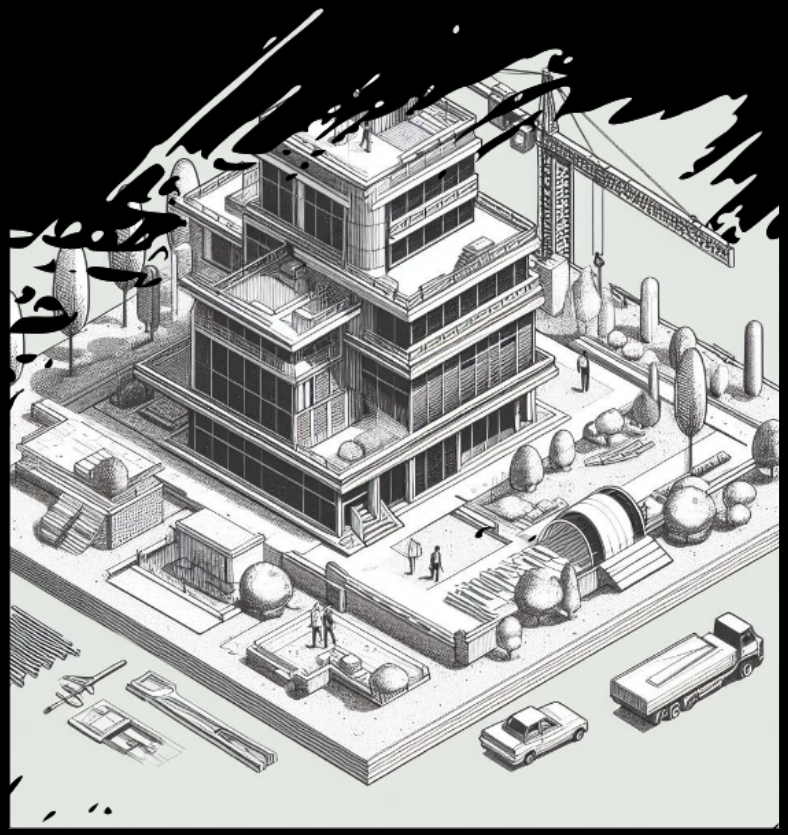
LES CAHIERS DATA DE JÉRÔME



Data-Framework

Volet 2 :

Déployer l'Intelligence Artificielle à l'échelle de l'organisation.

Jérôme Toscano, CDO

Juillet 2024

INTRODUCTION - DÉPLOYER L'IA GÉNÉRATIVE ET LA DATA-SCIENCE À L'ÉCHELLE DE L'ORGANISATION

À travers ce livre, je souhaite vulgariser des notions qui peuvent paraitre très abstraites pour de nombreuses personnes.

Il est clair qu'on peut se dire : « tant que ça marche, pas besoin de comprendre comment ça marche ». De mon point de vue, cela est une grave erreur car il convient de considérer que l'IA et la data science en particulier sont des domaines pour lesquels une organisation, peut importe sa taille, doit comprendre les concepts fondamentaux et les impacts de toute nature sur son organisation.

L'objet n'est pas de faire de vous des experts en la matière, mais de démystifier les croyances qui, encore aujourd'hui, continuent de graviter autour de ces sujets, encore davantage à l'ère de l'AI générative, propice aux illusionnistes de toute sorte.

L'IA ne sert pas seulement à « prédire » ou « générer » des choses, mais grâce aux connaissances que je vais vous donner ici, laissez moi vous prédire que vous ne considérerez plus ces sujets comme des « boites noires », mais comme de vraies opportunités qu'il vous faut saisir, maintenant.

Alors comment faire en sorte d'utiliser ces approches à bon escient ? À quel moment de votre projet data devez vous aborder ces sujets ? Comment faire en sorte que vos projet d'IA et de Data Science passe à l'échelle et génèrent réellement de la valeur ? Comment éviter les dérapages (pas seulement financiers) dans la mise en place d'une telle démarche ?

Alors par où commencer ?

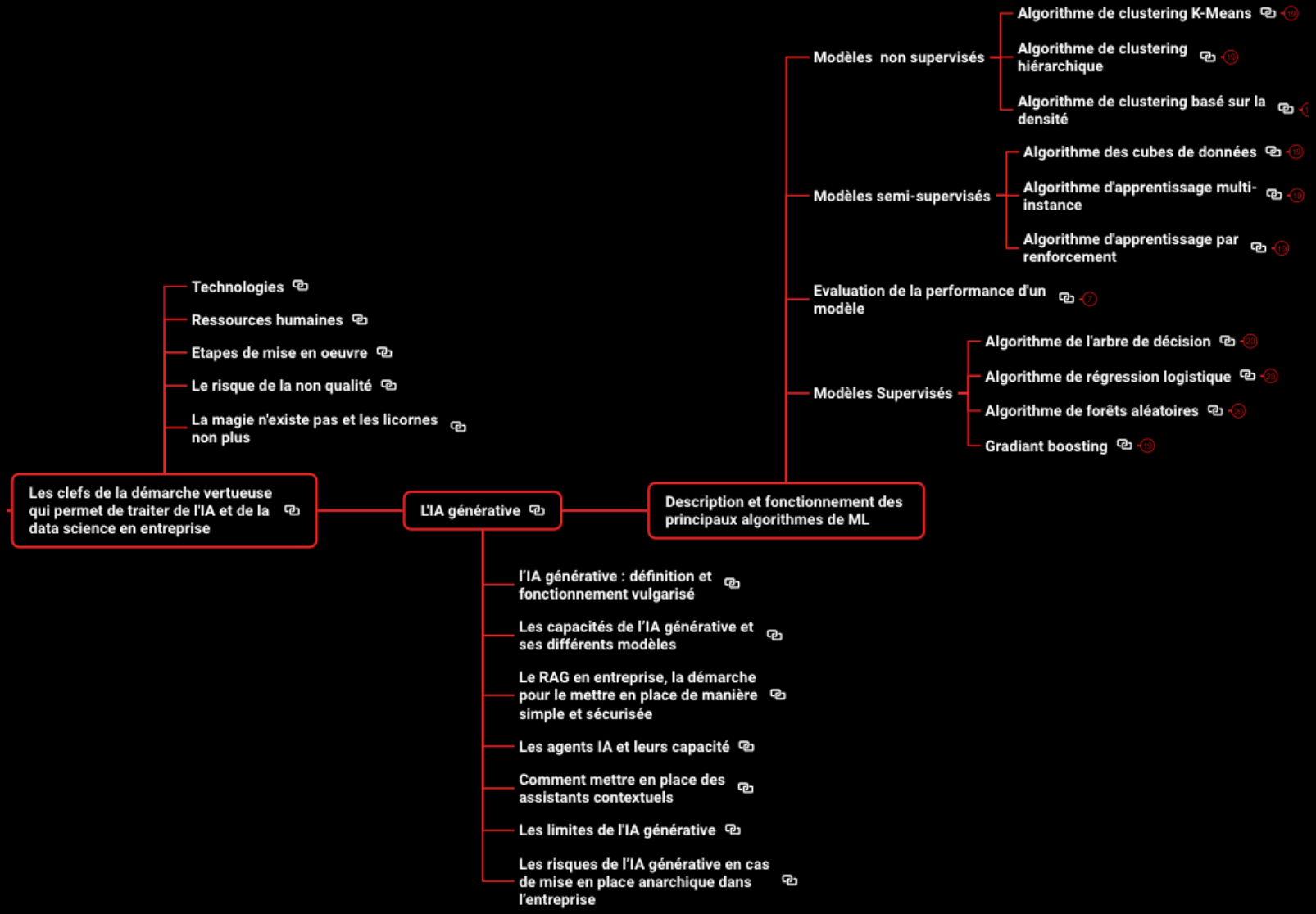
Nous allons ici aborder les points clefs à respecter pour mettre en oeuvre une stratégie « IA GEN / Data Science ». Comment déployer l'IA générative de manière pragmatique et efficace ? (je peux d'autant plus vous en parler que je l'ai fait dans la vraie vie) Nous aborderons aussi les principes de fonctionnement des différents modèles de Machine Learning (ML) qui sont les ingrédients indispensables à vos futurs projets de data-science.

Comme toujours, tout cela sera abordé de manière ludique et synthétique pour votre totale compréhension des sujets.

Bienvenue dans ce livret 2 du **Data-Framework** .

