Glossary for sales professionals and other non-technical roles requiring a basic knowledge of technical concepts.

Author: Abdelwahid Benslimane
Contact: wahid.benslimane@gmail.com
Version: 5.1
Date of publication: 01/04/2023

This document is primarily intended for sales professionals and other non-technical roles working in the data industry. Its primary purpose is to provide them with foundational knowledge and conceptual understanding of the various products and technical notions that are frequently considered complex and likely to be encountered during the course of their duties. Subsequent versions of this document will be released in the future, and I would be delighted to receive any questions or feedback you may have.

Index

[Security] SSL certificate: SSL stands for Secure Sockets Layer. An SSL certificate is an object that is used to secure communication across a network between a service provider (for example a website) and a client of this service (for example a web browser such as Chrome). The SSL certificate is mainly composed of 2 elements, one called private key, and the other, public key. In order to explain the role played by each of the elements I have just listed, I will use an allusion to a real-life situation that is unlikely to occur but that makes it much easier to understand.

Let's say you want to receive secret communications from several people. You will create boxes that lock with one of the keys and open with the other. For example, if the box is locked with the public key, only the private key can open it and vice versa. You will distribute duplicates of the public key to each of your interlocutors who will have to put their message in the famous box and close it with the public key.

As a box closed with a public key can only be opened by the holder of the private key, only you will be able to open the box because you are the only one who has the private key.

If you want to send a message to someone and want to guarantee that it is from you, then you can use a box and close it with the private key. If the box opens with a public key, it proves that you sent it, because if it was closed with a copy of the public key, another copy of the public key would not open it. However, the message cannot be confidential in that case because anyone with a public key will be able to open a box locked with the private key.

In fact, each person who wants to receive confidential messages will have to issue his/her own set of keys (one public and one private) and distribute copies of the public key to all the people from whom he or she must receive confidential messages stored in boxes that can be closed and opened according to the same principle.

This is roughly how the SSL certificates are used, for either guarantee the confidentiality or the authenticity of the data transmitted over a network.

[Data/Big Data] Relational and NoSQL databases: A relational database is used to store information that is important for the proper functioning of your organization, and that is expected to remain current and up-to-date, or in other words, accurate at the time it is accessed. The relational aspect comes from the fact that data of a different nature will be stored in separate storage areas, called tables, and that data in one table can refer to data in another table. The interest of this way of proceeding is that it allows to save disk space and that the information can be updated more easily and in a reliable way. This last aspect is called data consistency.

Let's take a simple example. You want to keep track of sales history and identify the salesperson responsible for each sale. A sale is characterized by the amount of the sale, the date, the customer, and the identity of the salesperson. You have 3 entities, the sale, the customer, and the salesperson. You will therefore create 3 tables, one per entity.

As the sale is supposed to have information on the customer and the salesperson, the corresponding table will have, for each sale, a link to the salesperson who made the sale in the salesperson table and a link to the customer concerned in the customer table. This way, if several sales are made to the same customer, or a salesperson makes several sales, you don't have to constantly report the same information such as customer name, billing address, salesperson name, etc. This information is only stored once and not as many times as there are sales. Also, if you find out that you have entered the wrong billing address for a customer, you can simply change the address in one place, in the customer table.

Relational databases have a major flaw, however. Because the information related to an entity can be scattered in different tables, consulting the whole data can be very expensive because the relational database management system (RDBMS), which is the orchestra conductor that allows to organize the data, will have to perform operations to retrieve and gather all the necessary information. Indeed, if you decide to print all the sales with the name of the salesperson and the name of the customer, the RDBMS will have to navigate each time between the salesperson, customer, and sales tables. For your information, the language that allows you to write and read in the database is called SQL. SQL stands for Structured Query Language. This is used to give instructions to the RDBMS.

Some examples of RDBMS: MySQL, PostgreSQL, Oracle, Microsoft SQL Server etc.

