upload any kind of non-structured data, which might no be valid at all for your final purposes. You need to have in place in this case data cleansing and validation pipelines to ensure what you are going to consume to make decision is accurate.

of your data. While considering ingestion times within ETL or ELT pipelines you see that schema-on-read is much faster than the schema-on-write process, there is no need to take of what and how you capture it – and it has the capacity to be up to speed for high-velocity data streaming. It is suitable for massive volumes of unstructured data, and better for experimental analytics and data exploration as it does not enforce a rigid schema in advance.

On the other side, when using schema-on-write, you should know the schema in advance, define it and transform your data to align to it in order to be ingested. This transformation step results in a slower process that is not suitable for capturing data at high-rates and volumes. However, if you need to run really efficient and fast business intelligence tasks to get near real-time information in order to make decisions, you should not discard this traditional approach.

Although I said that in a data lake, by following the schema-on-read pattern, you can throw in and, later on, deal with any piece of data you store, there are a number of Big Data file formats out there to make your life easier. These data file formats are your allies when dealing with mainly 2 of the 3 classical Vs of Big Data – Volume and Velocity.

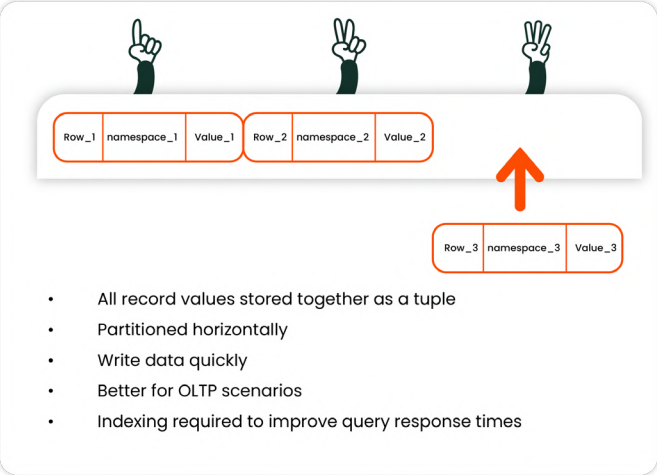
When writing or consuming information at high-velocity, and in big volumes, understanding and making good use of these file formats (based on your use cases) is a must! These will allow you to run performant analytics that can bring you the insights you are looking for at the times you need them - sooner than your competition, ready for your next business decision, closing the feedback loop with your customers faster, etc.

But at the end of the day, how you store the data in your data lake is very important if you want to have simple ways to traverse it later and good performance. You need to consider variables like file formats, their ability to store compressed data and design how you will partition the data – finally, by using them you provide some minimal schema to your data.

There are many data file formats out there, and as the addition of new vendors and technologies grow, more and more are being proposed and adopted. Today we have a mix of some old well-known formats like CSV and JSON, but there are others more recent like Avro, Parquet and ORC to mention most spread ones.

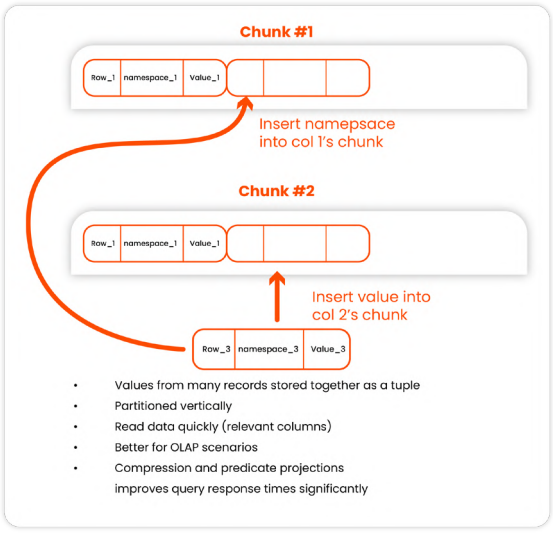
All mentioned data file formats have an underlying structure for each piece of data appended to them following basically one of these two patterns: row-based or column-based. At the simplest level row-based models are great for transaction processing and writing data to them at fast-pace. Nevertheless, they are not great for retrieving queries that have projections and predicates over some of their record's attributes. Bottom line, row-based models are suitable for intense writing and/or updating scenarios.

Row-based data storage

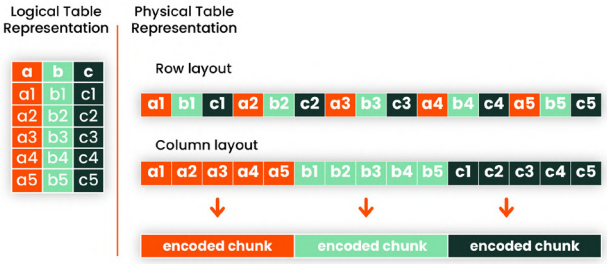


On the other side of the picture, a column-based model is great to handle aggregations of large volumes of data as subsets (columns). These models are great for highly analytical query processes. As a benefit, column-based storage can solve complex queries in just seconds. Bottom line, column-based models are a good fit for some intense reading scenarios, although you need to be sure your queries are really suited to it (a full-scan wouldn't be a good example).

Column-based data storage

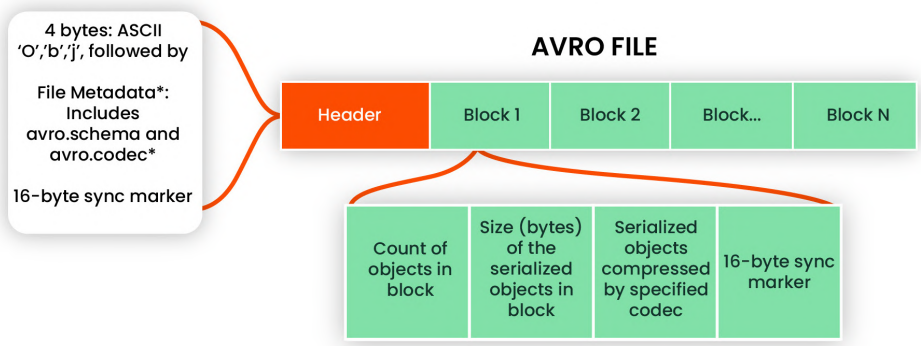


As you have seen, there are some clear differences on how the data is stored within each of these models. Although the logical representation of your data will be the same for both (in most cases), it is important that you carefully and wisely make the best decision on how you will store and manipulate your data. Most of the time the "one format to rule them all" does not work here, and you should design your data pipeline as plug and play pieces that can fulfill each of the different use cases you will have for ingestion and consumption. Understanding the inner working of the data file formats I'm discussing here, as well as all the platforms that make use of them is your best ally in achieving storage efficiency, operational performance and faster insights and results.



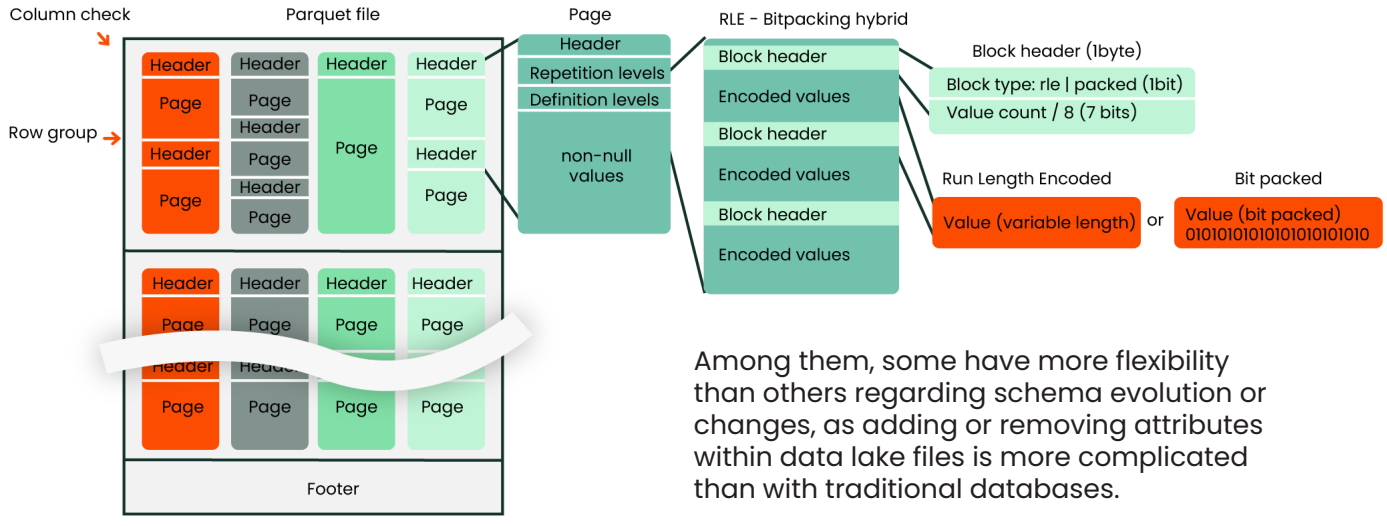
Row vs Column Based comparison

There are many interesting attributes to take into account in order to decide which format (or formats) applies better for your use cases. Each one of them have their own internal structure (a schema) to organize the data you capture inside. Some formats like JSON, Avro and Parquet can handle nested data while others cannot, although for most cases Avro is recommended over Parquet which handles this inefficiently.



Avro file format

Performance is also an important factor and depending on the use case you might need to choose one format or another. For example, Parquet (column-based) is great when you want to query your data lake using a MPP platform with SQL-like clauses, while Avro (row-based) is better for ETL ingestion performing faster row level appends. In addition to the storage type already mentioned, it is interesting to remark their ability to split one logical file into many physical ones – this also impacts performance, and based on how you organize your partition, you will achieve the best performance on your read or write use case.



Parquet file format

Among them, some have more flexibility than others regarding schema evolution or changes, as adding or removing attributes within data lake files is more complicated than with traditional databases.

Support for different data types within them is also important. Values encoded in binary form require much less storage than in their text form. Let's say you want to store a simple integer, 1234 requires only 2 bytes in binary format, while the string "1234" uses 4 bytes of storage.

Last but not least other attributes like compression, human readability and compatibility are also considered based on the use you want to give to the data stored within them.

Without entering into much detail on their underlying structure, use cases and inner workings, below is a summarized comparison table that can give you a better idea whether one or the other can provide the best results for your use case:

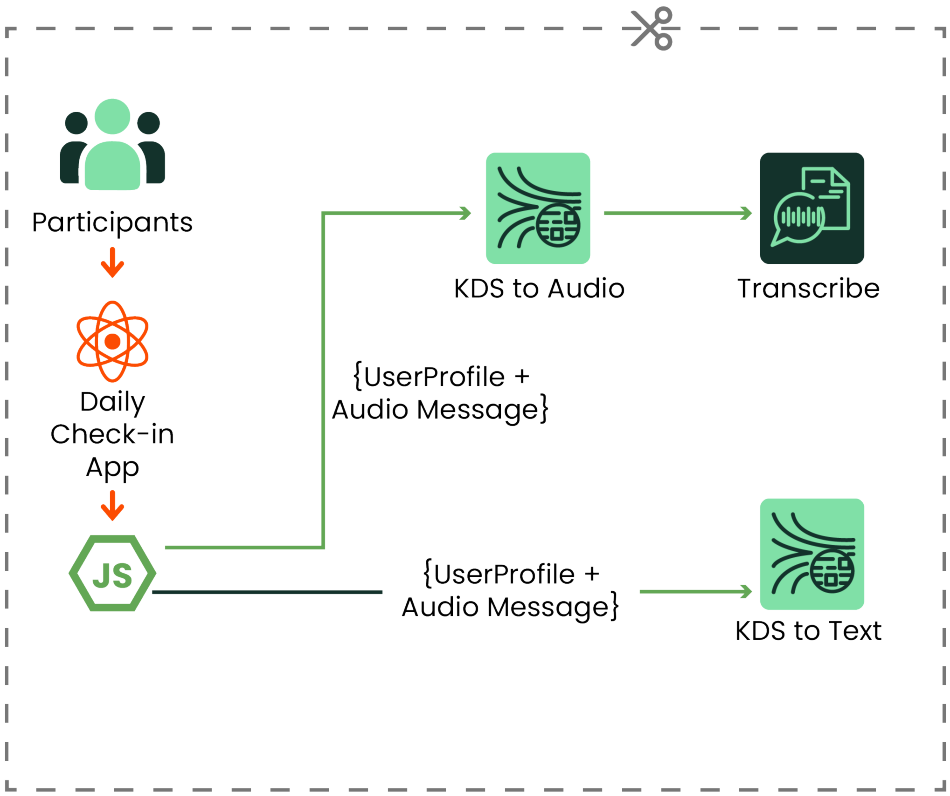
	CSV	JSON	Avro	Paruet	JSON
Nested Data	No	Yes	Best	Good	Better
Storage Type	Row	Row	Row	Column	Column
Splitability	Yes*	Yes*	Good	Good	Best
Performance	Low	Low	Write	Read	Read
Schema Evolution	No	No	Best	Good	Better
Data Types Support	No	Yes	Yes	Yes	Yes
Compression	Yes	Yes	Good	Better	Best
Human Readability	Text	Text	JSON + Binary	Binary	Binary
Batch usage	Yes	Yes	Yes	Yes	Yes
Stream usage	Yes	Yes	Yes	Yes	Yes

A TYPICAL SCENARIO – SENTIMENT FROM RAW TEXT/AUDIO

SOUTHWORKS published a scenario, and a sample application, that showcases a pipeline that is capable of ingesting huge amounts of data incoming from its participants. The solution allows participants to submit a text or audio, at their own pace, describing how they feel. A machine learning trained model process each piece of data submitted by these users and convert it to useful information that determines if someone is 'happy' or 'sad' at the time being.

The whole scenario is far more complex than what's described above, and it is part of a series of post that showcases many other things, although the part I would like you to focus on is related to data pipelines mentioned within this chapter.

The depicted piece below directly shows how a streaming data ingestion architecture can be implemented within AWS and how you can scale up the processing piece of the pipeline in order to cope with high load (think of it as a Twitter with millions of users uploading text at their will).



Read the full story here

CUSTOMER STORY – DATA INGESTION AND PROCESSING PIPELINE

Our customer, a big ticketing platform in LatAm, turned to us with specific problems in the way they generated their business and marketing metrics - They were obtaining their business KPIs by generating manual reports on demand on a weekly basis.
The different business departments were required to frequently ask for database exports from different platforms. The technical team addressed those requests, and with help from the marketing and sales departments they crunched the data to generate very basic excel reports. Each of these reports were generated per dimension, some were 'event title', 'organizer', etc. A tedious, error-prone, and time-consuming process just to have a very basic idea of the business pulse.

So, they turned to SOUTHWORKS.
In less than a month, our Fireteam helped them to instrument their top-3 customer facing applications leveraging customer tracking information they did not have before, while working on the correlation of those user activities and information from their marketing campaigns and sales transactions to simplify the process of generating all (old and new) business metrics automatically. All by ingesting live metrics being generated by the instrumented sites into Azure Synapse Analytics, and data sources provided by other departments as marketing and sales.

Finally, two different Power BI reports with key indicators to monitor the business from the Sales & Marketing point of view were created.

Now, our customer does not need to wait a whole week, or perform a tedious process of going gather information from different departments, involving several members of their staff, using different platforms, and crunch all the data manually to visualize numbers in an excel sheet. In just the tap of a button, they have the power of on-demand metrics – and with the dashboards created according to their business requirements, their metrics look fancier and more consumable than ever.