Sommaire

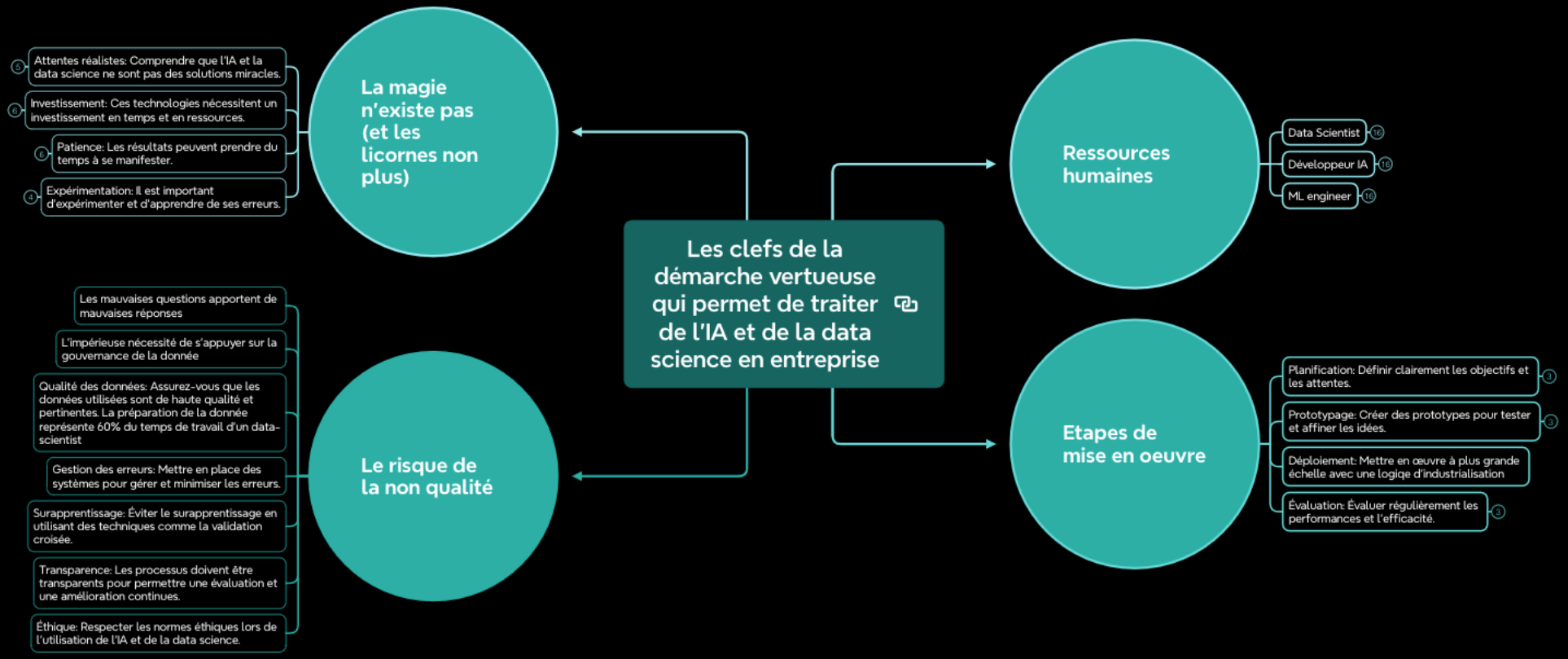
Synthèse des sujets abordés dans cet ouvrage



Premiere partie :
Les clefs d'une démarche vertueuse

Mettre en oeuvre une stratégie d'IA / Data science dans une organisation nécessite de se poser au moins trois questions :

Quelles ressources (technologiques et humaines) ? Quelles étapes de mise en oeuvre ? Comment identifier les risques et éviter les effets d'enthousiasme inappropriés ?

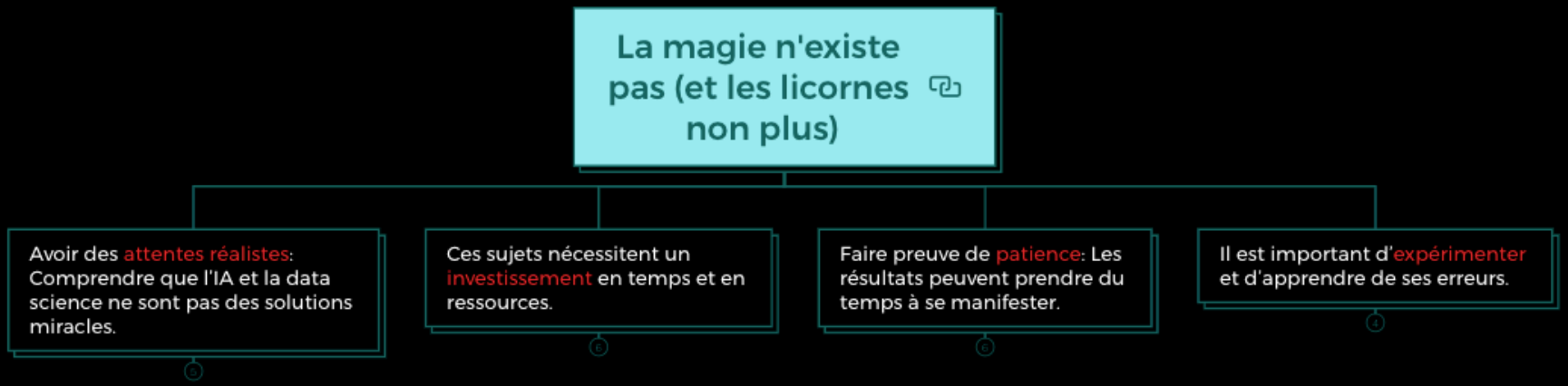


LA MAGIE N’EXISTE PAS (ET LES LICORNES NON PLUS)

Et oui, il n’y a pas de magie dans la mise en place d’une approche IA/DS. Espérer un résultat pertinent et rentable à court terme lorsque l’on jamais initié une telle démarche est l’équivalent de trouver une Ferrari à vendre avec 1000km au compteur à 2000 €. (oui il y a un loup)

De manière simple, réfléchissez sur ces 4 points :

- L’investissement nécessaire
- L’aspect expérimentation
- Avoir des attentes réalistes
- Avoir de la patience



Regardons cela en détail.

AVOIR DES ATTENTES RÉALISTES

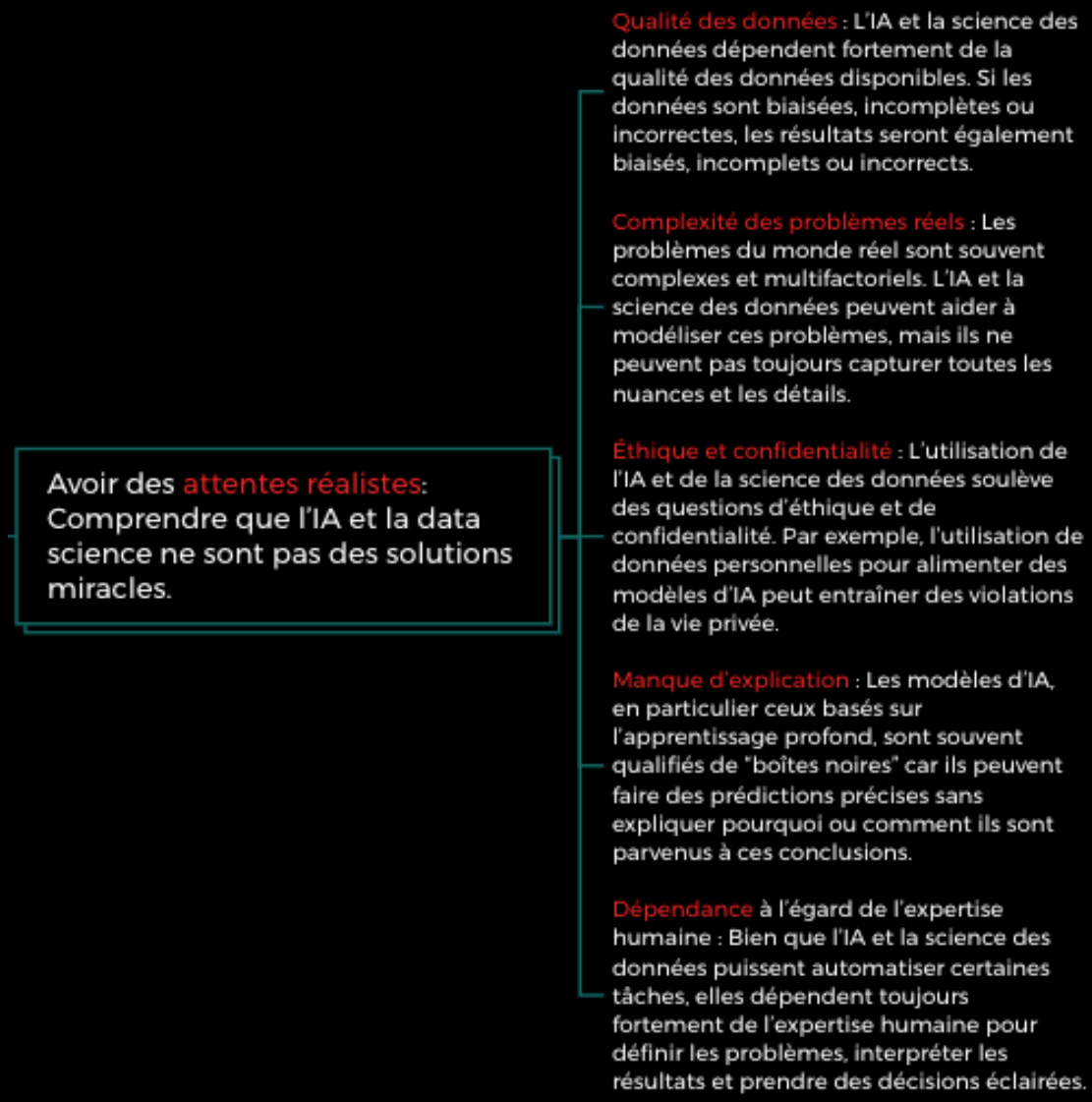
Avant de vouloir jouer aux apprentis sorciers, il est nécessaire de considérer certaines choses et de ne pas foncer tête baissée dans le mur.