A NoSQL database has the disadvantage of not having the advantages of relational databases, and the advantage of not having their disadvantages. Indeed, to take the previous example, if we had to store sales in a NoSQL database, then each object (or document) of type sale will contain all the information that characterizes it. In other words, in the sale object itself, we will find the name of the customer, the billing address, the name of the salesperson, etc. And if several sales are made to the same customer or by the same salesperson, this information will be copied each time. This is called redundancy.

The main advantages offered by NoSQL databases are:

- a fast reading of the data;

- the implementation of parallel processing (distributed computing) for the realization of a task that can be divided into distinct operations and sequential phases. This allows a saving of time.

The main disadvantages are:

- a more laborious update of the data, and as a result, it is difficult to envisage having consistent data. Indeed, if you wanted to correct the billing address of a customer, you would have to do it in each object representing a sale to this customer, instead of making the correction only in one place as in a relational database;

- the data is hierarchical. In fact, the data are written in files that the non-relational DBMS will have to go through to retrieve information. If the salesperson's name is at the very end of a sales object/file, it will have to go through all the other information before arriving at the salesperson's name if we are interested in this particular data. This can result in a certain cost in terms of reading time if you browse many documents that are long enough to retrieve the salesperson's name each time.

NoSQL databases are suitable, for example, if you want to model catalogs, because they are designed to be accessed more than modified. NoSQL is a contraction of "Not only SQL", which means that this kind of database offers an alternative to relational databases.

Some examples of non-relational DBMS: MongoDB, Cassandra, Couchbase etc.

There is no standardized language for interacting with a non-relational database. This will depend on the editor/DBMS. SQL cannot be used in any case (Cassandra has a language that looks a bit like SQL called CQL, but it is not SQL)

[Data/Big Data] Structured data: Structured data are data that meet the requirements of a pre-determined format and could eventually be stored in Excel spreadsheets or relational databases.

The number, name, and nature of the characteristics define the standard by which the data is formatted. The values corresponding to each characteristic should be of a known type (e.g. date, numeric value, string...) but not necessarily understandable by a human.

For example, if in your information system, cars are always defined by itheir brand, model name, engine capacity and date of manufacture, you are then handling structured data.

Structured data can be represented in many file formats: Excel file, CSV file, XML file, JSON file etc. Note that semi-structured data can also be represented in XML and JSON formats.

[Data/Big Data] Semi-structured data: Semi-structured data represent complex objects (in the computing sense of the term) that can have optional components, the "schema" of the data is not rigid, which offers greater freedom in the way the objects are defined. However, these data remain partly structured because they contain a set of characteristics that are like markers for the different information they contain.

This is a difference from the relational model, where the number of attributes and the organization of the schema is decided in advance.

The easiest way to define semi-structured data is by example.

Suppose you want to model the menus ordered by customers in a restaurant. Depending on the customer, the menu will contain different dishes in a different number. The different dishes will also be defined by different ingredients. If you compare two menus, you will see two different data structures even though both model the same type of object.

The preferred formats for representing semi-structured data are XML and JSON. NoSQL databases are well suited to store semi-structured data.

[Data/Big Data] Unstructured data: Unstructured data are data that can neither be considered as structured nor as semi-structured data. In other words, it is raw data that is not organized. Rather than trying to define precisely unstructured data, it is better to know the most commonly used types of unstructured data: text, image, video and audio files.

[AI] AI, machine learning and deep learning: Deep learning is a form of machine learning which itself is a form of artificial intelligence. These are sets of techniques based on algorithmic and statistical methods that process large sets of empirical data in order to extract patterns that will put them on the right track to accomplish the tasks for which they have been implemented with the greatest possible efficiency and automatically. This is called training.

It should be noted that there is another form of artificial intelligence which is based on formal logic, and which does not require data to be trained. This one will not be further defined here.