05.

An additional effort required to obtain knowledge

FROM DATA TO KNOWLEDGE

In the previous chapters I explored how you handle small or big volumes of data, how you put it at your service in order to start the journey of converting it into something meaningful, to take the first step into having information from it prior to move forward.

But what's information? As you might already have guessed through the exploratory analysis done in the previous chapter, it is a set of data relevant to your organization at a point in time and under a determined context. You have it once you have cleaned the data, something that you do often during the transformation step of an ETL pipeline (or later on, if you take the ELT approach) – it should be something easy to analyze, process and visualize for a specific purpose in a structured meaningful way. It is used to solve uncertainty.

It can go from attributes that define an entity, to a collection of data points obtained by combining different sets of data, always ensuring that what you have is valid for future purposes. Let's take, for example, the following scenario: you have one device (or camera) that captures the spatial location of 50,000 data points in a three-dimensional space and informs you the color of each of those data points. What if you take not one sample, but several snapshots at different intervals? Do the data points remain static or not? For example, they can represent a tree, depending on how you establish relationships between them, but they can also mean something totally different, or they might have not connection at all. Then let's suppose that some data points are not exactly at the same X, Y, Z they were before – are those the same data points? Would it not be good to have snapshots of the wind conditions at the same intervals?

So... how and when are you able to solve this riddle?

A theoretical description of knowledge goes: "Knowledge is a familiarity, awareness, or understanding of someone or something, such as facts (descriptive knowledge), skills (procedural knowledge), or objects (acquaintance knowledge)."

But, how do we achieve it? We need to understand the purpose and meaning of every little piece of information collected as a whole, finding ways on determining how more or less relevant they are towards our goals.

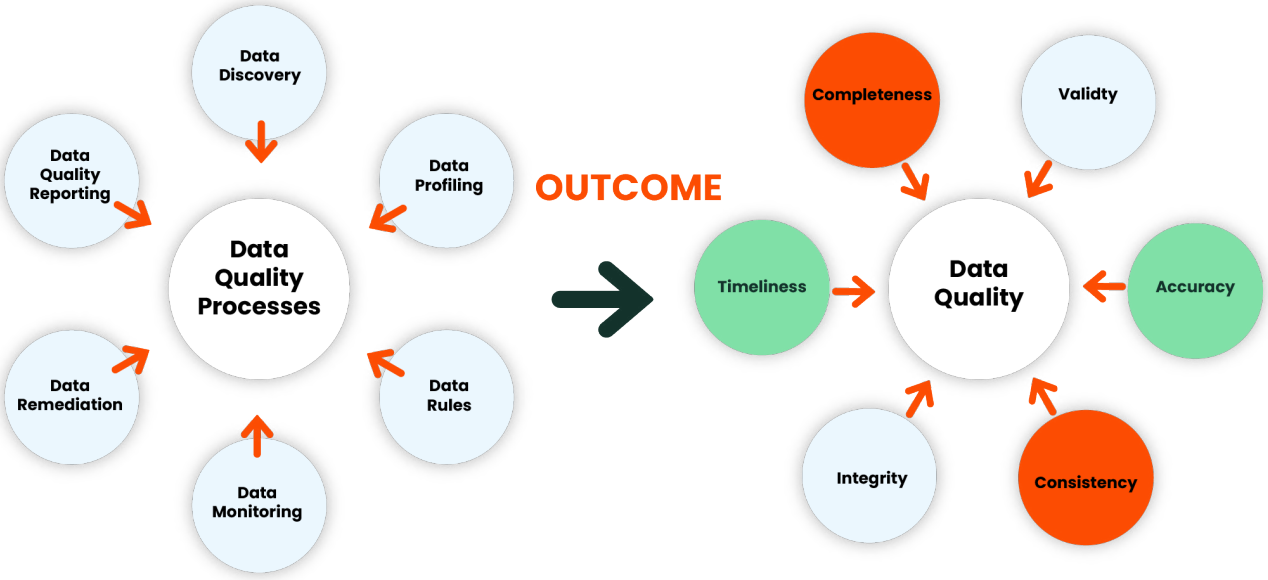
This is an important step to get the ROI from your data integration project. Let's say you built one or several data pipelines to capture information of every user interaction with your platform, and you are on the way to build a customer-360 view - but what's next? How do you correlate that information with other data sources to see whether you should allocate advertising budget? You might be using Facebook and Google, and already know the cost of one or the other, but which one is more effective? From which platform do you get most friend referrals? You already have all the information to answer these questions at hand scattered among several datasets, but to correlate and properly segment result sets for different marketing campaigns and budget-allocation decisions requires an extra effort and understanding.

During this chapter, I will explore more on the topic. From the different approaches you have at your disposal to manipulate and correlate big datasets, to modeling and data exploration and visualization.

EXPLORATORY ANALYSIS

First things first. On your way to obtaining knowledge from your data, it is a must that you can trust it! So, it is extremely important you put together a step, or set of steps, to ensure the quality of it – to conduct a replicable process to achieve outstanding levels of quality assurance.

Once you get into this endeavor, it is a continuous process that has its own lifecycle, indicators, and result metrics. You usually kick it off by performing a data exploration or discovery process where you gather and organize metadata from your data for what is coming next. You often check for data validity here, whether all values for attributes are within possible ranges defined by your business domain.



The Data Quality Lifecycle & Dimensions

Later on, it is very important to know whether (or not) you have all the information that you need to get the results you are looking for – this is one of the data quality dimensions known as completeness. This step is known as the data profiling step, where you carefully examine the data you have at hand with that dimension, and the accuracy dimension in mind. The later referring to whether your data reflects the real-world and comes from verifiable sources - inaccurate information can lead to severe consequences.

The set of rules you put into motion to ensure the consistency of your data is crucial to avoid mismatching pieces of data between systems and processes. These rules also help to maintain data integrity ensuring that the different relationships within each piece of information you have at hand is correct. What would the result of running a correlation pipeline with a dataset lacking these two dimensions be? As part of the lifecycle of a good quality assurance process, you need to constantly monitor that these rules are not bent or skipped by any means.

The last two steps involve fixing or correcting all data issues that you might have encountered along the way. The data remediation step is used to automatically or manually take care of any exception that you have found – whether any of the indicators for a particular data quality dimension is below a minimum acceptable threshold or not acceptable under any circumstance.

And last but not least, you use the last step to learn and improve the whole data quality lifecycle. During the reporting step, the last one in the pipeline, you fill up scorecards with all data quality findings and, later on, visualize them using dashboards to help you tweak here or there to execute more efficiently next time – by increasing the quality level with new rules and analytics that were missing, or achieving much more in less time but without neglecting on quality.

At the end of the data quality lifecycle, you need to ensure that your data is available to your business intelligence processes at any time they need it, and that it is up to date. This is an important data quality characteristic known as timeliness – obsolete or stale information can lead you into making wrong decisions for your business and your customers. This is the whole purpose of an efficient and performant data quality pipeline that helps you drive your business without any time penalties.

In this and the next chapter, I will discuss data science, which among its different alternatives today, it is better known for advanced statistical and machine learning techniques. But let's not forget something that is crucial in your journey, which is exploratory data analysis (EDA).

This classical approach helps you to quickly establish a relationship with your data – existent and new. The different set of techniques used here focus on helping you to better understand datasets, identifying common patterns, outliers if any, for future use cases, and while getting more acquainted with them, understand their past, present, and future purposes. As an outcome of this, you will have a roadmap ahead with questions and hypothesis you would like to validate in order to define what insights you can extract from them. Nowadays, EDA has become a mandatory phase in every data science project, and it is closely related with all data quality assurance concepts I have mentioned before.

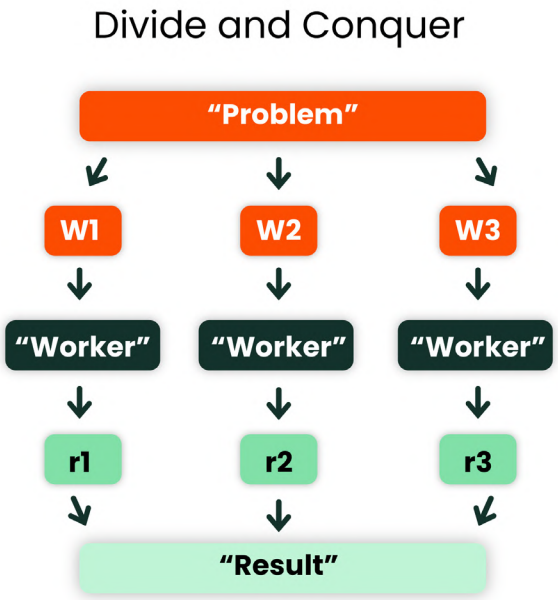
From its beginnings, and still today, the simple statistical analysis you do over datasets within the exploration steps rely on many graphical tools like histograms and different data points plots to learn about how your datasets 'behave' – the final goal before moving forward is to understand data trends, correlations if any, what type of the distribution they might have, variance, outliers, and many other features. Moreover, today, with the aid of visual interactive tools, this is a must that one cannot skip.

Not to extend much here about this topic, but to give you more detail and visual samples of what I'm talking about, let me refer you to this post I published with SOUTHWORKS to analyze different types of COVID-19 datasets to understand the impact of different social behaviors during the pandemic outbreak. There are many techniques that are associated with EDA, and through the post we tackle some of them, such as data types and variables identification and validation, missing values and outliers' treatment, dimensionality reduction, and correlation analysis to mention a few.

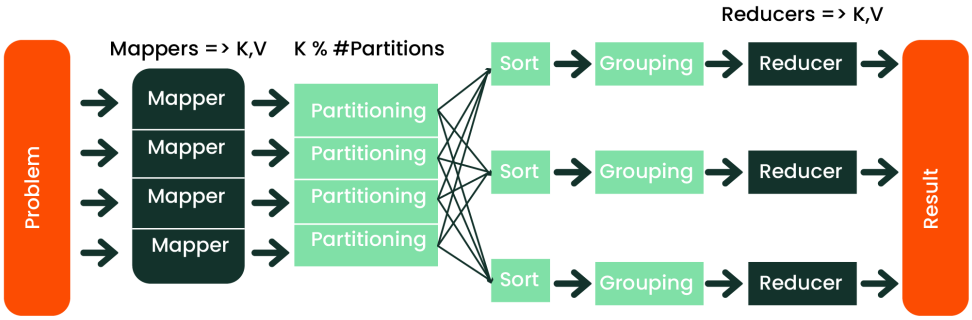
Today, with computing power and modern analytics tools, interactive data exploration is an easy to kick and easy to engage process. A must within every data project you start in order to discover unexpected value in large amounts of complex data – in the sample I shared with you before, we use two Massive Parallel Processing platforms to manipulate data on-the-fly, and a powerful graphical platform for interactive data visualization.

These Massive Parallel Processing platforms (MPP) are not a totally new concept, for decades there were many clusters of computers out there dedicated to solving very complex but splitable problems. There is a lot of literature available, and diverse architecture types have risen over the years, such as symmetric multiprocessing, asymmetric multiprocessing, and this massively parallel processing one. Next, I will expand on massively parallel processing as it has a direct relation with columnar databases and big data processing architectures.

Over the years, many different theoretical paradigms were put into practice and relied on in these architectures, such as divide and conquer algorithms, map-reduce techniques, the fork-join model, decomposable aggregate functions, and many others. Also, with the progress made in computing power, highly accessible scalable architectures, the cloud, etc. all of these architectures became more readily available to everyone that wishes to leverage their power and harvest the results of using them - faster and simpler than ever.

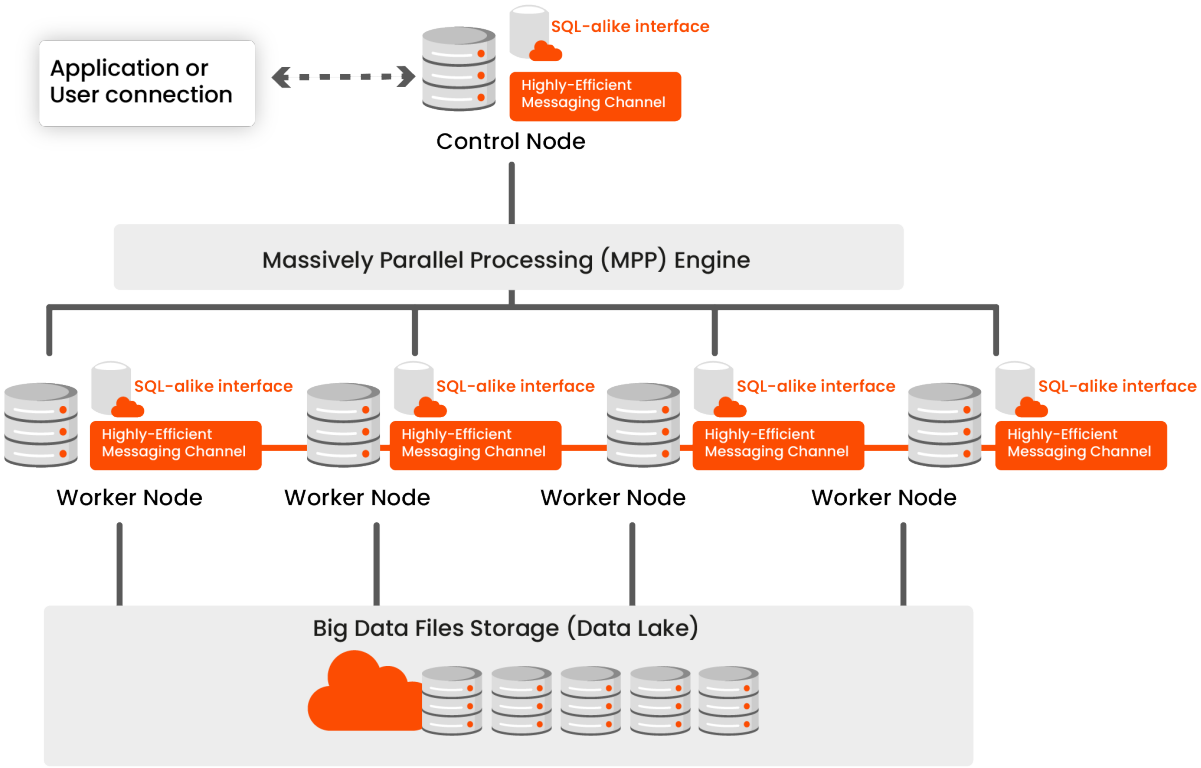


Representative Divide and Conquer & Map Reduce pipelines



Today MPP platforms consist of a large number of processing, or worker, nodes that are connected between each other over a high-speed network. Typically, these kinds of platforms are also known as a loosely-coupled or shared-nothing systems – each one of the nodes that form part of the cluster are independent of each other, as they don't share memory, the operating system nor even CPU in most cases, but they are controlled, or orchestrated, by a master node that is in charge of assigning a task to solve and gather the results for putting together the final outcome once all partial results are available. Depending on the platform itself, this orchestration might happen in a coordinated or uncoordinated way, depending on whether all nodes are doing the same in terms of processing and exchanging data to solve a task along the network, or they might act independently and exchange asynchronous messages once their state changes.