Comme dans la plupart des cas, les projets de mise en oeuvre d'IA nécessitent des pré-requis, dont le premier est l'impérieuse nécessité d'avoir évalué (ou à minima d'avoir pris conscience) de la **qualité des données** sur lesquelles vous allez travailler. Tant en terme de qualité qu'en terme de véracité, il vous faut savoir le niveau de l'information que vous soumettrez à votre approche.

Mettez vous également immédiatement dans une démarche **d'industrialisation** et répondez à des problèmes concrets qui, si vous les résolvez, apporteront une valeur immédiate (en temps ou en argent, ou les deux :)).

Ne sous estimez pas non plus la **vulgarisation** et votre capacité à expliquer un résultat. Vous devez savoir évaluer en permanence vos résultats et le rapport bénéfice/coût (autrement appelé R.O.I) qu'ils représentent.

En d'autres termes, ne soyez pas trop ambitieux et ne vendez pas du rêve à vos interlocuteurs. Savoir dire qu'on s'est trompé est souvent la première pierre d'un édifice bien plus robuste.



L'INVESTISSEMENT NÉCESSAIRE

Afin de faire de votre démarche un succès, il va falloir considérer plusieurs choses :

Le temps est le premier ingrédient. Les démarches d'IA et de data science sont longues, et nécessitent de nombreux échecs pour produire des résultats concluant.

La seconde est que vous possédiez un minimum de connaissances sur le sujet. L'investissement dont je parle ici est donc votre propre **acculturation** (qui nécessite également du temps, et l'objet de ce livre au passage…). Faute de quoi, lorsque vous devrez faire des arbitrages ou des choix, vous allez trop souvent vous en remettre à la parole d'un expert en expertise dont vous ne comprendrez ni les orientations, ni les risques qui sous tendent ses orientations.

Avoir un socle technique robuste est également un investissement par lequel vous devrez passer. Je ne parle pas ici d'un vulgaire POC sur l'ordinateur boosté de votre data-scientist, mais d'une réelle **infrastructure** pensée et réfléchie pour l'optimisation de l'exécution des modèles de ML ou encore la réduction des couts liés à un système de RAG (*spoiler : ce qui marche sur son pc est très loin d'être optimal dans un cadre de production*).

La **ressource humaine** est également nécessaire. (bah oui c'est un métier :)), nous en parlons plus tard dans ce livre. Comme pour tout investissement, pour l'obtenir il va vous falloir convaincre, démontrer des premiers résultats et avoir une vision éclairée des marches à gravir.

FAIRE PREUVE DE PATIENCE

Nous le savons tous, le cadre dans lequel nous évoluons présente souvent une limite : la patience.

En effet, une organisation qui investie dans une démarche d'IA va en général avoir tendance à demander un R.O.I rapide.

Même si certains POC peuvent l'être, il n'en demeure pas moins que construire une démarche pérenne et solide **nécessite une approche méthodique**.

Pour justifier de cette approche, n'oubliez pas de **communiquer** sur vos avancées (échec ou réussites) et donnez de la visibilité à vos interlocuteurs.

J'ai souvent été confronté à ce sujet, le meilleur angle de réponse est de mettre son interlocuteur en face de ses contradictions (en fonction du contexte) .

Par exemple, si la gouvernance de la donnée n'est pas établie dans l'organisation et que la qualité de la donnée est déplorable, dites le de suite car vous n'êtes pas magicien et étant donné que le point de départ est plus éloigné que ce qu'on vous avait dit au départ, l'atteinte à l'objectif sera forcément plus long.

EXPÉRIMENTER, ENCORE ET TOUJOURS

Obtenir un premier résultat probant est en soit une petite victoire. Mais ce n'est souvent que le début du vrai challenge.

Si vous construisez un modèle qui son fonctionne, vous aller devoir le faire fonctionner **mieux**. Soyez constamment au fait des biais de vos modèles pour éviter qu'ils dérivent (drift).

Optimisez également le ratio coût/performance (MLOPS) et demandez-vous toujours si la récurrence des résultats que vous obtenez est adaptée à la capacité de vos interlocuteurs d'agir en conséquence.

En effet, il ne sert à rien de fournir une prédiction toute les heures qui ne permet pas à votre organisation de s'adapter aux conclusions fournies.

Essayez également de **penser contre vous même** en testant des chemins complètement différents de ceux que vous avez déjà emprunté, vous serez peut être agréablement surpris du résultat.

Il est important d'expérimenter et d'apprendre de ses erreurs.

Amélioration des modèles : L'expérimentation permet de tester différentes approches et techniques pour améliorer la performance des modèles. C'est là encore une phase qui peut être considérée comme itérative, la perfection n'étant jamais atteignable, mais on s'en rapproche

Compréhension des données : En expérimentant avec différentes méthodes et des angles d'approche différents, nous pouvons mieux comprendre nos données et découvrir des tendances ou des axes d'améliorations jusqu'alors cachés.

Innovation : L'expérimentation encourage l'innovation. Essayez de répondre à des questions que les autres ne se posent pas encore ou en tenant compte d'axes atypiques permettra potentiellement d'améliorer nos modèles et nos résultats.

Gestion des attentes : Dans le domaine de l'IA et de la science des données, il est important de comprendre que l'erreur est une partie normale du processus. La réalisation de petits objectifs est plus motivant qu'avoir un gros échec qui génèrerait de l'attente.

LE RISQUE DE LA NON QUALITÉ

Comme nous l'avons vu, il n'y a pas de magie dans la mise en place d'une approche IA/DS.

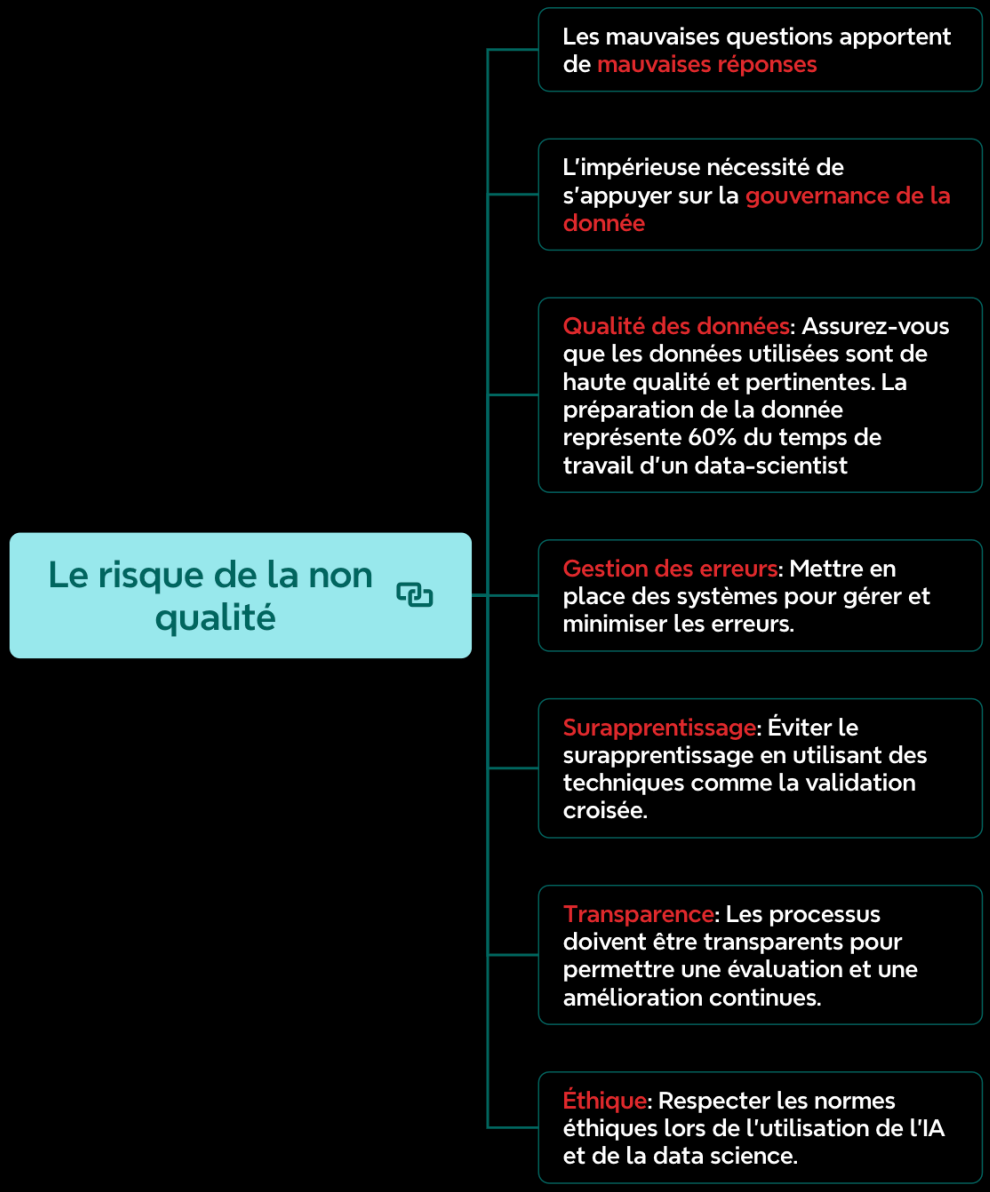
Je l'ai évoqué précédemment, mais le luxe absolu pour mener à bien un projet d'IA, c'est de disposer d'une gouvernance de la donnée établie qui vous permettra de travailler sur un socle compréhensible. Il s'agit là d'un travail de fond.