Both classical machine learning algorithms and deep learning algorithms can be used for regression (prediction of a numerical value) tasks or classification tasks. Among the regression tasks we can mention for example the estimation of the price of a real estate according to its characteristics or the evolution of the price of a product according to the history of the prices (time series), or the prediction of sales to manage the supply of a stock.

Examples of classification tasks are the categorization of customers according to their characteristics or the recognition of objects in images.

But there are many other use cases such as medical diagnosis, prevention of incidents, spam or malware detection etc., as well as other applications more specific to deep learning such as synthetic data generation, natural language processing (NLP) to analyse a text or generate text as for a chatbot etc., all without human intervention.

Let's now see what differentiates classical machine learning from deep learning.

First of all, classical machine learning algorithms can only be trained with structured data. These models are generally less data and computationally intensive than deep learning models and can provide very good results for most of the use cases of the companies that use them. The advantage of using structured data is that it can often be processed to increase the efficiency of the algorithms it will be used to train.

We start talking about deep learning as soon as the models used involve neural networks. There are all kinds of neural networks, here are the most classical ones which are the basis of the most modern models: multi-layer perceptron (or artificial neural network), convolutional neural network, recurrent neural network (like the LSTM or the GRU). It is of course not necessary to know them in detail if you are not in charge of their development and training.

Deep learning models can be used to process structured or unstructured data (depending on the need and the model). When it comes to processing unstructured data such as text files (NLP) or images/videos for example, it is deep learning that is used. Deep learning models are complex and require much more data and computing power than machine learning models.

[AI] Supervised, unsupervised and semi-supervised learning: Supervised learning seeks to build a decision model from data for which a label is available. For example, if we train a model with images having a label to define if it is the picture of a cat or a bird, so that this model can then automatically extract this information from the pictures (which were not used for its training) that will be submitted to it and that will not have a label, then it is a typical case of supervised learning.

Regression is another type of supervised learning method that aims to understand the relationship between the information to be predicted and other available information.

In the case of unsupervised learning, no supervision information (no label, no information to be predicted) is available, e.g. we do not know the class to which the data used for training belong. Often, we do not even know which classes are relevant to characterize these data. The model obtained by unsupervised learning is generally used as a descriptive model and helps to better define a decision problem that will be addressed later.

We speak of semi-supervised learning in the case where the supervision information is only partially available (for example for a small part of the training data). Some probabilistic approaches allow to use data without supervision information in order to improve the decision model compared to the exclusive use of data for which this information is present.

[AI] AutoML: Machine learning in general offers many methods to deal with the same problem, and it can be laborious to test all of them to try to obtain the maximum efficiency, especially since the models are also characterised by a set of hyperparameters that, unlike the parameters, have to be adjusted manually by the data scientist (they are not determined automatically by the learning algorithms whose behaviour depends on them).

The learning algorithm actually aims to determine the parameters of the model, not the hyperparameters. There is no need to detail further the notion of parameters and hyperparameters but be aware that a model is characterized by both and that the hyperparameters must be adjusted manually.

Depending on the model chosen and the combination of hyperparameters, the performance obtained will be more or less high.

Given the number of possibilities, it can quickly become difficult for a data scientist to attempt to build a model that is the most satisfactory possible and this is precisely where autoML comes into play.

AutoML solutions, as the name suggests, will automate the creation and training of different models, and then show you the results of the selected approaches, allowing you to compare and choose.

All you have to do is submit training and test datasets and the autoML solution will do the rest. It is with the test dataset that the performance of a model is measured to see how well it generalises, i.e. how well it performs on new data that it has not been trained with.

[Security] IdP (Identity Provider): An IdP is a solution that will store and manage digital identities, not only of human resources within an organisation, but also of IT resources (a server, a printer etc.). When you join a company, the IdP administrator adds an entry in the appropriate database within the IdP solution and will provide you with a login and password so that you can authenticate yourself to the IdP afterwards.