As data at high rates can be streamed through the network, and addressed to specific nodes, an MPP platform is perfectly suited for data parallel applications. It is here, in this case, where all worker nodes execute the same logic on different pieces of data. Moreover, Analytical Massively Parallel Processing Databases (MPPDB) are optimized for analytical workloads that require aggregation and processing of huge datasets. These databases tend to be columnar so, as I showed you before when I talked about big data file formats, the same physical approaches for storing data are applied and rather than storing each record attributes in rows they use the columnar approach. As I explained before, and because of the many benefits associated with it, these structures allow complex analytical queries to be run much more efficiently. Here, as each one of the worker nodes contain their own storage and compute capabilities, a portion of the query to be solved is assigned to them by a master node that orchestrates the whole thing by distributing big datasets among them to collect and merge the results once each one has completed its task.



Massively Parallel Processing Database high-level architecture

As I referenced before, the scenario published by SOUTHWORKS to analyze and correlate different big data sources to analyze COVID-19 evolution through social behavior showcases how you can utilize and leverage these platforms for your benefit.

So... When do you start to harvest the result of all the hard work you have undertaken? How do you get the insights you need to boost your business?

Probably you have already heard about Online Analytical Processing (OLAP) and some of the many tools that are out there. Nowadays, these platforms are the perfect companion for your MPP tool and, what lies underneath, all the data that you already have in excellent condition for analysis after everything you have done to reach this stage – in the past, but also today, they were integrated with your data warehouse solution, and they used to work directly with the data marts built within. Bottom line, OLAP tools allows you to query and analyze data interactively from multiple perspectives – they are based on multidimensional data structures.

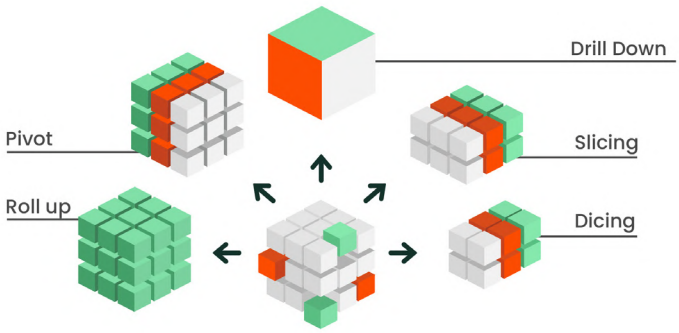
OLAP platforms provide a broad set of operations to manipulate data in order to view it from different angles – as all the information retrieved is glued into a cube-shape form. These operations allow you to navigate the different dimensions the dataset you have queried has, but also allows you to manipulate interesting subsets of the cube itself.

A roll-up operation performs an aggregation of data that can be accumulated in one or more dimensions through the hierarchy using 'measures', which are the different aggregation functions the tool provides you when you are performing this operation – they vary from one solution to another, but the common ones are there (sum, mix, max, count, etc).

In contrast, a drill-down operation allows you to navigate all the way down to the last dimension in the hierarchy (a leaf node in a graph representation) to see and understand further details of how that value for that dimension is there and the path (all previous dimension values) that lead to it.

A pivot operation allows you to reorder the dimension hierarchy of how you see your data. This interactive operation provides a way to understand how different paths lead to a particular value on a drill-down operation, or why an aggregate shows another – at the end of the day, it will help you understand how each dimension influences others, and through careful analysis you will certainly get valuable insights.

Finally, slicing and dicing operations allow you to cut through or take out a specific set of the cube, and view each of the the slices from different perspectives using all previously mentioned operations.



Online Analytical Processing Tools Operations

As each cube you can build is highly customizable, you have a powerful ally in performing pattern detection within your information to extract the knowledge you are looking for (at least manually, later on I will discuss techniques where you can automate this, up to a point). It is clear that you have hundreds of (or as many as you design) potential ways in which you can group or break down your data, analyze it from different angles and perspectives, to finally understand how one dimension affects others and so on. You are not tied to runnig or building the perfect query with the hope of getting what you need to improve your business from the first attempt. At the end of the day, if you have done your homework properly before reaching this point, you have an amazing tool to begin an interactive process where you refine the analysis over time.

Later on, we rely on visual elements like graphs, histograms, maps, charts, plots, etc. to simplify the process of identifying patterns, trends and outliers in big data sets. Our eyes (and brains) are trained to work with colors and shapes to easily extract insights from what it is being seen – this process is part of a visual culture developed through history. It is a process that you see every day, executives rely on graphs and visual elements to share information with company's stakeholders, teachers use it to help their students to visually retain ideas, explain complex concepts or share class-performance indicators, and finally, in our field of interest, computer scientists make extend use of them in order to clean and prepare datasets for further processing – in this last case the numerical results from what might be complex algorithms and techniques, are more easily interpreted by visualizing their outputs.

Data visualization tools and technologies, in big data contexts, are essential for analyzing massive amounts of information and making data-driven decisions. The tools and techniques used within this domain are crucial to understand the story underneath your data. Whether you might have a very good knowledge of the business domain you are working with, or not, the usage of these tools removes visual background noise from data and highlights what turns to be useful information.

Data visualization is also important in cases where you have very little, or no, domain knowledge. It can help you to identify and understand key relationships within your data through charts and plots. You start to learn about the domain behavior – always based on the paradigm that you have reliable and accurate data.

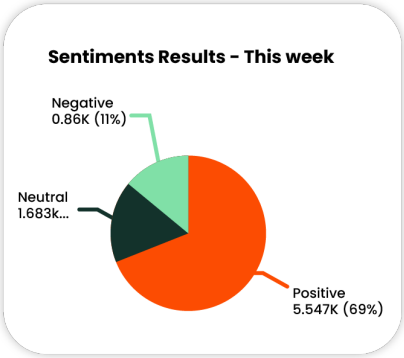
Nevertheless, you should use them carefully. A simple plain graph with few points might not say anything about the data, or it might steer you towards making the wrong assumptions – it won't transmit the proper message at the end of the day. On the other side of the picture, the most complex and elaborate graph might hide interesting data insights under a wide spectrum of colors and shapes. As simple as this topic might be, effective data visualization requires expertise – a well established balance between data and visualization will bring up the story you are ultimately looking for.

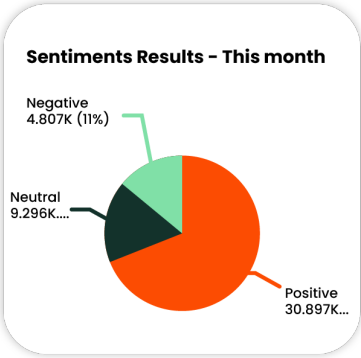
When dealing with Big Data, this requires powerful analytical systems that, through well-defined data pipelines, make the job of these tools easier when reaching the final step. Luckily today, there are several amazing tools out there, and by having great integration capabilities with cloud infrastructures, they finally put at your fingertips all of the power you are going to need. However, as you see through this and previous chapters, there is a long way from raw data to when you start using visualization tools - once you have walked through it properly you will get the results that are going to boost your business to the next level.

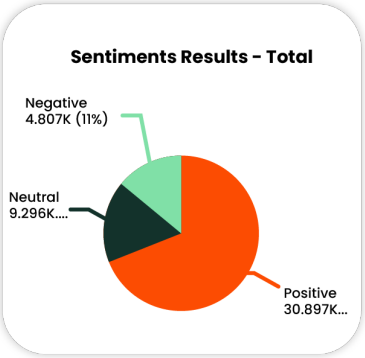
A TYPICAL SCENARIO – SENTIMENT SEGMENTATION

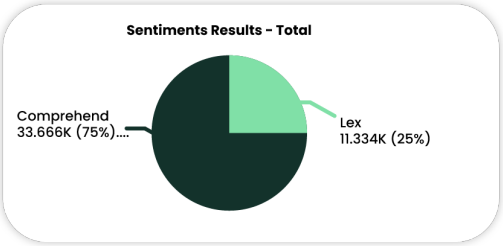
SOUTHWORKS published a scenario that captures and analyzes different pieces of information submitted from participants in text or audio format describing how they feel. A processing pipeline stores into a data lake the result of analyzing each piece of data being converted to information for later and further processing.

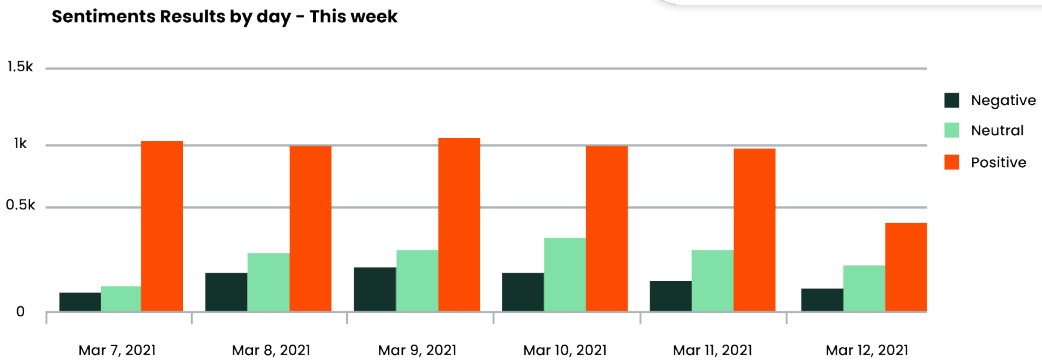
This is the second piece of the scenario described in the previous chapter and, while the series of posts showcases many other things, I want you to focus on the segmentation using different data attributes and correlations, and the visualization analysis with the tools being used.











Want to learn more? Read the full write-up

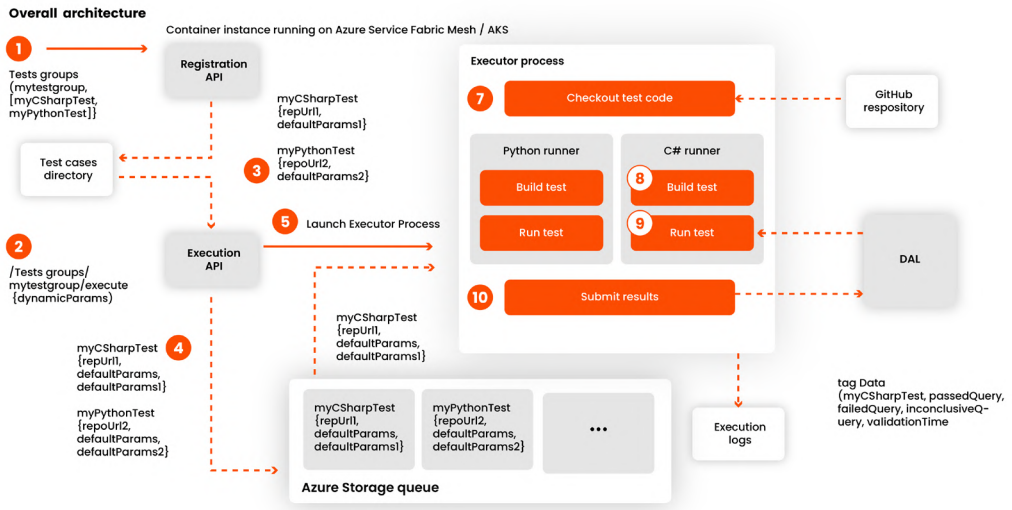
CUSTOMER STORY – DATA QUALITY FRAMEWORK SOUTHWORKS

Partnered with a NY-based financial company that manages over $10 trillion in assets, and designed and built a data quality assurance framework that validates financial information as it comes in. They can create and maintain test suites for every ingested piece of data and map them to the correct assets to automatically run them as it integrates with their event bus orchestration solution.

We put together a solution that relies on Docker images and can be deployed to Azure Service Fabric Mesh or Azure Kubernetes Services to orchestrate the different components it has. Starting from a proof-of-concept, demonstrating the whole concept using Azure Container Instances (to scale up or down to cope with evaluation rules demand), to a minimum viable product (MVP), and integrating this important piece that ensures data quality - in less than 8 weeks.

The solution, part of a bigger data pipeline, catches errors — from duplicates to incorrect inputs — in near real-time so that they can resolve any errors before they cause problems, and they can leverage whatever custom logic is required, within the tests: from static algorithms to machine learning trained models that can raise the warnings necessary.

Now, their data is more accurate than ever. Because each update is properly verified, they can be confident their clients' financial decisions are based on the right information.



06.

When can you start taking Data-Driven decisions?

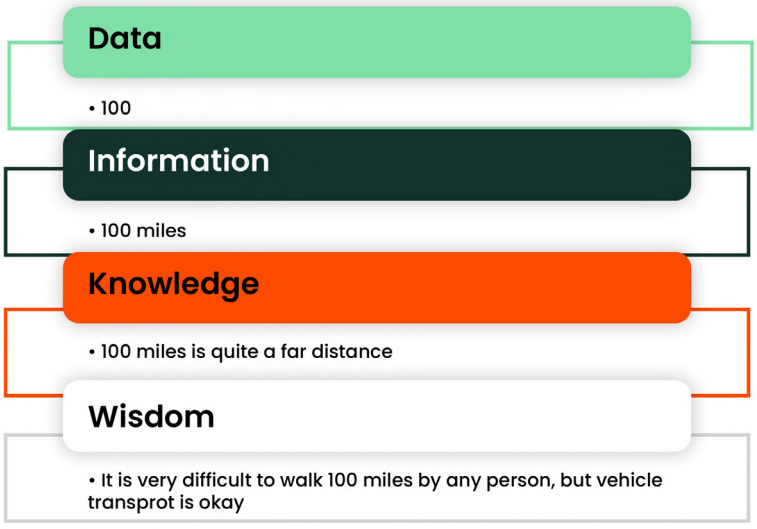
DATA-DRIVEN DECISION MAKING

The main driver of everything we do is to understand the world around us by processing all of the information that is constantly being input to us, and then act consequently. But when do you start seeing beyond individual concepts? At which point can you say that you understand something, and have found meaning?

Let's go with a simple example. Picture this: you are with a non-English speaker friend listening a song, and the lyrics go: "Fun how ev'rything was Roses when we held on to the Guns" . Your friend asks you to translate it. So... Will you do it literally, or do you process and translate the metaphor? Which one adds more value? Some people might argue one or the other, depending on the goal they want to achieve. But at the end of the day, both are super valuable – the raw input, and its apparent (processed) meaning. Your friend can agree with you or not, you can engage in a fun discussion, but if you also share how you arrived to that meaning they can learn.

Is this Wisdom? Well... You may argue it is, or isn't, but it is the last step of our data journey evolution. In the previous example, if the friend just receives the literal translation, if they are smart enough, they will understand the underlying meaning of the metaphor hidden in the phrase without any direct teaching linked to it (with no previous 'programming') through a cognitive reasoning process that will likely get the correct answer. This is something that machines have not achieved just yet, and if you were told that you needed to train your brain in advance through different mathematical functions as derivatives, quadratic errors, and so forth in order to arrive at the correct answer, you would have thought that's nonsense.

During this chapter, I will explore how the term 'wisdom' can be applied to data pipelines with the goal of decision making (automated or not). From the different approaches you have to monitor, and to react to your data almost instantly, to develop a self-aware system. I'll wrap up the chapter with an important topic on Data Governance that you should always keep on the radar through the data journey.



The journey – from raw data to wisdom

EXPLORATORY ANALYSIS

What's the simplest approach to start with when you have millions of data points available from past experiences? The answer is straight-forward, forecasting.