Qui plus est, si vous disposez déjà d'une vue sur le niveau de qualité des données, vos résultats n'en seront que plus rapides à obtenir.

Mais c'est rarement le cas malheureusement, **car les organisations veulent aborder immédiatement le sujet de la valeur sans travailler le fond.**

De manière générale, dans une stratégie data d'entreprise, l'aspect IA devra donc être abordé une fois ces pré-requis remplis, faute de quoi vous allez essayer de construire un immeuble de 15 étages sur des sables mouvants.

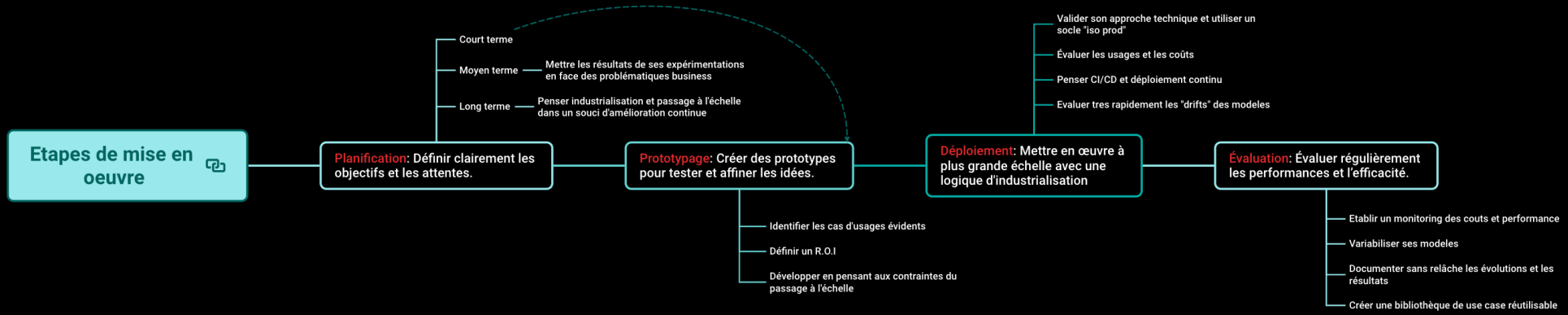
Les entreprises doivent bien considérer ce point : **l'IA et la Data science ne sont en général pas le point d'entrée d'une stratégie data aboutie**, c'est plutôt l'inverse..



LES ÉTAPES DE MISE EN OEUVRE

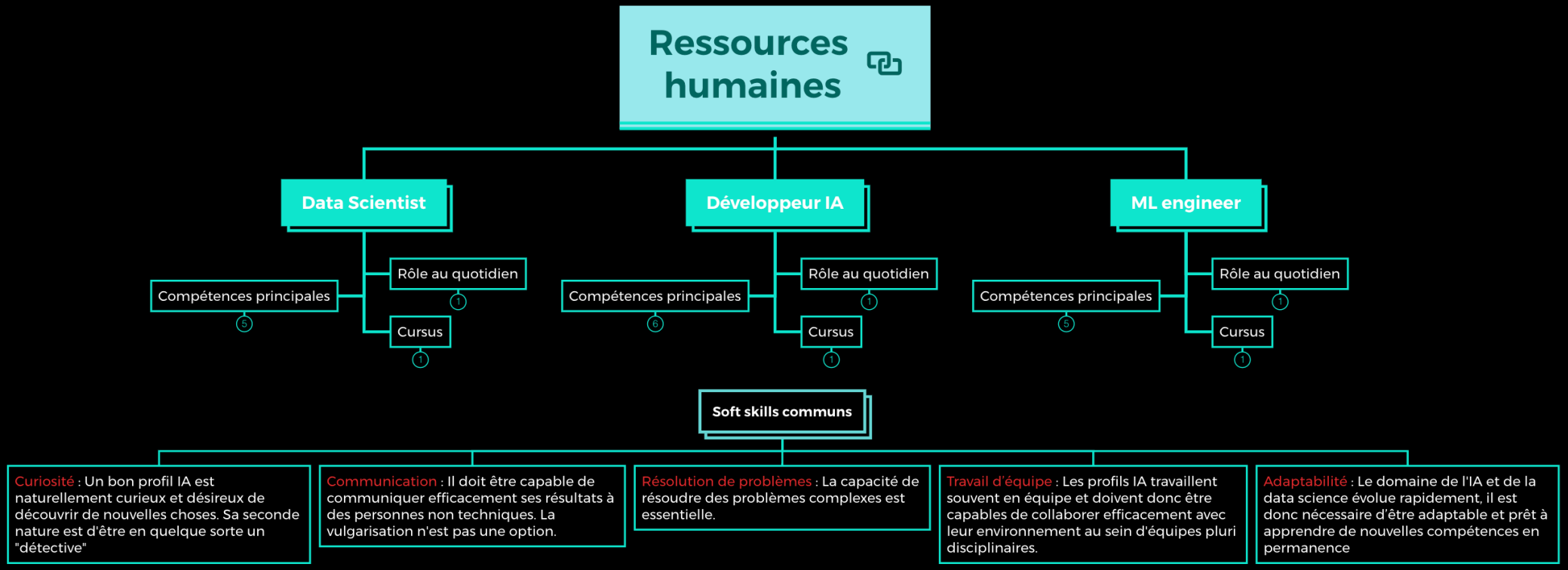
En synthèse, voici donc la démarche a adopter.

(Avec bien entendu les pré-requis mentionnés précédemment que je ne précise pas sur ce schéma)



RESSOURCES HUMAINES
LES MÉTIERS DE L'IA ET DE LA DATA SCIENCE D'AUJOURD'HUI ET DE DEMAIN

Les principaux métiers de l'IA et de la data-science que l'on va trouver principalement sont les 3 suivants : Data scientist , Développeur/ingénieur en IA et ML engineer. Chacun joue un rôle différent dans votre approche IA/ML. (j'exclu volontairement les rôles de management)

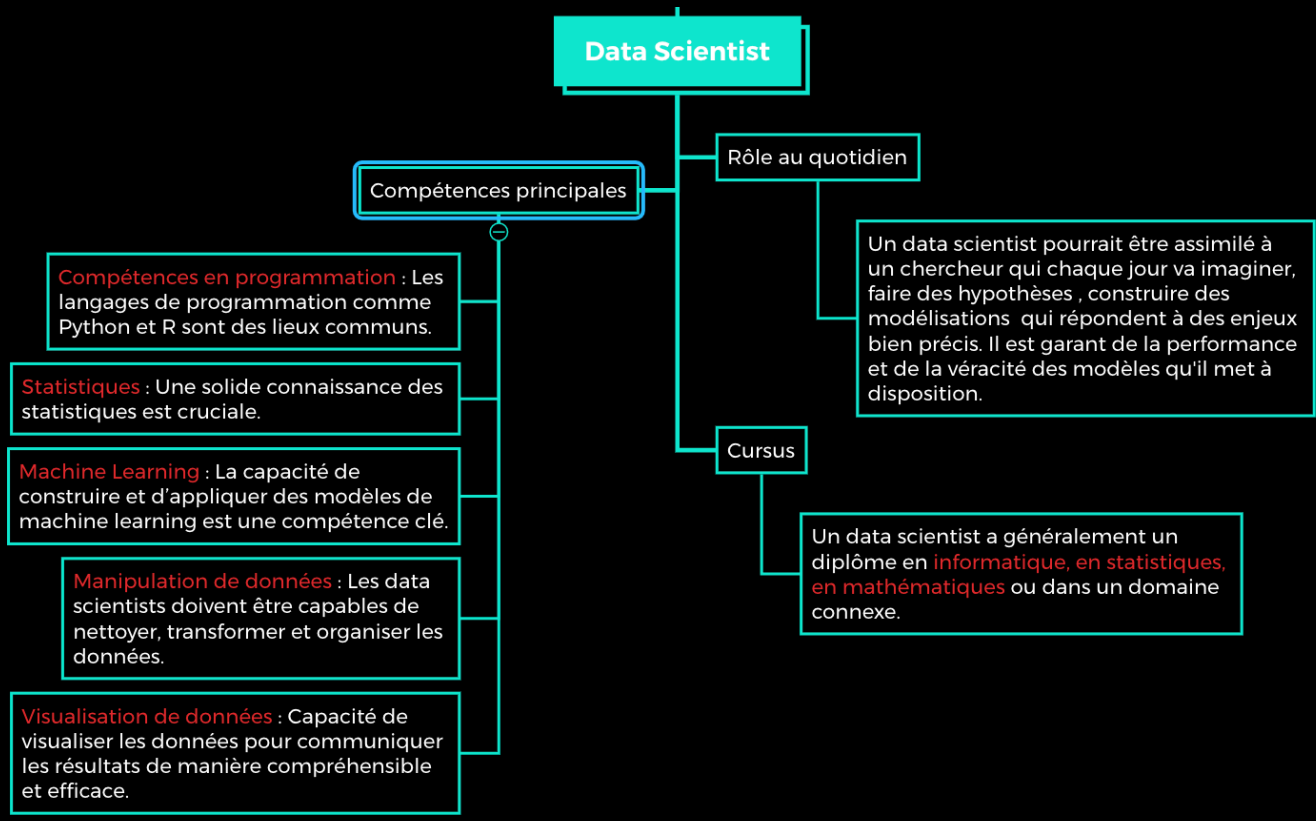


Longtemps dénommé « métier le plus sexy du 21eme siecle », le data scientist joue un rôle prépondérant dans la mise en place d'une stratégie IA au sein des organisation.