For example, did you know that when you log on to your company computer, you are likely to authenticate to a very common IdP called Active Directory (Microsoft's solution) at the same time. This is what then allows you to access or not access certain internal resources also registered within the IdP such as SharePoint for example.

Another popular IdP is Okta (also a service provider).

It is the security policies defined within the IdP that will determine that you must change your password when you first log in, and that you have to change it at regular intervals thereafter.

The IdP is intimately tied to the notion of SSO also defined in this document.

[Security] SSO (Sigle Sign-On) and SAML (Security Assertion Markup Language): SSO is a service that allows users to log in to all their applications hosted in the cloud by authenticating only once to an identity provider. It's a bit like swapping all your keys to open all your doors for one key that opens them all.

SAML is a language, or rather a communication standard, that allows different services to exchange information about a user. SSO and SAML combine to make your life easier in the way described by the scenario below:

1) A user accesses the Okta authentication portal and enters a login and password

2) If Okta is the main IdP, then it will check its database to see if the user exists and is allowed to log in.

It is also possible that Okta is synchronised with the Active Directory of the user's company and that the identities are in fact stored centrally on Active Directory (AD) which is in this case the main IdP. In this case Okta will check with the AD that the user is legitimate and will authorise him/her to connect if so.

3) On the Okta portal, the user clicks on the icon representing the application to which he or she wants to connect. Okta creates an authentication token containing the user's identity and communicates it to the application using SAML.

4) The application checks the validity of the SAML request, to see if it is legitimate, and the content of the token before allowing the user to have access.

If users have access to 50 applications via Okta, they do not have to create and manage as many logins and passwords. They only need to authenticate to Okta, and it is Okta, as the SSO service provider, that will authenticate them to all the applications they want to access.

To draw a parallel with the administration, the AD (or Okta)/IdP plays the role of the State, Okta/SSO service is the department of the town hall that issues passports, the passport is the authentication token and the applications hosted in the cloud are all the countries you could go to. SAML is all the steps and protocol you follow from the moment you want to enter the boarding area at the departure airport until your passport is checked by the customs officer at the arrival airport.

[Software and Infra] API (Application Programming Interface): An API is a way for an application to allow other applications or services to interact with it and obtain data by sending requests through a set of methods and protocols that make up the API.

This is what allows for the integration of a third-party service into an application. For example, if a company using Qlik Sense wants to develop its own graphical interface to administer this solution, they will need to use the Qlik Sense API. The company can then build its own interface to reload applications or track utilisation etc.

An API also allows some actions to be automated by calling its methods in small programs called scripts developed to perform tasks in an automated way rather than manually.

[Software and Infra] Docker and Kubernetes: Docker is a solution that facilitates the deployment and the portability of applications.

This solution actually encapsulates an application within an isolated virtual environment called container that contains everything the application needs to run.

Docker is what allows the containers to run, and the containers are the environments that allow the applications to run.

So, if you wish to migrate an application that runs on Docker to another server, all you have to do is export the associated container and install it on another server, as long as Docker is installed on it.

Kubernetes is often combined with Docker to allow for easier management of containers.

It is in fact a container orchestrator. In particular, the high availability of applications can be managed very easily with Kubernetes, which will create several instances of the same application and then distribute them on different machines called nodes. Kubernetes will then monitor the state of the containers and nodes and do the necessary to ensure that an application is always available (for example by restarting the container if it is stopped or by replacing a node if it is no longer responding).

[Data/Big Data] Data Fabric and Data Mesh: Data Fabric and Data Mesh are two different concepts related to data architecture.

Data Fabric approach advocates for centralized management and a unified view of data from multiple sources (various databases, flat files, data from sensors, etc.). The data don't need to be physically in the same place as an abstraction layer (a solution in concrete terms) can allow for accessing and managing the different data seamlessly regardless of their location.

On the other hand, Data Mesh is a concept that is somewhat opposite to Data Fabric but not totally incompatible. Data Mesh aims for each department within an organization to store and manage its data autonomously, and an application or service layer that would be added on top of this decentralized architecture would allow for data convergence and collaboration between different business units when needed.