Put it in simple terms it is the ability to predict (mostly estimate) a future trend or event based on what happened before and what it is currently happening. Once you have a 'good amount' of reliable curated data, and depending on the domain you are dealing with, it is an impressive tool that can give you accurate results. Traditionally there are several types of well-tested forecasting methods based on mathematics and statistics that people have used over the years, from moving averages to multiple linear regressions, to mention a couple.

Of course, there are some things that are easier to forecast than others. Take for example weather forecasting: it is easy to know what time the sunset is going to happen tomorrow, and the day after and so on, but other times it is hard to know if tomorrow is going to rain at 4PM or 5PM, or neither, or both precisely – there are lot more variables to take into account to solve that statement, and ultimately, you use statistics here to infer about the relationships of those variables in a practical way.

The key factor here is knowing when something can be forecasted, or not, and what's the probability of what you forecasted, to what degree is it accurate – otherwise your forecast won't be better than flipping a coin in the air. There are three main dimensions to consider when you plan a forecast: what are the factors that affect the data (all variables), the volume of historical data you have available, and how the forecast might be self-affected by its own forecasting. If you handle all of them properly (by being aware of their impact), then you will end up with a good forecast from relationships and patterns that exist in the historical data.

The first two dimensions are clear, and to put in a simple example, let's take forecasts made by electrical companies. They take into considerations variables that might affect the data as temperature conditions in different seasons, economy fluctuations, and calendar variations like holidays or vacation periods. They understand the variables that have direct impact on the scenario, and they have a sufficient amount of historical data to make accurate predictions (or forecasts), and they don't need to worry about the forecast affecting the prediction by itself.

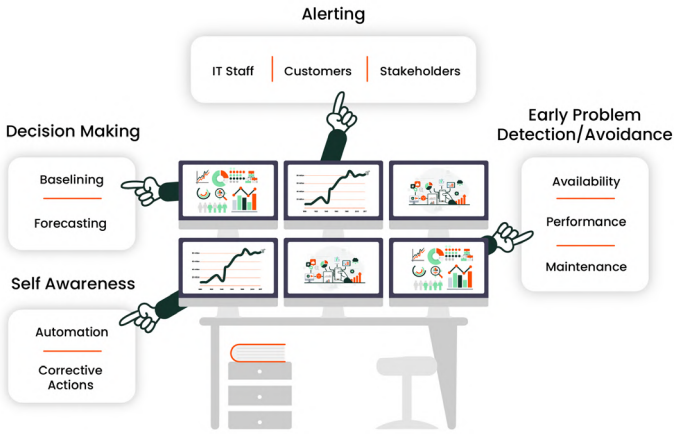
So, what about the last dimension? Let's suppose you are forecasting stock prices. Here you can be confident that only one dimension is fully fulfilled which is the amount of data available you have at hand. But what about variables that might affect the forecasting? Those are difficult to determine, they might vary from time to time, and we have very little knowledge of how new factors can get into the picture and be of less/big impact. Finally, when you share the forecast people react by buying or selling according to it and the forecast itself is subject to change dramatically – price changes might go on the opposite direction to what you forecasted.

All those forecasting and estimation alternatives are excellent to 'know the future', or at least try, and to take decisions based on the relationships discovered – if A and B go this way, then C 'should' behave like that. But you also need to put your eyes on the present, on what's is currently happening in front of you – the real time analysis of data flowing through your systems is an effort that pays off immediately.

On the proactive side of the picture, by having a monitoring framework you define a set of tools, a clear process, and activities, you fulfill the simple objective of being aware of the state of the system at every moment. This proactive approach eases your maintenance efforts, and no matter if it is due to your operators actively watching dashboards with visual indicators and metrics, or automated by a system with a specific set of rules and corrective actions, you rely on current snapshots of processed data to make decisions. In some cases, whether you use the forecasting techniques discussed above or machine learning algorithms, you can also put preventive maintenance efforts in place to anticipate and correct issues before they even happen.

But monitoring benefits are not only related to avoid issues within your system going unnoticed, it is great also to evaluate and improve the system's performance and discover parts of it that can be automated (or better automated if already done). By constantly having an 'eye' (human or digital one) on what's happening under the hood, you end up with a more reliable and resilient system that suffers almost no downtime.

On the reactive side of the picture, by having alerting mechanisms to let your maintenance teams to look at a specific incident, or indicators that might cause an incident, you can react sooner than your competition and keep your business up and running with no outages or disruptions. By relying on customized processing pipelines for different pieces of information flowing through the system, these alerts can be targeted to humans or systems that can take the proper corrective actions to put everything back on track. More and more delivery channels to make alerts reach their destination are available every day, from emails, text messages (SMS), instant messages (IM), phone calls, to HTTP REST APIs, when you automate this between systems.



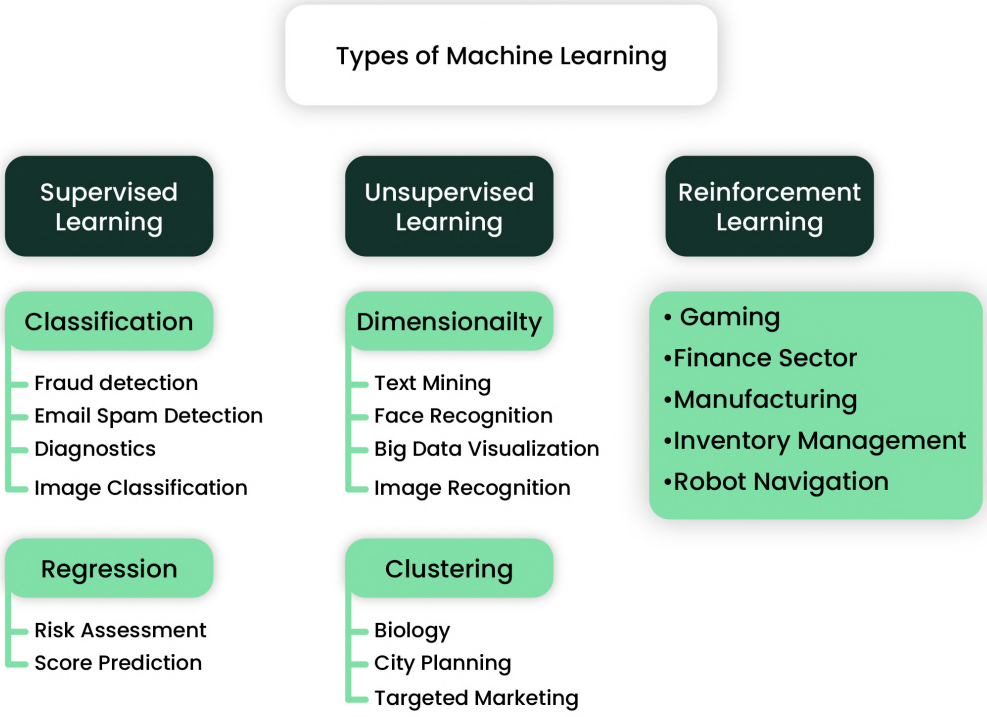
Opportunities of Real-time Monitoring and Alerting

With the aid of the real-time processing, statistics, and data analysis tools that exist today, you can combine monitoring and alerting, plus automated corrective actions to bring to life a 'self-aware' system – one that can detect, or even anticipate, its problems and correct them. Sounds like science fiction? Well, of course you will have constraints and limitations that won't let you take this the whole way (you won't get Skynet), but more and more advances every day – connectivity boost through 4G/5G networks, cloud services and platforms, scalable computing power, IoT devices, mobile phones - are extending the reach of what you can connect, monitor, and automate.

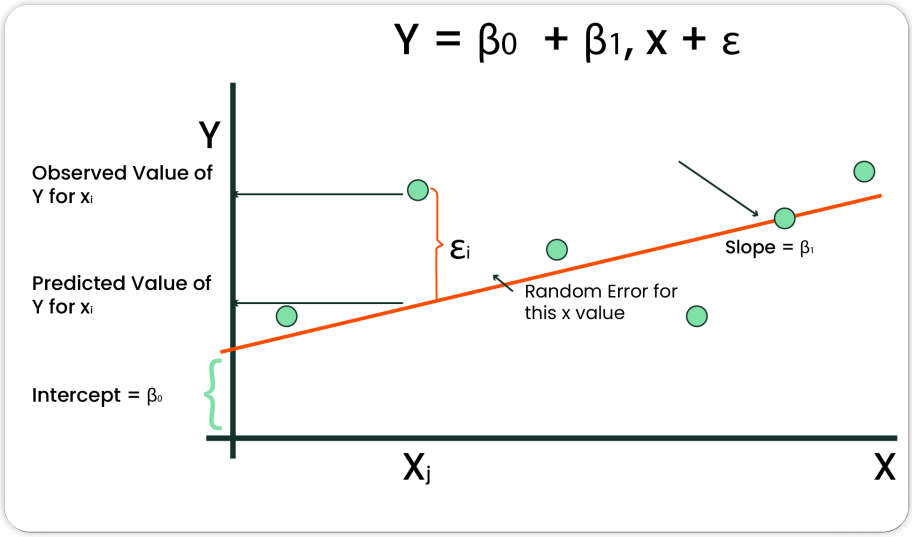
On my way to wrapping up the chapter and the whole data journey story, I want to touch on another prediction method that is a buzzword today: machine learning. This is not at all a new concept or modern approach, it is based on statistical learning theory, and was developed in the 60's – due to the requirements on computing power and massive amount of data required it was used in very specific fields back then; today it is more accessible to everyone thanks to all the data available and cloud computing.

Besides, machine learning is also built on-top of statistics, there are differences from the statistical models used within forecasting approaches. When we apply statistical models at the service of forecasting, as I mentioned before, we are interested in understanding relationships between the variables in the data with the final goal of making a 'prediction' (rather more an estimation) based on that 'wisdom'. A machine learning trained model does not care about helping you understand these relationships - although the model someway internally might 'do', within multi-layer deep neural networks for example. The model just cares about providing a result that is good enough (accurate) for what you want to predict.

In addition, how you prepare both models is different. Although both have specific parameters, the statistical model does not require you to put together a test set to complete its evaluation, where the machine learning model does. In the latter case, you need to carefully split your dataset into at least two subsets: training and test – sometimes you can arbitrarily split it, depending on the distribution of your data for example Gaussian or not, otherwise you can't, and need to be extremely careful otherwise your results are going to be biased.



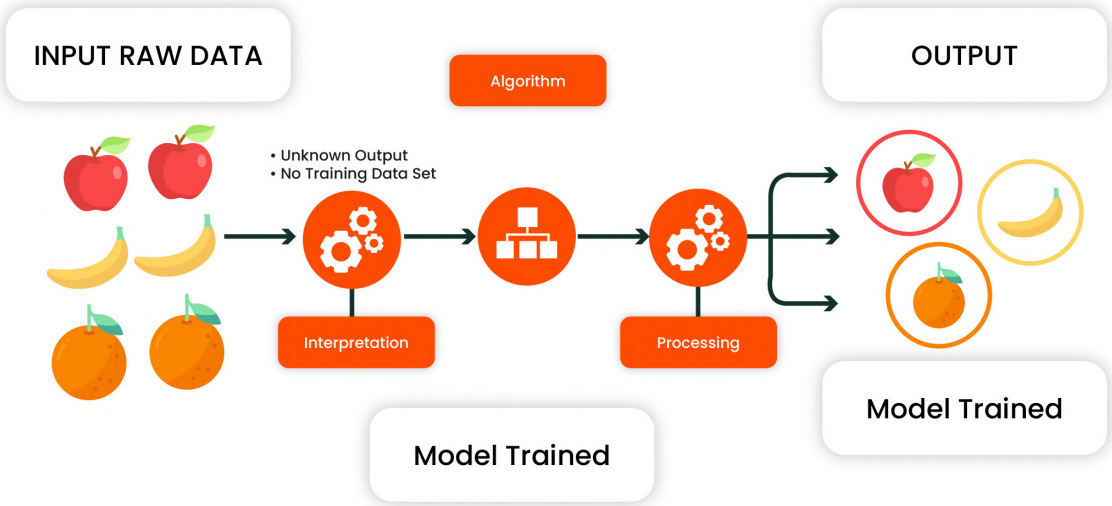
Machine Learning Techniques and Use Cases



What's a Linear Regression?

Let's explore a simple example taking a linear regression. The statistical model has an epsilon parameter that is taken into consideration to minimize the mean quadratic error between your data points (of course here we all assuming the data has a linear behavior) – there a statistician chooses the best function that can be applied to solve the problem.

A supervised machine learning will take a different approach, although it also relies on statistical algorithms to solve the problem, the mechanism will be different: here it has a space of different linear regression algorithms (like simple linear regression, gradient descent, ordinary least squares, or many others) to choose from, and its goal is to pick up the best. You will define hyperparameters (inputs the process needs), and then it will iterate through the data while adjusting weights and verifying results using a loss function (a method it can use to compare its result against what you expect) until converging to a threshold or value that it is acceptable for you – at the end of this process you will get what is called a trained model (with an underlying algorithm of its choice) that is acceptable to your standards, or you can repeat the process with a different set of hyperparameters.
Later, you will (must) test how successfully the model was trained by running it against your testing subset. This testing is required because (without entering into too much detail), the algorithm focuses on minimizing loss, which impacts directly on the expected risk (in practice more empirical) by either overfitting or underfitting the model to the submitted data, depending on whether you have enough data for the training process. Regardless, both an overfit or underfit model will predict far from expected given unseen data.



A typical Supervised Learning Model Construction

Other machine learning techniques use different approaches, algorithms, and models, but they are more or less based in the same concepts: a training set, verification sets, a test set, and a loss function for the model training iterations. They need lot of data to return a trained model, and as I say at the beginning of the article: the more data the better. Models with high accuracy can detect fraudulent transactions, valuable customers for selling opportunities, credit risk towards loans, or even devices that require maintenance because they are about to fail. These models assume that historical patterns that happened frequently in the past, within the domain you are analyzing, will repeat in the future. In contrast to the forecasting, more statistical models discussed before that provide you an estimation, these models are here to give you a prediction.

We will leave our discussion of machine learning there for this paper, as ML is a huge topic. Keep an eye out for more to come from SOUTHWORKS on the topic of ML.

	Statistics	Machine Learning
Approach	Data Generating Process	Algorithmic Model
Data Size	Any Reasonable Set	Big Data
Dimensions	Used Mostly for Low Dimensions	High Dimensional Data
Driver	Math, Theory	Fitting Data
Focus	Hypothesis Testing, Interpretability	Predictive Accuracy
Inference	Parameter Estimation, Predictions, Estimating Error Bars	Prediction
Model Choice	Parameter Significance, Insample Goodness of Fit	Cross-validation of Predictive Accuracy on Partitions of Data

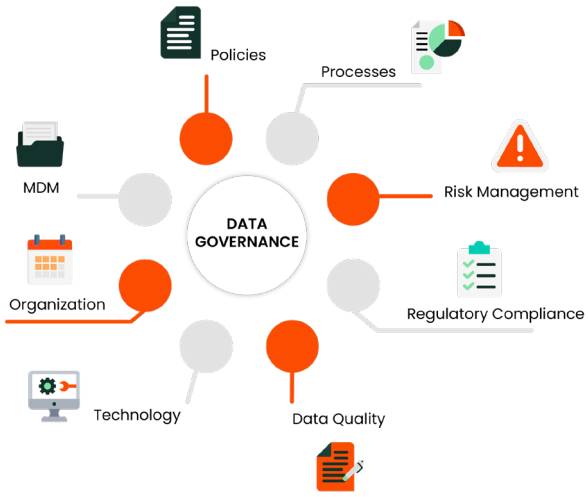
Statistics Estimation (Forecasting) vs Machine Learning Prediction

UNDERSTANDING THE REGULATORY LANDSCAPE IN THE DATA JOURNEY

Before we conclude our data journey story, let me raise that it is very important to be completely aware of how data moves through your organization - through the different departments and groups. Nowadays, there are regulatory standards and requirements, plus privacy laws, that you need to comply with depending on which industry and regions you operate in and what kind of business you are. As a core premise that crosses all the different data pipelines and process you might have built, security and confidentiality are two non-negotiable governance items you must handle very carefully.