Avec un fort background statistique et mathématique ainsi qu'une connaissance aboutie des modèles (voir partie 3 de ce livre), ils sont ceux qui savent les manipuler au mieux pour atteindre des objectifs bien précis, particulièrement des modèles prédictifs (combien de produits vais-je vendre d'ici 3mois ?) ou encore automatiser des moteurs de recommandation basés sur des comportements permettant de s'adapter à des profils spécifiques.

De manière générale, ils sont un ingrédient essentiel de votre équipe data et sont des profils à forte valeur ajoutée. Ne les confondez pas avec des « data analyst » car ils ne jouent pas le même rôle. Un analyst va interpréter des faits, un data scientist va les anticiper.

Pour les attirer et les retenir, il va vous falloir leurs fixer des objectifs bien précis et démontrer que vous comprenez la valeur ajoutée qu'ils peuvent vous apporter.



DÉVELOPPEUR AI / AI ENGINEER

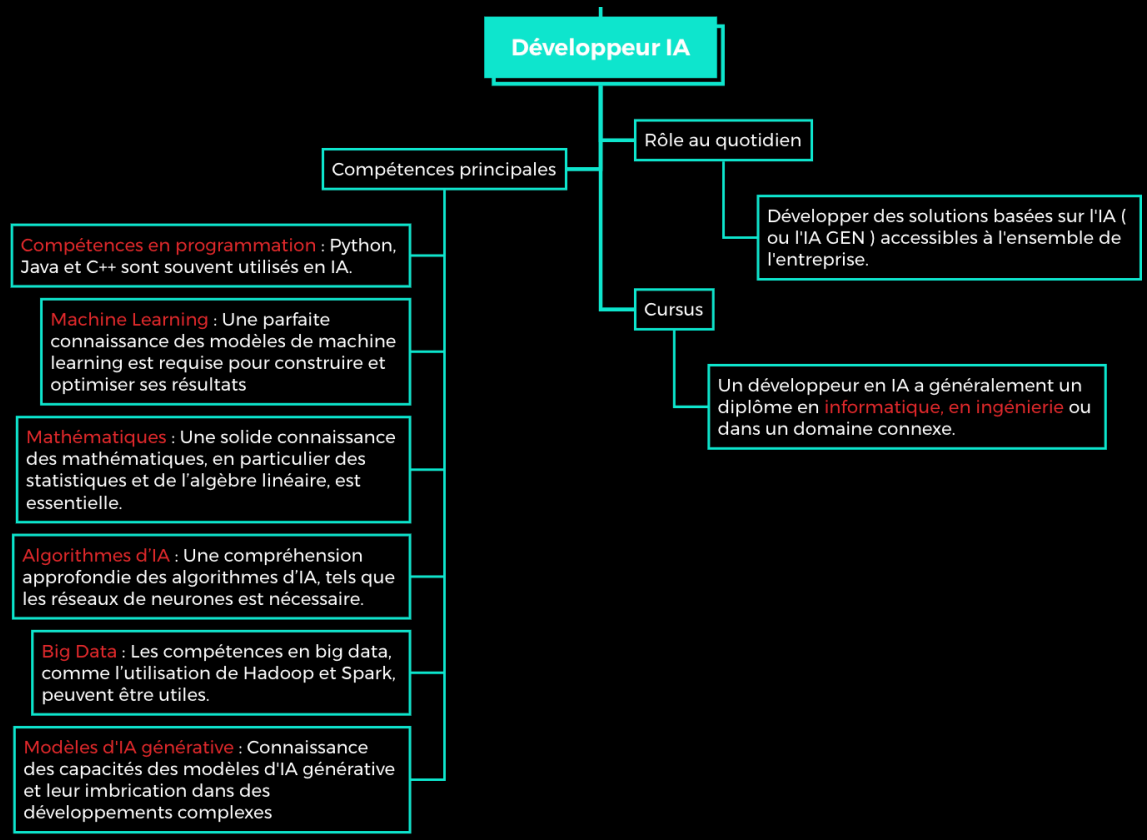
À la différence d'un data scientist, l'AI engineer possède des caractéristiques davantage axées sur la notion de développement informatique.

Il ne va pas spécialement concevoir des modèles de ML (bien qu'il en connaisse les contours) mais plutôt comment apporter des briques fonctionnelles d'IA à votre organisation.

Ainsi, l'IA engineer va savoir considérer l'infrastructure nécessaire au bon déploiement de votre solution d'IA (particulièrement générative), quelles sont les ressources techniques, comment les sécuriser au maximum et comment les optimiser en terme de performance.

Confondre ce rôle avec celui de Data scientist est une erreur car l'angle d'approche n'est pas le même et la finalité différente.

L'AI engineer possède donc un background plutôt technique que mathématique.

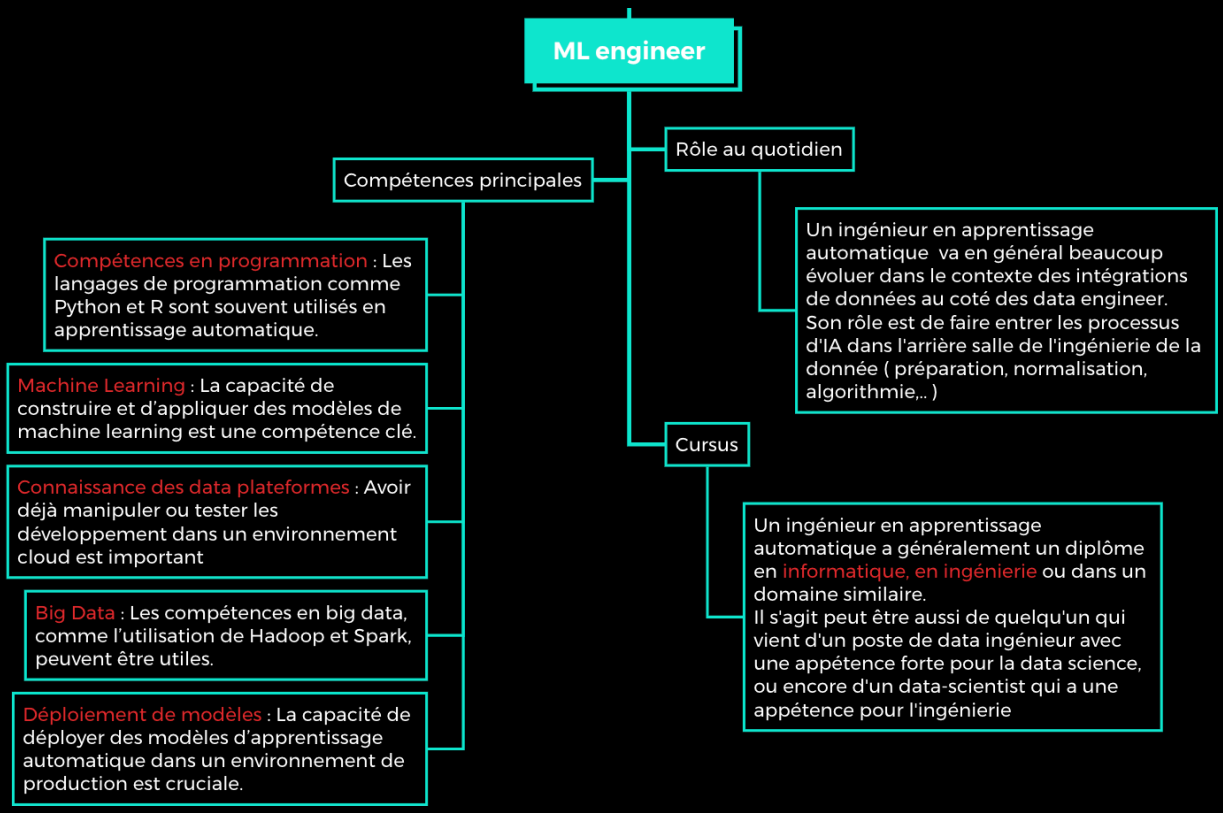


Avec l'évolution des approches et des enjeux liés à l'IA et au ML, le rôle de ML engineer diffère là encore des deux rôles précédent en ce sens qu'il va intervenir plutôt non pas du coté « front » mais plutôt du coté « back » .

Doté de compétences similaires à celles d'un data scientist, il sera en mesure de déployer des modèles qui auront pour but d'optimiser et d'enrichir les flux d'intégration de données dans une data plateforme ou encore de mettre en place des algorithmes complexes permettant d'automatiser certains traitements comme la normalisation ou l'enrichissement de donnée par exemple.

Le ML engineer joue un rôle sous estimé par beaucoup d'organisation en ce sens qu'il créé surtout de la valeur pour l'équipe data elle même et pas directement pour les utilisateurs finaux de l'organisation.

C'est un profil à mi-chemin entre le data scientist et le data engineer.