[Software and Infra] Apache Spark: Apache Spark is a solution that allows for parallel processing of very large volumes of data by distributing the workload among multiple nodes in a cluster.

The processing is divided into multiple tasks that can be executed simultaneously by separate computation units, each responsible for a subset of data that has been grouped based on specific criteria beforehand. It should also be noted that the data is loaded into memory, which contributes to good performance.

An example can better illustrate the principle of Apache Spark and the concept of distributed computing.

Let's say you want to sum up all the numbers from 1 to 10 using an Apache Spark cluster with 2 nodes. You would command Apache Spark to divide the work into multiple phases and sub-tasks, by summing up the even and odd numbers separately, and finally summing up the two results obtained previously.

Once Apache Spark has received the instructions, it will form the group of even numbers ({2, 4, 6, 8, 10}) and the group of odd numbers ({1, 3, 5, 7, 9}) and assign each node the task of adding up the numbers from one of the two groups. Apache Spark will then add up the results of the two previous operations (30 + 25). All of this will be done in an optimized and very fast manner.

When applied to a more complex processing on a much larger volume of data, the use of Apache Spark allows for a considerable time gain compared to non-distributed processing.

[Vendors solutions] Databricks: Databricks is a cloud solution based on Apache Spark that allows different teams of technical experts working on data, such as data engineers and data scientists, to collaborate effectively and securely in real-time, which sets it apart.

It must be noted that is not a data analysis tool for business users like Qlik Sense or Power BI, but rather a data processing tool that sits upstream of the use of this data by the business.

The various tools provided by Databricks facilitate collaboration between teams and within the same team. For example, data engineers can create and manage data pipelines and quickly allow data scientists to access them. Data scientists can then process the data using pre-built machine learning models available on the Databricks platform and work collaboratively using various platform features such as a user-friendly interface for code sharing and instant messaging.

The data can then be exported and transferred to the business side in an automated way if needed thanks to the Databricks API. There are also connectors to allow different analysis tools such as Qlik Sense and Power BI to import data from Databricks.

[Vendors solutions] Snowflake: Snowflake is a relational database available on the cloud platforms AWS (Amazon Web Service), Microsoft Azure, and GCP (Google Cloud Platform). It was designed specifically to integrate with cloud technologies and enable high availability and simplified scalability on these environments. For these reasons, it is said cloud-native.

Additionally, Snowflake is column-oriented. The most important thing to know regarding this aspect is that it allows for more efficient data compression and improved performance for certain types of queries.

Conceptually, Snowflake is organized into three layers:

1) the storage layer, which is responsible for storing the data as its name indicates,

2) the processing layer, which handles queries to the database,

3) the services layer, which is the interface between users and the other two layers. It manages authentication, data access, and resource allocation, among other things.

This separation into three layers facilitates horizontal scalability (i.e., platform expansion) and optimized resource management.

[Data/Big Data] Data warehouse: A data warehouse is a database that contains data from various sources (transactional databases, ERP, flat files, etc.) and is designed to facilitate in-depth analysis for decision-making. While a data warehouse is hosted on a relational database management system, the data warehouse model is typically not relational, but dimensional. The most important thing to know about it is that the dimensional model is implemented to execute analytical queries more easily and with better performance, such as the getting number of sales made, the total amount of sales to a customer or group of customers, etc. A data warehouse can be implemented on various database management systems such as Microsoft SQL Server, Snowflake, Amazon Redshift etc.

[Data/Big Data] Data lake: A data lake is a collection of data in the form of files hosted on platforms optimized for storing and retrieving large files such as Amazon S3 (Simple Storage Service), Microsoft Azure Data Lake Storage (ADLS) or Databricks. The data can be structured, semi-structured, or unstructured. Data lakes are generally set up to allow data scientists to explore data quickly and easily.