Although each of your systems might have different access policies and requirements, your data storing, transmission, manipulation methods should be compliant and consistent organization wide – from using encryption standards for data in-transit and at-rest, to multi-factor authentication policies to access very sensitive information, no matter which system needs to query it. Data Compliance procedures define how your organization handles sensitive data such as private personal information. Well-defined rules for sharing, storing, retrieving personal information are mandatory to almost every organization, and it is your company who is responsible for safeguarding it.

A full and compelling Data Governance framework helps you manage your data effectively through all its journey. Starting from collecting it as raw from its origins, flowing through all different data pipelines you have in place to transform and process it, and finally reaching its consumption point, where business intelligence tools and analytics processes are in place to extract valuable insights. This is a must for every organization – without it, data might fail to meet regulation, comply with data standards, and through different risks it might be exposed to security holes that could compromise its integrity, the integrity of your decisions, and worse, decisions your customer takes based on the data you share with them.



Data Governance Framework

All these data governance and compliance mechanisms should be on your radar at all times. Every action you take to synchronize, read, write data should be aligned with specific needs. Any breach in these protocols or procedures might lead to fines coming from government and industry regulations, and if that's not the case, at least it will draw your business to lose reputation – potentially losing customers, and ultimately, revenue.

CONCLUSION - WHERE DOES THE DATA JOURNEY GO FROM HERE?

To wrap up our paper, I would like to briefly discuss what to expect for the future. Well, it is obvious that both the amount of data and the pace at which it is being generated are going to keep increasing.

By 2025, the International Data Corporation (IDC) predicts that worldwide data will grow 61% to 175 zettabytes - as much of this data residing in the cloud as it is in data centers, although the first option is becoming an increasing trend. It is predicted that by 2025:

Almost 50% of data will be stored in public cloud environments

IoT devices will play a very important role generating almost 51% of the total amount

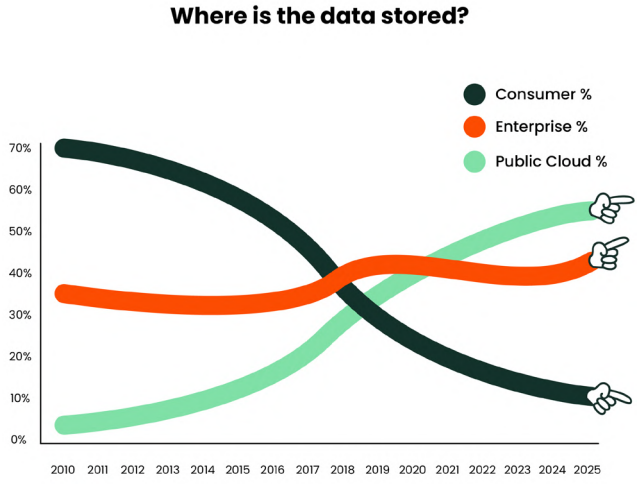
There are going to be more real-time use cases to process and consume that information reaching almost 30% of the it

So... How do we expect to cope with bigger volume of data to drive our businesses better? Well, one particular forecast says that the storage industry will be able to deliver 42ZB of capacity over the next seven years.

Computing power also doubles about every two years according to Moore's law, and cloud platforms build more and more data centers in different regions of the planet.

Additionally, the number of data generation and consumption actors will increase dramatically by billions of connected users, devices, and embedded systems will constantly be delivering new pieces of information at increasing paces. Storage systems that collect and share all that raw data being generated for data analytics and other use cases every day, will be more accessible and available all over the world.

Enterprises will continue to have bigger and greater opportunities to store and analyze these huge volumes of data. The trend will also be to store more and more data in the cloud environment, and leverage infrastructure and services offered by those platforms to build your data pipelines.



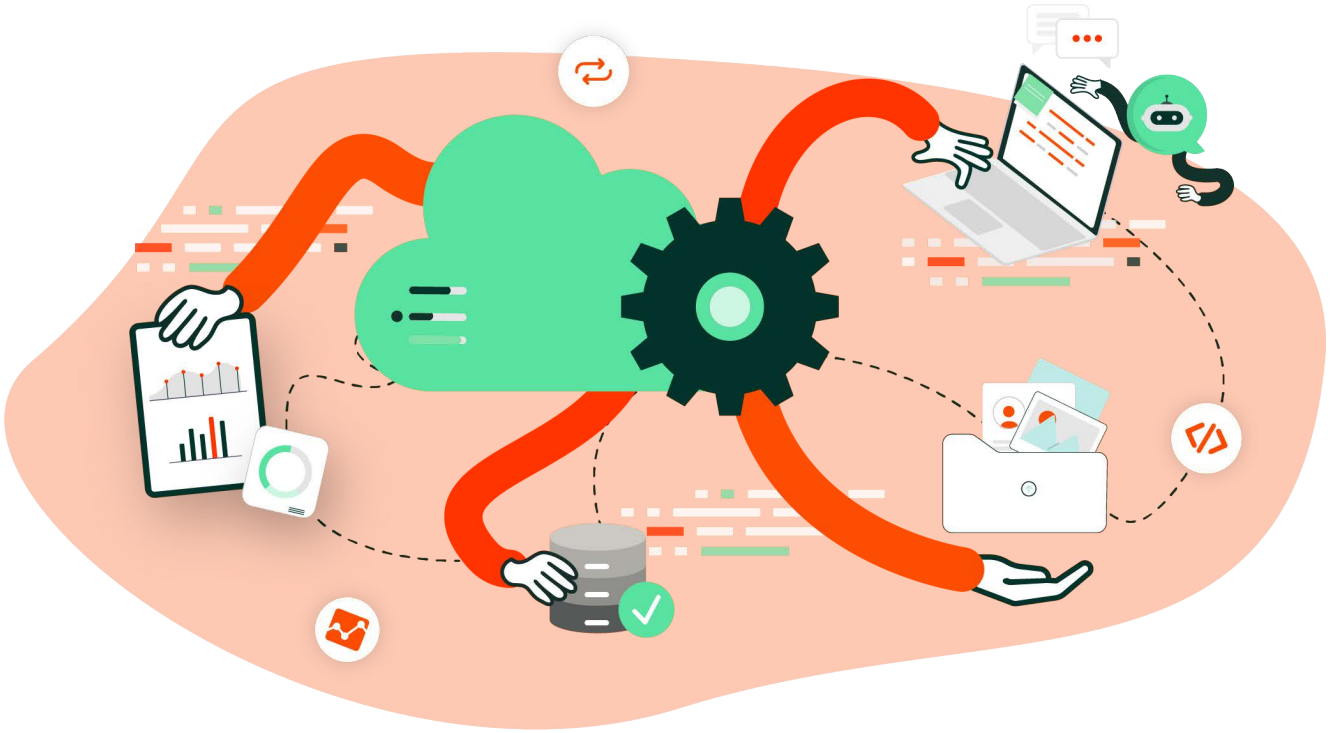
Data Storage Forecasting

And how to leverage all that available amount of data will keep evolving. With continued investments on AI and machine learning to build from better business intelligence platforms that take or to help you taking the best decisions, to smart agents, assistants and robots that help consumers take them.

It is clear that whether you want to leverage the power of AI to drive your business better, or with the aid of ML automation be able to predict and solve problems faster than your competitors, you must first set your data straight! Even more than anyone else if your business is within healthcare, manufacturing and retailing, financial services or media and entertainment industries - hot spaces for Data & AI scenarios.

This continuously evolving and challenging landscape brings lot of opportunities, but it also brings lot of challenges. The topic that I mentioned earlier in this chapter regarding Data Governance and Compliance will continue to be an increasingly hot topic that you should pay special attention to. Protecting your data from intrusions, cyberattacks, leaks, wrong permissions, etc. it is a must if you want your business not to lose reputation and revenue – today, maintaining high-levels of data protection with data growth rates is not a piece of cake.

If you are not sure how to put all this into action and start using your data to create competitive advantage, don't despair! Keep reading and I will tell you how we at SOUTHWORKS we have been helping many different types of organizations to embrace data and how we can help you.

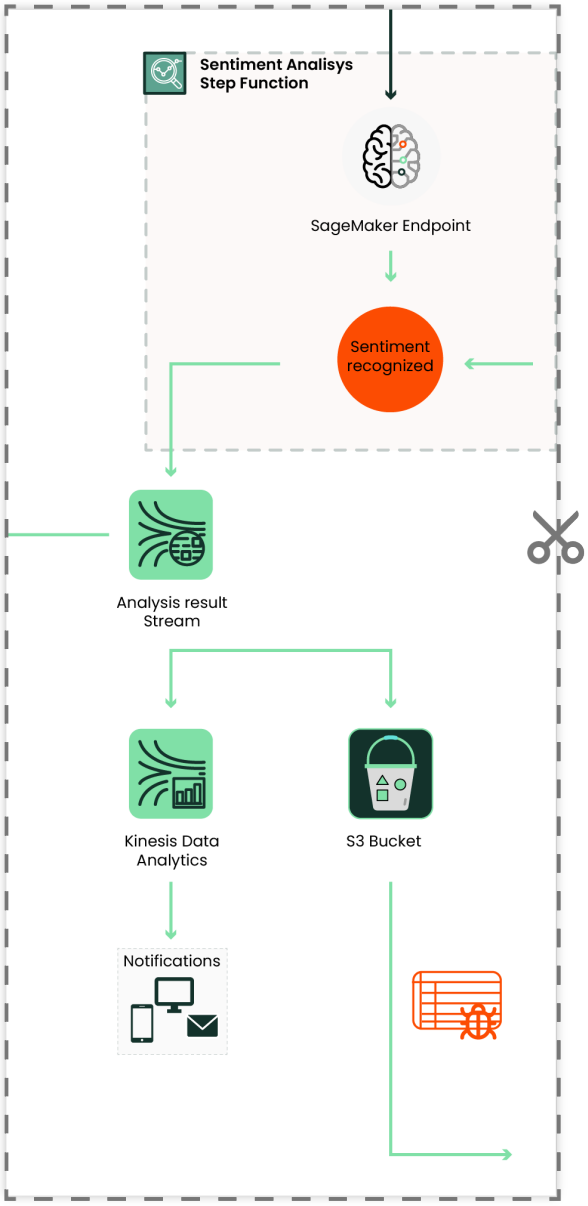


A TYPICAL SCENARIO – REAL TIME MONITORING AND ALERTING

SOUTHWORKS published a scenario that captures and analyzes different pieces of information submitted from participants in text or audio format describing how they feel. A real-time processing pipeline analyze each piece of information submitted by using a custom ML model and pass-through the result of that analysis to other part of the pipeline that does several things, among them, and the one that is of interest to see here, is to monitor and alert in real-time on specific scenarios where a manager should intervene.

The platform allows managers to create groups and invite participants to register into these each of these groups. Once done, participants are allowed to submit feelings (check-in points) and Data-AI pipeline automatically analyzes uploaded messages — creating time-series graphs on participants feeling evolution and notifying a manager in case an immediate action needs to take place. Sentiment analysis is a central piece leveraging AWS SageMaker to use advanced machine learning techniques.

Showcasing, and to better understand, what I discussed within this chapter, I suggest you focus on how the piece of the pipeline highlighted below works:



Want to learn more? Read the full write-up

CUSTOMER STORY – SELF-AWARE CONVENIENCE STORE

Our customer is the world's largest convenience store chain, consistently ranked as one of the top franchises in the world. Open 24 hours a day, their easily recognizable stores meet people whenever, wherever, with an innovation-forward approach to convenience.

But manual software and hardware maintenance is throwing a wrench in store operations. For example, when a key service is down —if the store associate even notices it's down to begin with— your associates are forced to do things manually while they contact support and wait for someone to investigate remotely, which just slows everything down.

They turned to SOUTWHORKS, who designed a solution using IoT agents and AI/ machine-learning algorithms, that give the operational team a real-time view into the health of the stores' services.

Operators get a heads up when a service is unhealthy, they can identify issues remotely (without no need of store-support being contacted from the store), or even they can setup up self-healing rules for some crucial services and leverage their automatic recover to the self-aware platform developed. Not only does resolution happen sooner, but they also get more accurate information about what's going on in your stores.

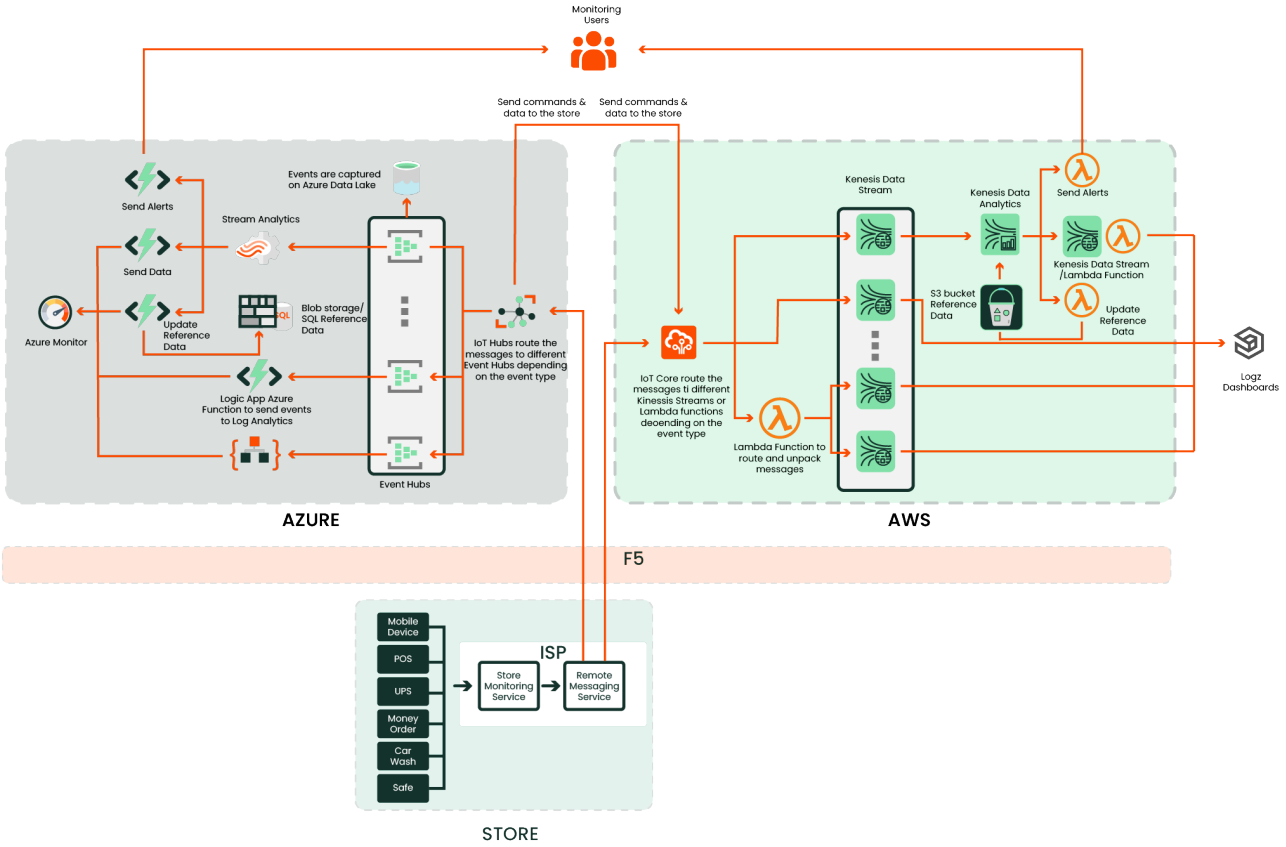
In addition, predictive maintenance algorithms raise alerts to operators and store-support based on the telemetry being received over time by the different systems and devices running at place. This way they can schedule store-maintenance team visits more accurate than ever, and they can proactively prevent service shutdown by having all individual components required to fulfill transactions up and running better than ever.