LES MÉTIERS DE DEMAIN

Avec les usages et les différents retours d'expérience, il est déjà écrit que de nouveaux métiers vont désormais apparaitre. Parmi eux je vous en livre deux qui sont une évidence , ainsi qu'une petite prospective personnelle ce ce que j'entrevois dans les prochaines années :

AI security engineer : La sécurité est déjà un enjeu fort contemporain. Mais comment sécuriser les systèmes et les usages de l'IA dans les organisations. De plus en plus, le rôle d'un ingénieur en sécurité IA va s'imposer de là à devenir indispensable dès prochainement. Il ne s'agit pas simplement de sécuriser un socle technique, mais bien de se prémunir de toute menace portant sur la manière dont les systèmes d'IA agissent . Par exemple, que se passerait-il si demain quelqu'un piratait la voiture autonome dans laquelle vous êtes ?

Juriste spécialisé en IA : Avec les nouveaux usages viendront de nouvelles contraintes normatives (particulièrement en Europe). C'est déjà le cas avec l'AI ACT européen. Le métier de juriste en droit de l'IA va nécessairement émerger dans un contexte ou il conviendra de déterminer les responsabilités liées aux décisions découlants des systèmes d'IA. Ce sera également le cas en entreprise ou l'ont cherchera à déterminer d'où proviennent les responsabilités.

Il est aussi possible de penser à ces autres perspectives :

Intégrateur IA/Objet , NLP engineer, Ethicien de l'IA, Concepteur d'expérience utilisateur pour l'IA, Spécialiste en automatisation des processus robotiques (RPA) … La liste est très longue.

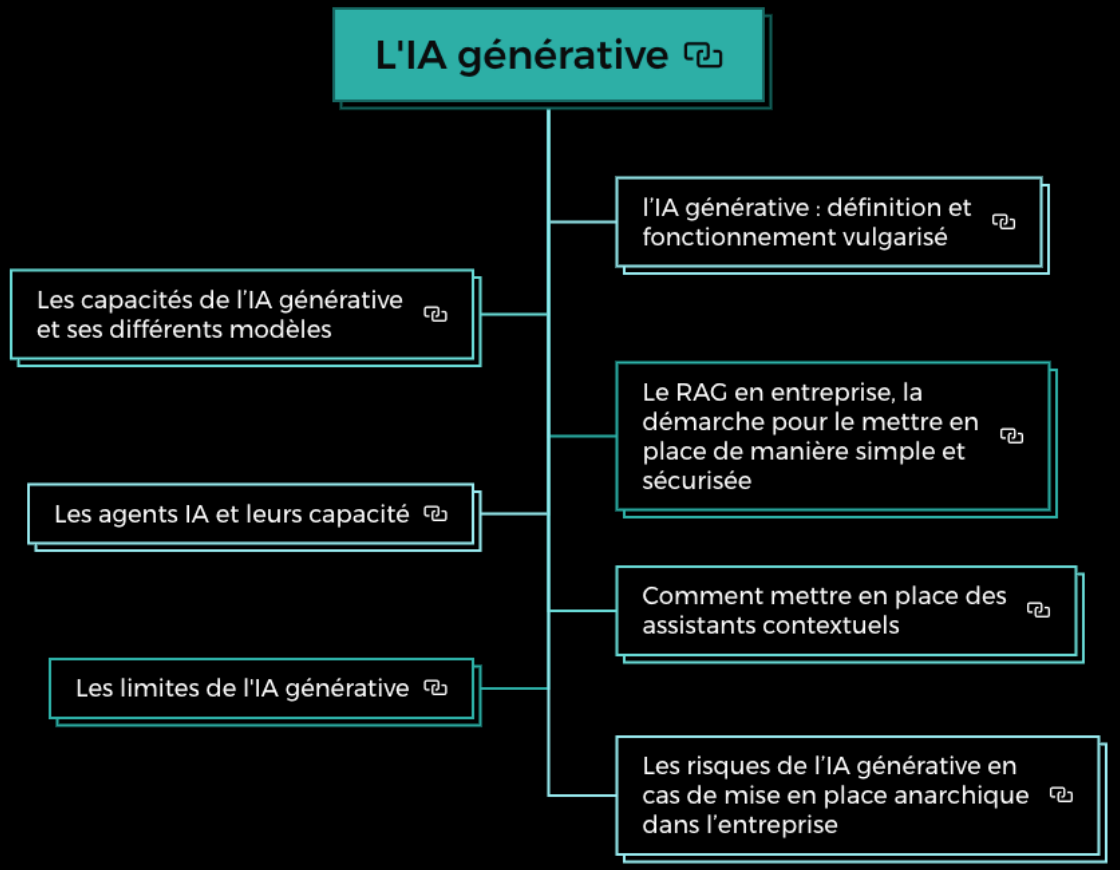
Je n'évoque pas ici le métier souvent cité du « prompt engineer » qui est un mirage absolu. Certes savoir rédiger une instruction optimale pour interagir avec l'IA générative est une compétence utile, de là à lui attribuer le titre d'ingénieur.. c'est un peu exagéré selon moi.

Seconde partie :
L'IA Générative

Outre l'aspect « Waouh » qu'a engendré la vague Chat GPT pour tout un chacun, les organisations s'interrogent désormais sur la manière la plus appropriée de déployer cette technologie en se posant un ensemble de questions pertinentes :

- Comment marche cette technologie ?
- Est ce que c'est sécurisé ?
- Quels sont les gains potentiels, et sont-ils quantifiables ?
- Puis-je interroger mes propres informations facilement avec l'IA générative ?
- Combien ça va me couter ?

Et bien d'autres encore… Je vais essayer de clarifier tout cela de manière pragmatique et avec le recul de ma propre expérience sur ces sujets car **j'ai déployé ceci à l'échelle d'une entreprise de 1400 personnes**.



DÉFINITION ET FONCTIONNEMENT

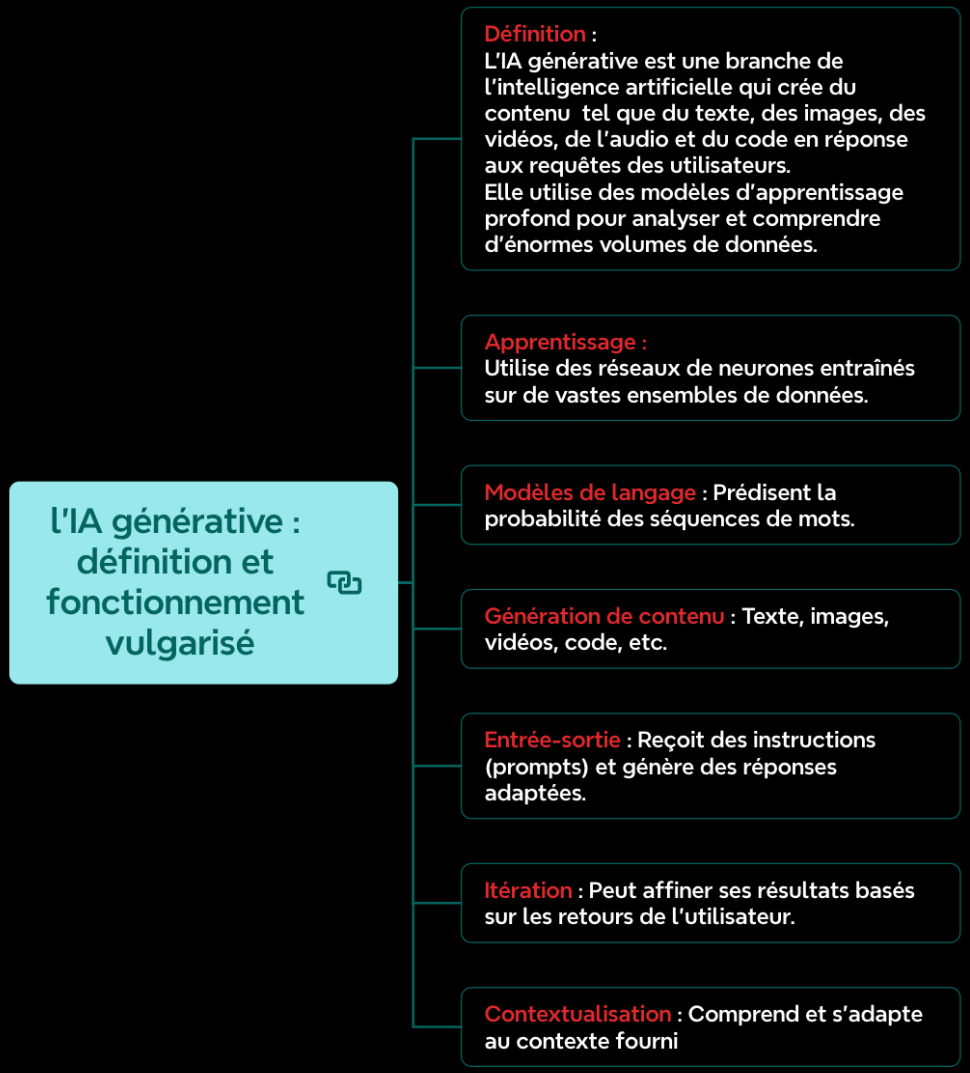
L'IA générative a (et va) boulversé notre quotidien. Qu'il s'agisse de modification des méthodes de travail, de notre approche à l'accès à l'information, cette technologie qui parait magique se doit d'être offerte aux collaborateurs d'une organisation de **manière cadrée, et bien définie**.

Mais de quoi est capable cette technologie, et de quoi sera-t-elle capable demain ? Les cas d'usages sont innombrables et défient dores et déjà l'entendement.

Là où la rupture technologique est totale, c'est surtout dans la manière d'interagir avec la machine et les modèles. Une simple phrase en langage naturel (l'entrée) vous permet désormais d'avoir une réponse à propos dans un contexte définit et pertinent (la sortie) . Ces entrées et sorties peuvent être de différentes natures (voir ci-contre).

Mais si je dois vous expliquer brièvement comment la réponse apportée semble tellement « proche » de ce que vous répondrait un de vos congénères, laissez moi vous expliquer ceci :

Vous disposez d'une suite de dominos numérotés de 1 à 20 avec des couleurs différentes. Vous retirez le n°4 et le n° 13. L'objectif pour l'IA va être de trouver le meilleur n° 4 et n° 13 par rapport aux dominos restant sur la table avec les bonnes couleurs . Probabilité, compréhension du contexte, vitesse de réponse. Rien n'est magique mais vous n'y voyez que du feu.



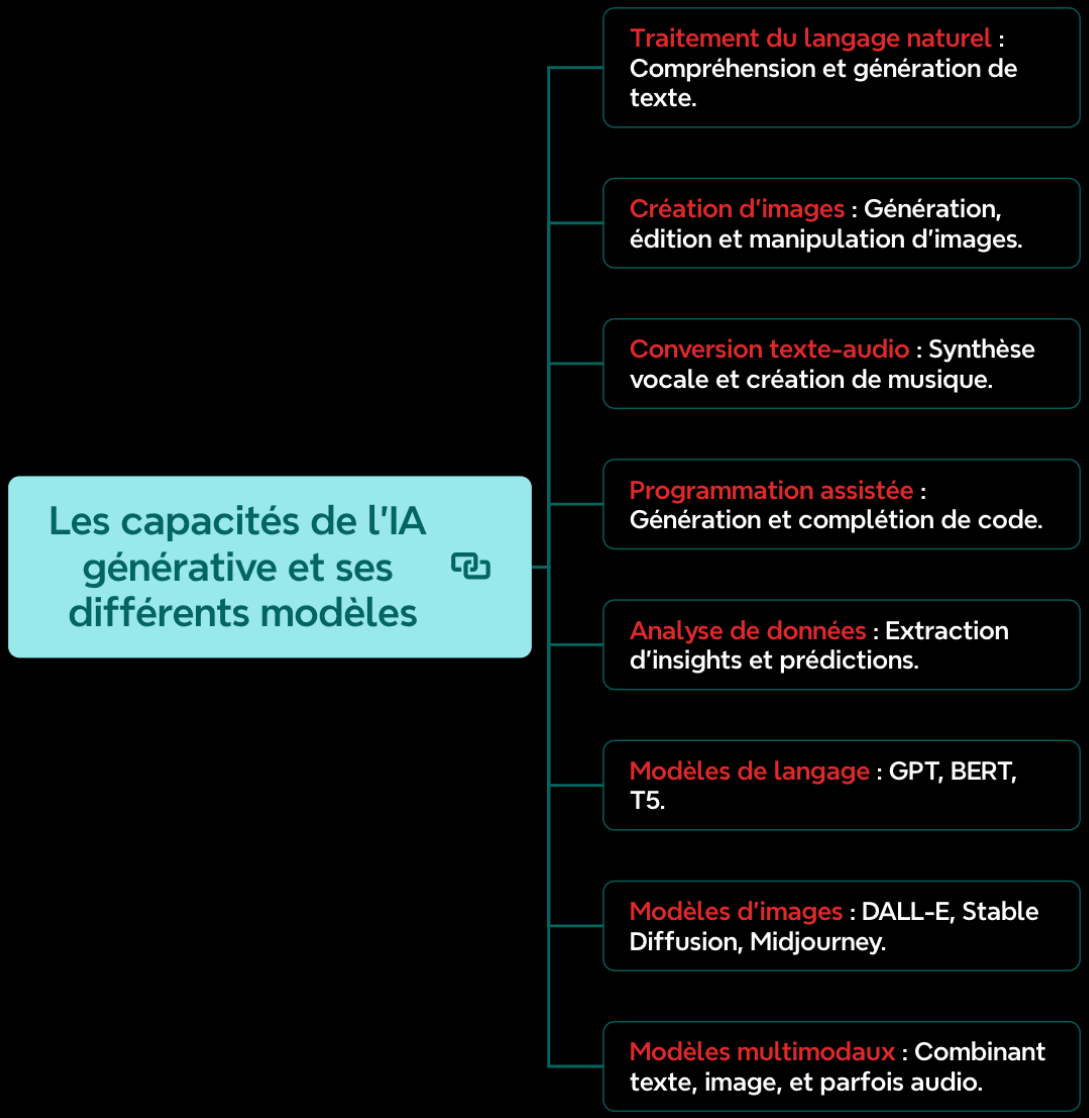
LES CAPACITÉ DE L'IA GÉNÉRATIVE ET SES DIFFÉRENTS MODÈLES

Pour vous apporter la réponse la plus à propos par rapport à une problématique précise, je vais vous indiquer que certains modèles sont plus performant que d'autres.

Mais qu'est ce qu'un modele ? Il s'agit en quelques sorte du « moteur » qui réalisera ce que vous voulez faire. Par exemple pour générer une image à partir d'une phrase (aussi appelé « prompt ») vous devrez utiliser un modele particulier (comme DALL-E ou Midjourney). Par contre pour demander la description d'une image, et donc avoir un texte de description, vous devrez utiliser un autre modele.

Les modèles possèdent des capacités et des versions particulières (par exemple GPT3.5, puis GPT4, puis GPT4o). La différence c'est qu'ils sont « entrainés » su une base de connaissance différente ou enrichie, mais présentent aussi d'autres capacités de traitement des entrées et des sorties.

Ainsi il existe aussi des modèles « multi-modaux » qui sont un peu les couteaux suisse de l'IA GEN. Ils sont capables de traiter plusieurs types d'entrées et d'obtenir différents types de sorties. (comme GPT4o ou Claude Sonnet 3.5 au moment ou j'écris ces lignes)



LE RAG (RETRIEVAL-AUGMENTED-GENERATION) EN ENTREPRISE

On touche ici au graal absolu de l'usage de l'IA GEN. La promesse du RAG est simple : pouvoir poser des questions à ses propres données.

Ça parait logique et simple, mais la complexité de construire un socle RAG n'est pas encore à la portée de toutes les organisations.

Si vous souhaitez aborder ce sujet, je vous invite à commencer par un périmètre simple, et de considérer comme corpus documentaire (les éléments sur lesquels votre IA va se baser pour vous répondre) un échantillons d'éléments de différente nature. Imaginez par exemple poser des questions sur un ensemble de 50 documents contractuels, et demander à l'IA de vous identifier lesquels n'ont pas de clauses de réversibilité.

Par la suite, vous pourrez aller plus loin en constituant une base vectorielle qui vous permettra d'interagir avec un volume plus conséquent d'informations.

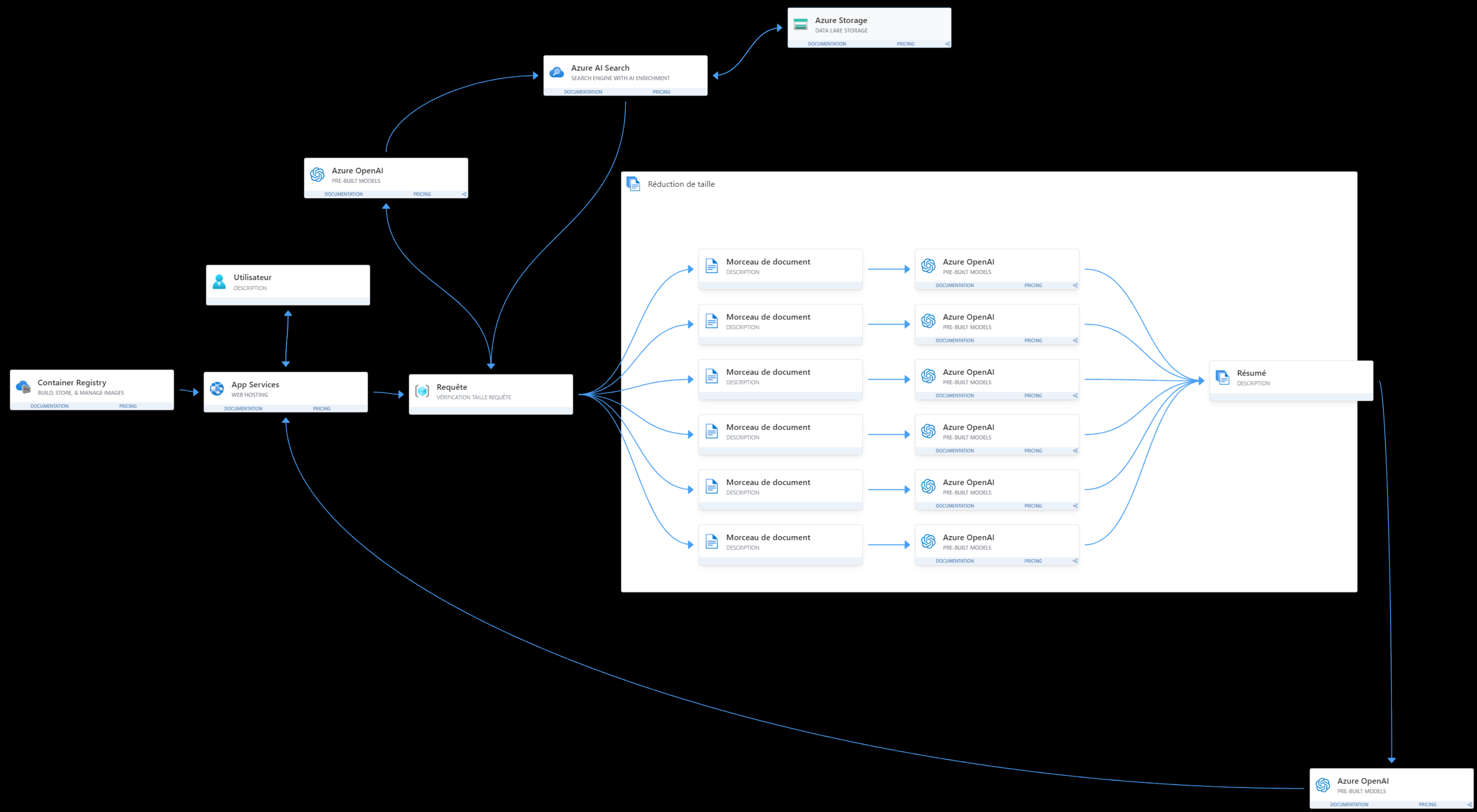
Je vous explique rapidement ce qu'est un « vecteur » :