IN A NUTSHELL

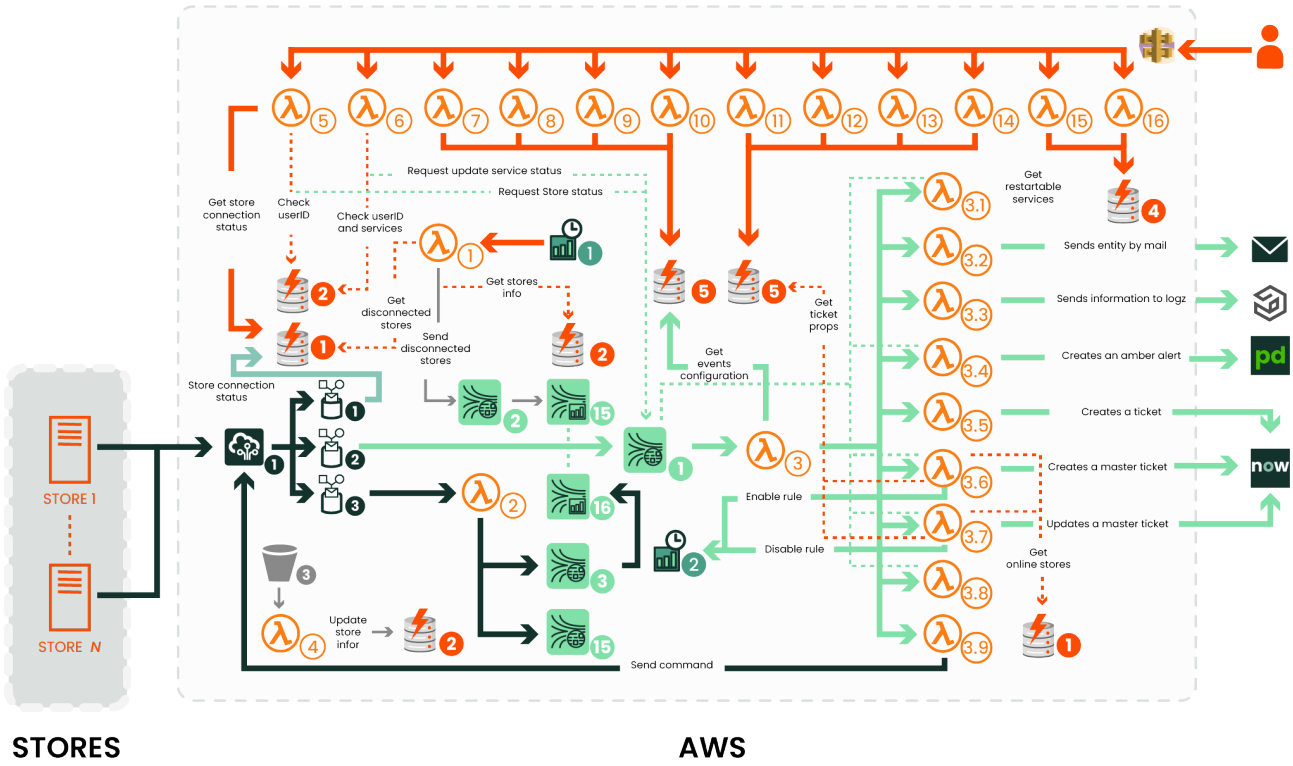
- Monitoring of 10K stores country-wide near real time
- Developed IoT and Cloud Analytics pipelines for a Multi-Cloud ecosystem (on both AWS & Azure platforms)
- Developed a new set of services & features running at store side
  - SMS (Store Monitoring Services)
  - SMS – RMS integration (to support MCO on AWS & Azure platforms)
- Integrated with 3-party services to raise alerts or trigger preventive actions (by an operator in a manual fashion or automatically leveraging ML/AI cloud services)

OVERALL ARCHITECTURE

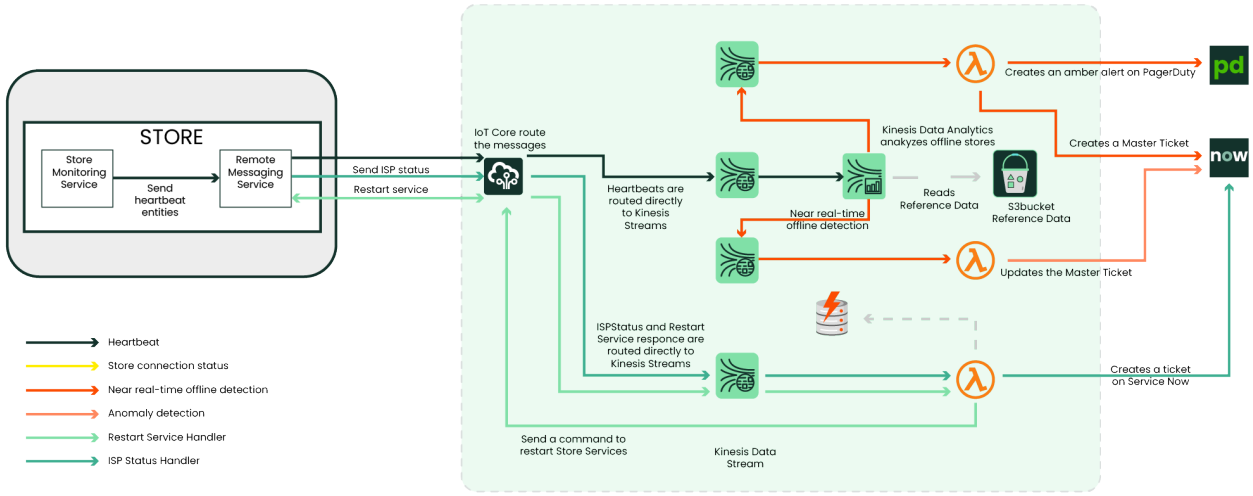
MONITORING AND ALERTING



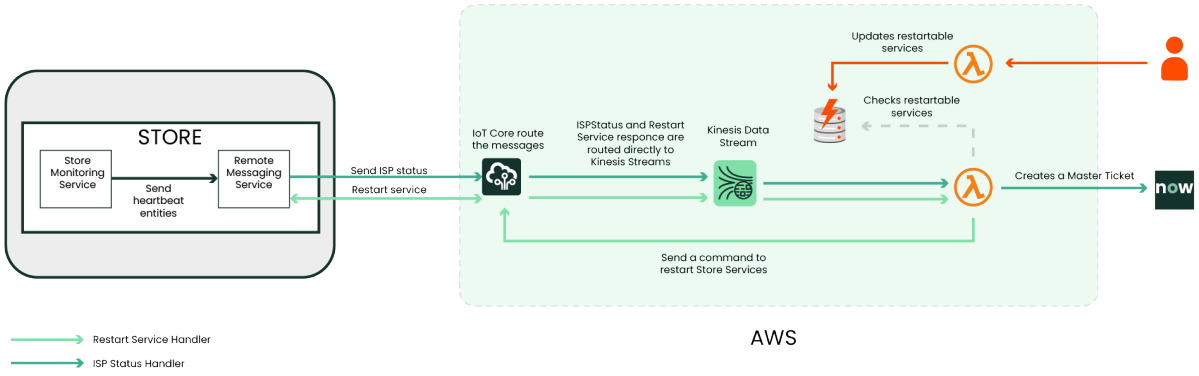
INFRASTRUCTURE



SELF-HEALING SCENARIO



AWS

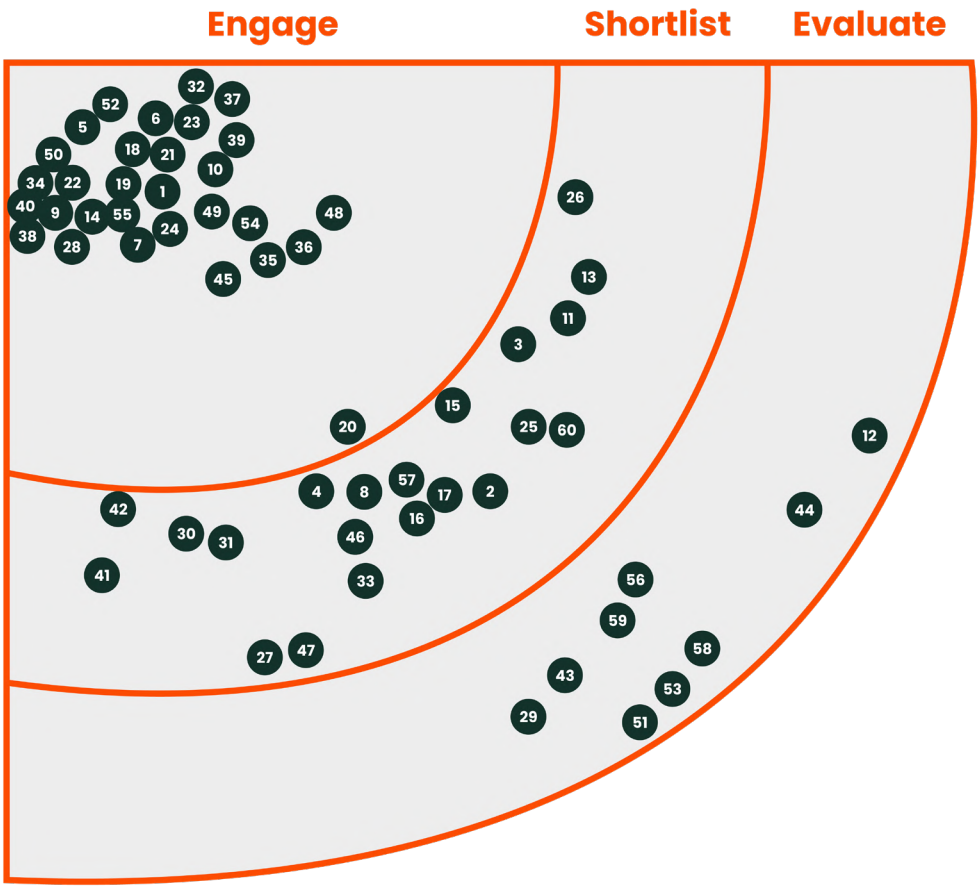


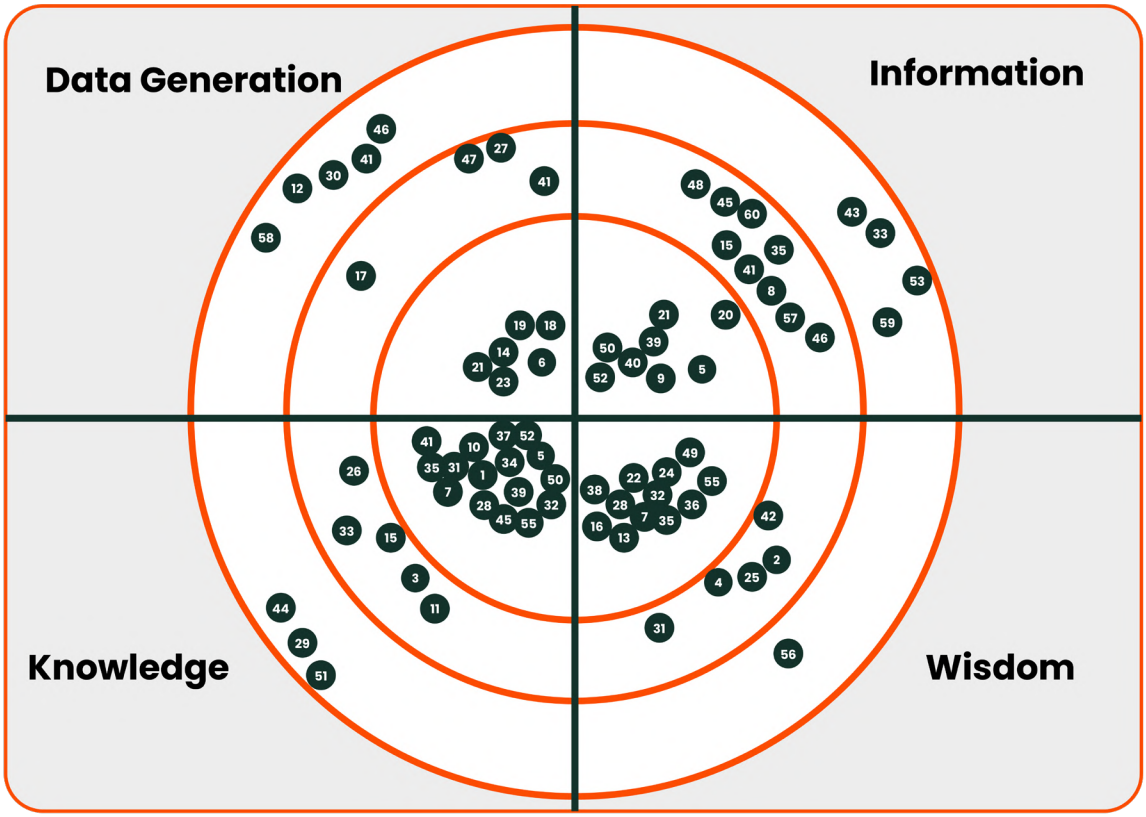
AWS

Our customer has reduced its time to resolution for their operational incidents. Their 24/7 stores are running smoothly around the clock. Through currently deployed (more to be rolled out) machine-learning features they have the ability to identify patterns, prevent downtime, and detect causes of unexpected scenarios in the future.

07.

Technology & Methodology Radar


Engage
Shortlist
Evaluate


Data Generation
Information
Knowledge
Wisdom

I've captured in this radar-shape diagram below the technologies, platforms, and methodologies most used and recommended to leverage during your data journey. As there are many technologies out there, and many more being added each day, this is not a complete list of all what is available, but the most common and used tools by partners, vendors, and the teams I work with through different SOUTHWORKS projects.
The diagram is divided in 4 quadrants, each one for each step of the data journey going from data, to information, to knowledge, and finally wisdom. Each of the quadrant is divided in 3 sections showing the following categories:

- Engage: Items within this category are the most commonly used and their results are well-proven by different companies and projects. If you want to use one of them in any of your projects, you should have no doubt about it and go ahead.

- Shortlist: Items contained here are not so frequently used, and sometimes they mainly cover specific scenarios or data (solution) stacks - whether you need to connect 2 specific platforms, or extract data from an old-legacy stuff for example. Of course, these are not to be discarded and, you should have them within your toolbox for your next endeavor towards data.

- Evaluate: Items from this category usually target different niches and very rare use cases – often they are proprietary solutions (far more than others) and if you consider using any of these, you will probably engage with a third- party or specific vendor to deliver the solution for you (a vendor-lock).

Technology recommendations to consider through your Data Journey

1. **Amazon Athena** – It is an interactive query service that makes it easy to analyze data in Amazon S3 using standard SQL– if you have built you data lake using S3 then it is a great choice. Simply point to your data in Amazon S3, and after defining your data structure with a schema-on-read approach, you can start querying using standard SQL. It also can infer the schema from the raw data, using Upsolver to parse and stream the data and Amazon Glue Data Catalog to connect the metadata, schema, and data. This makes it easy for anyone with SQL skills to quickly analyze large-scale datasets.

2. **BigML** - It is used for building datasets and then sharing them easily with other systems. Initially developed for machine learning, now it is widely used for creating practical data science algorithms, and you can easily classify data and find the anomalies/outliers in large sets of data. It has an interactive data visualization process that makes it easy for data scientists to make decisions. Its Scalable BigML platform is also used for time series forecasting, topic modeling, association discovery tasks, and much more.

3. **BigQuery** – Has several capabilities to tackle different use cases, but among them its BI Engine is an in-memory analysis service built into BigQuery that enables users to analyze large and complex datasets interactively with sub-second query response time and high concurrency. With this platform you can query streaming data in real time and get up-to-date information on all your business processes. Predict business outcomes easily with built-in machine learning–without the need to move data. A great service to analyze petabytes of data using ANSI SQL at blazing-fast speeds, with zero operational overhead, if you are considering GCP as your platform to go for hosting, transforming, and analyzing your data.

4. **Microsoft Cognitive Toolkit (CNTK)** – It is an open-source toolkit for commercial-grade distributed deep learning. It describes neural networks as a series of computational steps via a directed graph. CNTK allows the user to easily realize and combine popular model types such as feed forward DNNs, convolutional neural networks (CNNs) and recurrent neural networks (RNNs/LSTMs). CNTK implements stochastic gradient descent (SGD, error backpropagation) learning with automatic differentiation and parallelization across multiple GPUs and servers. You can include it as a library in your Python, C#, or C++ programs, or used as a standalone machine-learning tool through its own model description language (BrainScript), and you can use its model evaluation functionality from JAVA programs. It supports the Open Neural Network Exchange ONNX format, an open-source shared model representation for framework interoperability and shared optimization.
5. **DataBricks** – It is built on-top of Apache Spark – the advantage of it, is that is faster on both structured and unstructured data than SQL type engine like AWS Redshift. As a significant percentage of big data collections are unstructured in their nature, for real time analysis, it may be necessary to query against JSON-style data which is often loosely structured – as we know SQL requires structured data to work well.

6. **Azure Data Factory** – It is a fully managed serverless cloud service that scales on demand. Build on top of Apache Spark, but fully managed, gives you the possibility to prepare data, construct ETL and ELT processes, and orchestrate and monitor pipelines code-free. It offers more than 90 built-in connectors to acquire data from Big Data sources such as Amazon Redshift, Google BigQuery, HDFS; enterprise data warehouses such as Oracle Exadata, Teradata; SaaS apps such as Salesforce, Marketo and ServiceNow; and all Azure data services. It is the way to go if you plan to build your data lakes and data pipelines within Azure infrastructure.

7. **IBM Cognos Analytics** – It is a BI solution developed to clean and connect your data, create stunning data visualizations, and share them to showcase the valuable insights your teams extracted. It can easily connect to several data sources and organizes your data to see everything in one place. A good tool for data exploration as it has AI-aid to better help you shape and define your data towards finding pattern and insights you didn't be aware of – from predictive forecasting, decision trees, to the AI assistant and more.

8. **Cloud Data Fusion** – It is built using open-source project CDAP, and this open core ensures data pipeline portability for users. CDAP's broad integration with on-premises and public cloud platforms gives Cloud Data Fusion users the ability to break down silos and deliver insights that were previously inaccessible. Its integration with Google Cloud simplifies data security and ensures data is immediately available for analysis. Whether you're curating a data lake with Cloud Storage and Dataproc, moving data into BigQuery for data warehousing, or transforming data to land it in a relational store like Cloud Spanner, Cloud Data Fusion's integration makes development and iteration fast and easy. It is the tool you should chose to orchestrate your ETL/ELT pipelines -as it has broad library connectors and transformations (more than 150)- if you plan to leverage Google Cloud Platform for all your data analytics projects.

9. **Azure Data Lake** - Has all of the capabilities required to make it easy for developers, data scientists and analysts to store data of any size and shape and at any speed, and do all types of processing and analytics across platforms and languages. It removes the complexities of ingesting and storing all your data while making it faster to get up and running with batch, streaming and interactive analytics. Azure Data Lake works with existing IT investments for identity, management and security for simplified data management and governance. It also integrates seamlessly with operational stores and data warehouses so that you can extend current data applications. It is no brainer if you plan to build your data pipelines within Azure infrastructure.

10. **Azure Data Lake Analytics** – It can be used to easily develop and run massively parallel data transformation and processing programs in U-SQL, R, Python, and .NET over petabytes of data. With no infrastructure to manage, you can process data on demand, scale instantly, and only pay per job. Relies on U-SQL is a simple, expressive, and extensible language that allows you to write code once and have it automatically parallelized for the scale you need. Process petabytes of data for diverse workload categories such as querying, ETL, analytics, machine learning, machine translation, image processing, and sentiment analysis by leveraging existing libraries written in .NET languages, R, or Python.

11. **Data Studio** – It is part of Google's suite and allows you to easily access a wide variety of data. Its built-in, and partner, connectors make it possible to connect to virtually any kind of data - data from 800+ data sets leveraging more than over 440 connectors. You can quickly build interactive reports and dashboards with web-based reporting tools, and besides you team can collaborate in real-time, you can share later reports and dashboards with individuals, teams, or everyone. A tool to consider if your data and pipelines are within GCP.

12. **DataTorrent** - It can handle both streaming and rest data. It can process many million events per second and can recover from node outages without any loss of data – without needing any external human intervention.

13. **Delta Lake** - Delta Live Tables (DLT) makes it easy to build and manage reliable data pipelines that deliver high quality data on Delta Lake. DLT helps data engineering teams simplify ETL development and management with declarative pipeline development, automatic testing, and deep visibility for monitoring and recovery.

14. **Azure Event Hubs** - Azure Event Hubs cloud-based big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters. It supports Apache Kafka (1.0 and later) clients and applications – although you don't need to set up, configure, and manage your own Kafka and Zookeeper clusters or use some Kafka-as-a-Service offering not native to Azure. It is the way to go if you plan to build your data lakes and data pipelines within Azure infrastructure.

15. **Firebolt** – It has impressive data analytics performance metrics – a pay-as-go alternative, but you should be careful about the total cost of running your data pipelines. It proudly announces that is between 4-6000x faster than Snowflake, Redshift, Athena, and other existent data warehouses for individual queries in benchmarks by customers.
It was built from the ground up with huge datasets in mind. Combines the best of high-performance database architecture with the infinite scale of the data lake, guaranteeing unparalleled performance at any scale. Clusters of compute nodes use Massive Parallel Processing (MPP) to parallelize queries across nodes, through which fast performance can be maintained as data grows.

16. **Apache Flink** – It is a framework and distributed processing engine for stateful computations over unbounded and bounded data streams.
It has been designed to run in all common cluster environments, perform computations at in-memory speed and at any scale. Flink integrates with all common cluster resource managers such as Hadoop YARN, Apache Mesos, and Kubernetes but can also be setup to run as a stand-alone cluster. Bounded streams are internally processed by algorithms and data structures that are specifically designed for fixed sized data sets, yielding excellent performance. It features an ANSI-compliant SQL interface with unified semantics for batch and streaming queries, and provides a rich set of connectors to various storage systems such as Kafka, Kinesis, Elasticsearch, and JDBC database systems, and it is a great choice to fulfill scenarios when you need to build event-driven (scenarios from anomaly/fraud detection to rule-based alerting) and data analytics applications, or data pipelines.

17. **Apache Flume** - If your use case demands to stream data into HDFS, then Flume is a good option. Multiple Flume agents can also be used to collect data from multiple sources. It is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a straightforward and flexible architecture based on streaming data flows. Apache Flume is robust and faults tolerant with tunable reliability mechanisms and many failovers and recovery mechanisms. It uses a simple, extensible Big Data Security model that allows for an online analytic application and data ingestion process flow.

18. **Amazon Glue** – It is a serverless data integration service that makes it easy to discover, prepare, and combine data for analytics, machine learning, and application development. It provides both visual and code-based interfaces to make data integration easier. Users can easily find and access data using the AWS Glue Data Catalog. Data engineers and ETL (extract, transform, and load) developers can visually create, run, and monitor ETL workflows with a few clicks in AWS Glue Studio. It is the way to go if you plan to build your data lakes and data pipelines within AWS infrastructure.

19. **Apache Gobblin** - It is a data ingestion framework designed to handle multiple data sources like rest APIs, FTP/SFTP servers and filers, and load these onto Hadoop. It ingests data from different sources in the same execution framework, and a great choice for extracting, transforming, and loading a large volume of data from various data sources. Has great features such as auto scalability, fault tolerance, data quality assurance, extensibility, and the ability to handle data model evolution. It an easy-to-use, self-serving, and efficient data ingestion framework. A good choice that simplifies common aspects of Big Data integration such as data ingestion, replication, organization, and lifecycle management for both streaming and batch data ecosystems.

20. **Apache Hadoop** – It is a framework for the distributed processing of large data sets across clusters of computers using simple programming models - designed to scale up from single servers to thousands of machines, each offering local computation and storage. With the aid of several components such as the Hadoop Distributed File System (HDFS) providing high-throughput access to application data, YARN for job scheduling and cluster resource management, and a MapReduce paradigm implementation that allows parallel processing of large data sets, among other features, it was/is always one of the first candidates to consider when dealing with big datasets.

21. **Apache Kafka** – It is one of the best options to ingest streaming data into HDFS leveraging low latency and highly scalable data pipelines. Here you can fine-tune the throughput well based upon the requirement. Kafka is a distributed streaming platform. It is capable of building data streaming pipelines and apps. Opensource, horizontally scalable, fault-tolerant, and one of the swiftest in the market, it functions as a messaging agent, and provides a unified platform to handle high data throughput with fairly low-latencies, which enable it to which enable it to handle real-time data feed. Widely used for requirements such as collection of user activity data, device instrumentation, application metrics, logs, and stock markets - ensures high availability of streaming large volume data in real-time.

22. **Keras** – It is an open-source, higher-level deep learning framework written in Python - is a high-level API built on Tensorflow wrapping around functionalities of other ML and DL libraries, including TensorFlow, Theano, and CNTK, but closely tied to that TensorFlow. Data scientists like using Keras because it makes TensorFlow much easier to navigate—which means you're far less prone to make models that offer the wrong conclusions. It is great to build and train neural networks, very friendly and modular, so you can experiment more easily with deep neural networks. It abstracts away the computation backend, which can be TensorFlow, Theano or CNTK, but does not support a PyTorch backend.

23. **Amazon Kinesis** - Kinesis is a cloud-based service designed to handle real-time distributed data streams. It can handle multiple data types and sources like website clickstreams, financial transactions, social media feeds, IT logs, and location-tracking events. A fully managed service, it serves web applications, mobile devices, wearables, and industrial sensors. It is the way to go if you plan to build your data lakes and data pipelines within AWS infrastructure.

24. **Kinesis Data Analytics** – It is based on Apache Flink, an open-source framework and engine for processing data streams. Provides built-in functions to filter, aggregate, and transform streaming data for advanced analytics. It processes streaming data with sub-second latencies, enabling you to analyze and respond to incoming data and events in real time. It is a serverless solution that runs your streaming applications without requiring you to provision or manage any infrastructure. Amazon Kinesis Data Analytics automatically scales the infrastructure up and down as required to process incoming data. It supports building applications in SQL, Java, Scala, and Python, and it is the way to go if you host all your data and ingestion pipelines within AWS infrastructure and you want to extract great insights real-time.

25. **Knime** - It is an open-source platform widely used by data scientists due to its suit of tools for data reporting, mining, and analysis. Its ability to perform data extraction and transformation makes it one of the essential tools used in data science. The 'lego of analytics,' a data pipelining concept for integrating various components of data science. Its easy-to-use GUI helps perform data science tasks with minimum programming expertise. You can leverage its visual data pipelines to create interactive views for the given dataset.

26. **Apache Kylin** – It is an open-source, distributed analytical data warehouse for Big Data; it was designed to provide OLAP capabilities in the big data era - a multidimensional analytics engine. Designed to provide a SQL interface and MOLAP in synchronous with Hadoop to support large data sets. It supports rapid query processing by identifying the star schema, building an OLAP cube from data tables and running query to get results via APIs. You can integrate it with BI visualization tools like Tableau, Power BI, etc.

27. **Elastic Logstash** - It is an open-source data ingestion tool, server-side data processing pipeline that ingests data from many sources, simultaneously transforms it, and then sends it to your "stash" i.e., Elasticsearch. As data travels from source to store, Logstash filters parse each event, identify named fields to build structure, and transform them to converge on a common format for more powerful analysis and business value. If you plan to use it along with Elasticsearch to navigate your data, then it is a good candidate in your list.

28. **MATLAB** – It is a compelling programming and numeric computing platform today used by millions of engineers and scientists to analyze data, develop algorithms, and create models. For data analysis you can leverage a GUI live editor to organize, clean, and analyze complex data sets. A suite of MATLAB apps allows you to interactively perform iterative tasks such as training machine learning models or labeling data. These apps then generate the code needed to programmatically reproduce the work you did interactively – it has prebuilt family of functions for identifying and cleaning sensor drift, signal outliers, missing data, and noise. Its Parallel Server enables you to scale MATLAB programs and Simulink simulations to clusters and clouds. You can develop and prototype your programs and simulations on your desktop with Parallel Computing Toolbox and then run them on clusters and clouds. It is widely used to analyze and extract insights from diverse fields such as climatology, predictive maintenance, medical research, and finance.

29. **Mondrian** – It is part of the Pentaho Analysis Services an open-source OLAP (online analytical processing) server, written in Java. It is an interactive tool with outstanding features and strengths to work with categorical data, large data as well as geographical data. Basically, a general-purpose data visualization tool that consists of interlinked plots and queries. It works with data in standard tab-delimited or comma-separated ASCII files and can load data from R workspaces, but as a push-back it has basic support to work with databases directly.

30. **Apache NiFi** - It is another great data ingestion tool that provide an easy-to-use, powerful, and reliable system to process and distribute information. It supports robust and scalable directed graphs of data routing, transformation, and system mediation logic. Capabilities like link tracking data flow from beginning to end, the seamless experience between design, control, feedback, and monitoring, plus the security features build-in such as encrypted content via secure channels like SSL, SSH, HTTPS, make it a tool you should consider within your list – as it supports running on any JAVA device and is ideal in limited connectivity scenarios.

31. **NumPy** - It provides support for large multidimensional arrays and matrices along with a collection of mathematical functions to operate on these elements. The project relies on well-known packages implemented in other languages (like Fortran) to perform efficient computations, bringing the user both the expressiveness of Python and a performance like MATLAB or Fortran.

32. **Pandas** – It is a library built on top of two core libraries — matplotlib for data visualization and NumPy for mathematical operations, and is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language. Pandas acts as a wrapper over these libraries, allowing you to access many of matplotlib's and NumPy's methods with less code. It introduced two new types of objects for storing data that make analytical tasks easier and eliminate the need to switch tools: Series, which have a list-like structure, and DataFrames, which have a tabular structure.

33. **Pentaho BI** - It is business intelligence (BI) platform that provides data integration, OLAP services, reporting, information dashboards, data mining and extract, transform, load (ETL) capabilities. The Business Analytics Platform (BA Platform) makes up the core software piece that hosts content created both in the server itself through plug-ins or files published to the server from

the desktop applications. It includes features for managing security, running reports, displaying dashboards, report bursting, scripted business rules, OLAP analysis and scheduling out of the box. Commercial, and open-source plug-in projects, expand the capabilities of the server. It runs in the Apache Java Application Server and can be embedded into other Java Application Servers.

34. **Power BI** – It provides a wide variety of connectors, including on-premises data sources and offerings that live in other clouds. Easily connect to, model, and visualize your data, creating memorable reports personalized with your KPIs and brand. Get fast, AI-powered answers to your business questions—even when asking with conversational language. Your team can work together on the same data, collaborate on reports, and share insights across popular Microsoft Office applications such as Microsoft Teams and Excel. Users have the ability to create live dashboards, powerful and plentiful visualizations, and paginated reports. Gartner recognizes Microsoft as a Leader for the fourteenth consecutive year in the 2021 Gartner Magic Quadrant for Analytics and Business Intelligence Platforms.

35. **Python** – A language that grew and evolved significantly, through different set of libraries and extensions, making data science tools and technologies decisions not only limited to databases and platforms - choosing the right programming language for data science is really important. The language currently offers various libraries designed explicitly for data science operations. You can efficiently perform various mathematical, statistical, and scientific calculations using it – its most widely used libraries for data science are NumPy, SciPy, Matplotlib, Pandas, Keras, among others.

36. **PyTorch** - It is framework developed by Facebook and is widely used in research projects. It allows almost unlimited customization and is well adapted to running tensor operations on GPUs (same as TensorFlow). It gives us useful abstractions in certain domains and a convenient way to use them to solve concrete problems.

37. **Amazon QuickSight** – It is a scalable, serverless, embeddable, machine learning-powered business intelligence (BI) service built for the cloud. It lets you easily create and publish interactive BI dashboards that include Machine Learning-powered insights. It uses its super-fast parallel in-memory calculation engine (SPICE) platform to create dashboards that can be accessed from any device, and seamlessly embedded into your applications, portals, and websites. If you are working within the AWS infrastructure it is a good choice, as it can automatically detect AWS data sources, but it can also work with other data sources, such as SQL, Salesforce, etc.

38. **R** – It provides a scalable software environment for statistical analysis and is one of the many popular programming languages used in the data science sector – it is great for data clustering and classification, and various statistical models can be created using it, supporting both linear and nonlinear modeling offering many add-ons for data science like DBI, RMySQL, dplyr, ggmap, xtable, etc. In addition, you can perform data cleaning and visualization efficiently, as R represents the data visually in simple ways so that everyone can understand it.

39. **AWS Redshift** - It can query and combine exabytes of structured and semi-structured data across your data warehouse, operational database, and data lake using standard SQL. All queries of data used a PostgreSQL style of syntax, so understanding basic SQL made it possible to easily grab and transfer data without a lot of specialized knowledge. Redshift works fast with large data sets, and like many of Amazon's products, integrates well with AWS hosted files. Unlike traditional row-based organizational structure, it uses columns. While this approach may not be as strong for transactional querying, for the purpose of data ingestion this resulted in extremely fast and efficient queries. As a result, column-based indexing is particularly well-suited for the purpose of data analytics, particularly those which must be run in real time with large datasets. It is an alternative to consider highly if you plan to leverage your business intelligence and analytics pipelines within AWS infrastructure.

40. **Amazon S3** – It has all the capabilities you are going to need for building a data lake and run big data analytics, artificial intelligence (AI), machine learning (ML), and high-performance computing (HPC) applications to unlock data insights. You can benefit from storing/archiving data at very low costs, by moving on-premises archives to the low-cost S3 Glacier and S3 Glacier Deep Archive storage classes to eliminate operational complexities. It is no brainer if you plan to build your data pipelines within AWS infrastructure.

41. **Samza** - It provides distributed streaming processing. It uses Apache Kafka for messaging and Hadoop YARN for fault handling, processor isolation, security, and resource management. Samza handles restoration of a stream processor's state from outage using snapshotting, i.e., whenever a processor is re-started, it is restored to a consistent snapshot state. If a machine within the cluster fails, it uses YARN to migrate tasks to another machine.

42. **SciKit learn** - It is a general machine learning high-level Python library, built on top of NumPy, to construct traditional models. It features a lot of. It features a lot of utilities for general pre/post-processing of data, efficient tools for machine learning and statistical modeling including classification, regression, clustering, and dimensionality reduction via a consistence interface leveraging algorithms such as support vector machines, random forests, etc.

43. **SingleStore** – It is a distributed, relational, SQL database management system that features ANSI SQL support and is known for speed in data ingest, transaction processing, and query processing - primarily stores relational data, though it can also store JSON data, graph data, and time series data. Early versions only supported row-oriented tables, and were highly optimized for cases where all data can fit within main memory (based on the premise that RAM will be cheaper and cheaper). As the decreases in cost of memory slowed over time, it added general support for an on-disk column-based storage format to work alongside the in-memory row-store. Today, it supports blended workloads -commonly referred to as hybrid transactional/analytical processing workloads- as well as more traditional OLTP and OLAP use cases.

44. **Sisense** – It is a data visualization tool that is used to create dashboards and visualize large amounts of data. From health, manufacturing to social media marketing, you can use it to unlock data from cloud and on-premises, so everyone can analyze data to get interesting insights. A leading AI-driven embedded analytics platform (Sisense Fusion) helps you to infuse intelligence AI-aid capabilities to enhance your analytics.

45. **Snowflake** – It is a is a data warehouse provided as software-as-a-service (SaaS) - built on top of the Amazon Web Services or Microsoft Azure cloud infrastructure, and because of this it offers dynamic, scalable computing power with charges based purely on usage. Users can do data blending, analysis, and transformations against various types of data structures with a single language SQL, and it has a schema-on-read approach to access data in structured and semi-structured formats without having to map them to a predefined structure in advance – supports variety of data file formats including CSVs, JSON, XML, Parquet, Avro, and blend them using SQL language.

46. **Apache Spark Streaming** – Besides it actually utilizes batch processing. Here the ingested groups are simply smaller or prepared at shorter intervals, but still not processed individually. This type of processing is often called micro batching and considered by some to be another distinct category of data ingestion.

47. **Apache Sqoop** – It transfers bulk data between Apache Hadoop and structured data repositories like RDBs. It can handle incremental loading of a table or a free form SQL query. It saves jobs, which can then be run multiple times to import updates made to a database since the last import. These imports can be used to populate tables in Hive or HBase. If your use case is to migrate data from the RDBMS to HDFS, then it can help you with both batch and incremental load.

48. **Apache Storm** - It is another Apache distributed streaming product, which processes unbounded data streams and is compatible with most programming languages. It claims a processing speed of one million tuples per second per node and can be integrated with in-use queuing and database processes. Handles repartitioning of streams between stages of computation on a required basis. Its engine uses a lean threading model and a lock-free messaging methodology and backpressure to handle overloading. Storm is a useful adjunct to Enterprise Hadoop, and YARN & Slider, adding value to real time analytics capability and machine learning processes.

49. **Azure Stream Analytics** – It is a real-time analytics service that is designed for mission-critical workloads. With it you can build an end-to-end serverless streaming pipeline with just a few clicks, and go from zero to production in minutes using SQL —easily extensible with custom code- and built-in machine learning capabilities for more advanced scenarios. It helps you running your most demanding workloads offering a hybrid architecture for stream processing with the ability to run the same queries in the cloud and on the edge. It supports building applications in familiar SQL syntax, extensible with JavaScript and C# custom code, and it is the way to go if you host all your data and ingestion pipelines within Azure infrastructure and you want to extract great insights real-time.

50. **Azure Synapse Analytics** – Build on top of Apache Spark Engine, this service that brings together data integration, enterprise data warehousing, and big data analytics. It gives you the freedom to query data on your terms, using either serverless or dedicated resources—at scale. You can leverage more than 95 native connectors to build ETL/ELT processes in a code-free visual environment to easily ingest data. Relational and nonrelational data can be simply combined and you can easily query files in the data lake with the same service you use to build data warehousing solutions. It is an alternative to consider highly if you plan to leverage your business intelligence and analytics pipelines within Azure infrastructure.

51. **Syncsort** - It allows collection, integration, sorting, and distribution of data in a swift timeframe, using minimal resources. It deploys on Hadoop, Splunk, and the cloud. The data application design needs to be done just once thereafter it can be deployed on any platform: Windows, UNIX & Linux, or Hadoop; either on-premises or on the cloud. Ironstream for Splunk enables the processing of huge volumes of machine data streams from the mainframe.

52. **Tableu** – It is business intelligence platform that helps you see and understand your data. You can work with Big Data or any data - from spreadsheets to databases to Hadoop to cloud services. The BI tool's maturity provides a wealth of community knowledge, a vast number of visualizations, and a wide array of data connectors. Your dashboards are easy to share, just with a few clicks and they are live on the web and on mobile devices. Additionally, it also provides some ETL capabilities, so if those can cope with your ingestion and transformation needs you might not require an additional tool. Unlike other tools like Power BI and QuickSight, it is not tied to a specific cloud offering.

53. **Talend** - It is a popular suite of tools. They aim to provide a full top-to-bottom data analysis framework. Part of this platform includes an ETL tool. Easy to use for data ingestion, or "extraction" segment with its ETL functionality, and provides an easy-to-use interface. Most functionality is handled by dragging and dropping nodes or modules into a workspace. A suite you should consider due to its simplicity – you don't need to be an advanced data science professional to use it.

54. **TensorFlow** - It is widely used with various new-age technologies like data science, machine learning, artificial intelligence, etc.- easy to use as it is written in Python, this library can be used for building and training data science models, in addition to take data visualizations to the next level. More library than a framework, all operations are pretty low-level, and you will need to write lots of boilerplate code even when you might not want to – its name comes from using an N-dimensional array as its data type, which is also called a tensor. Trained models can be deployed across various devices.

55. **Teradata Vantage** – It is a connected multi-cloud data platform for enterprise analytics that unifies everything—data lakes, data warehouses, analytics, and new data sources and types. Supports all common data types and formats, including JSON, BSON, XML, Avro, Parquet, and CSV, among others. It can be deployed on public clouds (such as AWS, Azure, and Google Cloud), hybrid multi-cloud environments, on-premises with Teradata IntelliFlex, or on commodity hardware with VMware. The cost model is pay-as-you-go, with the platform delivered as a service approach, and thanks to its deployment options customers to easily connect to and analyze data within low-cost object stores such as Amazon S3, Azure Blob, ADLS Gen2, and Google Cloud Storage. Has lot of data governance and compliance features in place.

56. **Theano** – It is an open-source project released under the BSD license and was developed by the LISA (now MILA) group at the University of Montreal. Specifically designed to handle the types of computation required for large neural network algorithms used in deep learning, is a compiler for mathematical expressions in Python - knows how to take your structures and turn them into very efficient code that uses NumPy, efficient native libraries like BLAS and C++ to run as fast as possible on CPUs or GPUs. Better to use a high-level wrapper like Tensorflow, or even Keras, than dealing directly with it unless specifically needed for performance.

57. **Trifacta** - It is a tool used within data science for data cleaning and preparation. A cloud data lake that includes a mix of structured and unstructured data can be cleaned using it. The data preparation process is significantly paced via this platform as compared to other platforms. You can easily identify the errors, outliers, and other anomalies in the dataset, and prepare data in short time across a multi-cloud environment. Finally, the data visualization process and data pipeline management can be automated.

58. **Wavefront** - It is a hosted platform designed to ingest, store, visualize and issue alerts on metric data. It can ingest a large volume of data points per second. Its stream processing technique enables it to manipulate large volumes of data and it provides a 360-degree view across the IT infrastructure.

59. **Xplenty** – It is a complete cloud platform for building data pipelines, and being elastic and scalable can handle deployments, monitoring, scheduling, security, and maintenance. It provides features to integrate, process, and prepare data for business intelligence. It has coding, low-code, and no-code capabilities – the last one letting anyone create ETL pipelines. Has an API that provides advanced customization and flexibility. An intuitive graphic interface that will help you to implement ETL, ELT, or replication easily.

60. **Yellowbrick** – It is a is a modern, MPP analytic database designed for the most demanding batch, real-time, interactive, and mixed workloads. Their product group continuously implements the latest advances in software and hardware. Supports bulk load/unload data in parallel at line speed and ingest streams at millions of rows/secs, with no impact on other workloads. Based on ANSI SQL syntax and PostgreSQL front end for compatibility with enterprise BI and data motion tools, it can Run queries in milliseconds on billions of rows of data, with no need for indexes, partitions, or query tuning. It can be deployed to private data centers, public clouds, and on-premises or at the edge.

08.

How can SOUTHWORKS help?

ABOUT SOUTHWORKS

SOUTHWORKS is the global software development firm people turn to for their most complex, high-profile projects. Our **"Own it, Bring it, Prove it"** approach lets you **Make Everything Right™** – without the endless handholding and do-overs outsourcing is known for.

SOUTHWORKS global team of remote engineers bring the dev-intensity to your organization, delivering quick strategic wins that help you accelerate, grow, and scale.

It's about thinking beyond the reqs, find the fastest way to the best outcomes, and always keeping you in the know. That's why companies such as Amazon, Microsoft, 7-Eleven, Discovery, Twitch, BBC, and even the Olympic Games count on SOUTHWORKS to Make Everything Right™

INDUSTRIES SOUTHWORKS CONTRIBUTES TO

SOUTHWORKS works across a number of verticals, from helping companies within the big-data industries (those who are currently reaping the benefits of processing and analyzing their huge amounts of data constantly) such as healthcare, manufacturing and retail, financial services, media and entertainment, and sports.

But we also bring helping hands to smaller-data industries – those who are not constantly investing in data efforts, as their data catalogs are smaller, or they don't need a constant updated feed, and perhaps these kinds of solutions were not so in-demand or, in some cases, they are new to them - such as government, insurance, education, and hospitality among others.

WHERE SOUTHWORKS HELPED ITS CUSTOMERS TO ALWAYS BE ONE STEP AHEAD:

FURTHER READING

SOUTHWORKS has worked with start-ups, scale-ups and world-renowned brands across the world to solve some of their toughest dev challenges related to data. If you've enjoyed reading this paper and want to learn more about other data projects we've worked on, some of the other topics we've hit on in this paper such as ML, AI, Sentiment Analysis, or any of the other super cool tech projects we have the privelage to be part of, head on over to our blog where our team detail their learnings.

SOUTHWORKS Tech Edit

IF YOU'VE ENJOYED THIS, CHECK OUT:

→ Our Big Data work

Our work in AI ←

Find out how you can start solving your toughest dev challenges, fast.

Contact SOUTHWORKS

Follow us on