Dans votre base documentaire se trouve le mot « Apple », est-ce qu'on parle d'un fruit ou du fabricant d'iPhone ? (je ne peux pas faire plus simple :))

Le RAG en entreprise, la démarche pour le mettre en place de manière simple et sécurisée

- **Définition** : Retrieval-Augmented Generation, combinant recherche d'information et génération.
  Le RAG est une technique qui associe la recherche d'entreprise aux grands modèles de langage (LLM) pour améliorer la pertinence des résultats.
  Le processus du RAG peut être divisé en deux grandes étapes
  - **Récupération** : Lorsque le modèle reçoit une requête, il effectue une recherche dans un ensemble prédéfini de documents ou de données pour trouver les informations les plus pertinentes par rapport à la requête.
  - **Génération** : Une fois les informations pertinentes récupérées, le modèle les utilise, en plus de sa propre connaissance interne, pour générer une réponse ou un contenu qui non seulement répond à la requête initiale mais le fait de manière plus informée et précise
- **Le corpus documentaire :** Identification et préparation des sources d'information internes.
- **Indexation** : Création d'une base de connaissances searchable.
  Indexation : L'indexation est un processus qui consiste à analyser le contenu d'un document pour ensuite le reformuler dans une forme plus adaptée à son exploitation dans une application donnée.
  - **Analyse conceptuelle** : Le contenu est analysé et interprété par un documentaliste pour définir les principaux concepts permettant de le caractériser
  - **Reformulation documentaire** : L'analyse conceptuelle permet au documentaliste de reformuler le contenu dans une forme permettant sa manipulation
- **Embbeding** : Mise en place de contrôles d'accès et de chiffrement des données.
  C'est une technique largement utilisée qui consiste à représenter des objets ou des concepts de grande dimension (comme des mots, des phrases, ou même des documents entiers) sous forme de vecteurs de faible dimension.
  Ces **vecteurs** capturent les relations sémantiques et syntaxiques entre les objets, ce qui permet de réaliser des opérations significatives en utilisant des opérations vectorielles simples
- **Gouvernance IA et éthique** : Définir les politiques d'utilisation et de mise à jour.
  Ces conditions d'utilisations doivent mentionner clairement les durées de conservation ainsi que les limites de responsabilités liées à la pertinence des résultats (vous me remercierez plus tard)
- **Sécurité** : Mise en place de contrôles d'accès et de chiffrement des données. (Azure AD ou Entra par ex)
- **Formation** : Ateliers et accompagnement des employés à l'utilisation du système.

EXEMPLE D’UNE INFRASTRUCTURE SIMPLIFIÉE SOUS AZURE POUR OPÉRER DU RAG



LES AGENTS IA ET LEURS CAPACITÉS

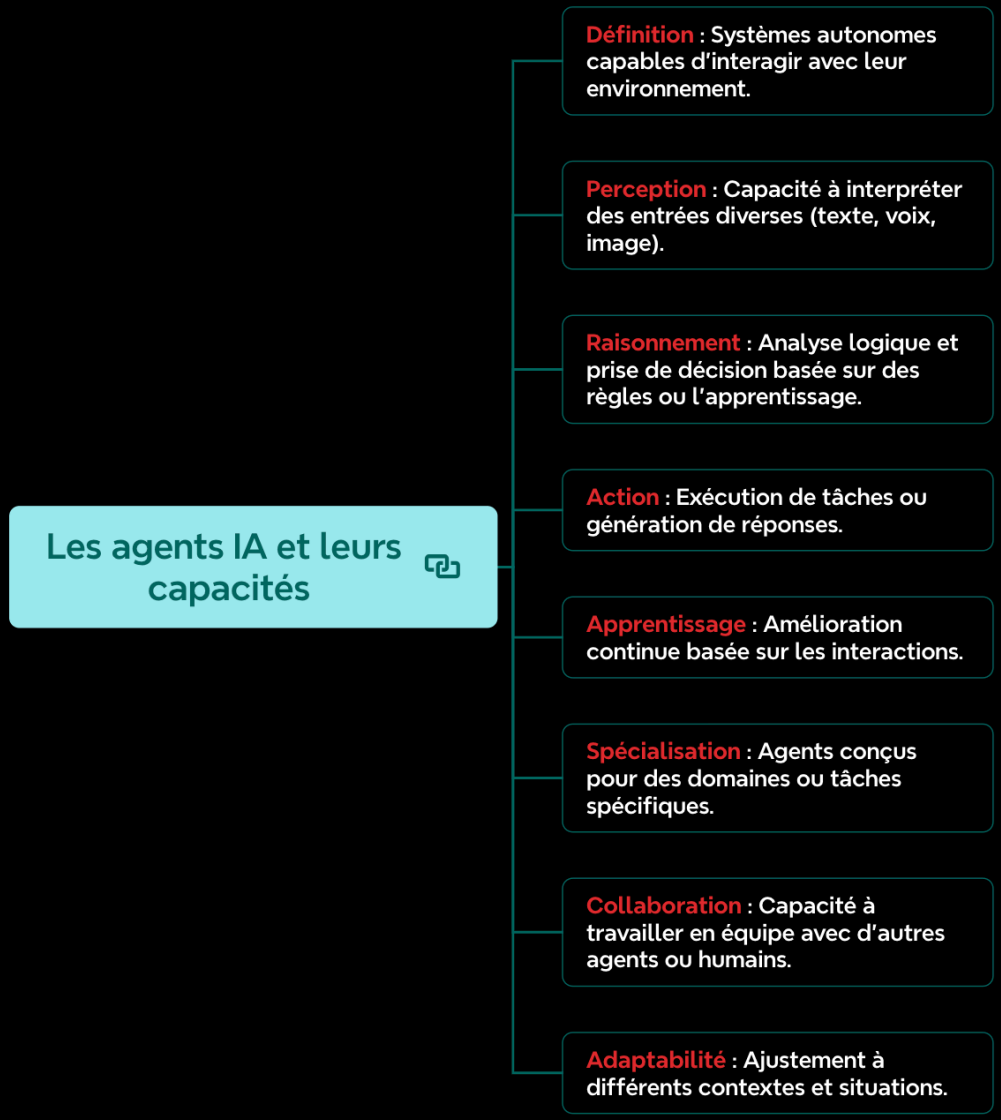
Encore très peu répandus les agents IA poussent encore le curseur un peu plus loin. Pour vous expliquer ce que sont les agents IA, prenons un exemple simple :

Vous voulez développer un logiciel qui scanne et organise vos factures. En temps normal, vous iriez voir votre DSI ou un éditeur. Là, vous demandez simplement la création de ce logiciel à travers une boite de dialogue. Que se passe-t-il ensuite ?

Un premier agent va être en charge de considérer votre demande et la qualifier si nécessaire. Il va ensuite la transmettre à un autre agent qui va la traduire en spécifications techniques. Un 3eme agent vérifie que le travail du 2nd correspond bien à la demande du 1er (vous me suivez ?) . Puis le cahier des charges est transmis à un 4eme agent qui est expert en développement et va générer le code de l'APP. Il livre son code à un 5eme agent qui optimise le code et effectue des contrôles de conformité et de qualité, puis un 6eme agent spécialisé en UX va, sur la base du code, concevoir l'interface.

Vote application fonctionnelle est générée en 2 minutes avec un mode d'emploi du déploiement et valide techniquement. Cet exemple résume simplement le potentiel des agents IA.

Je vous pose une question pour continuer à nourrir votre réflexion : Les agents pourront-ils générer d'autres agents ?



METTRE EN PLACE DES ASSISTANTS CONTEXTUELS

Les assistants contextuels partent d'un principe simple : contextualiser l'IA pour qu'elle ne vous réponde que dans ce contexte précis.

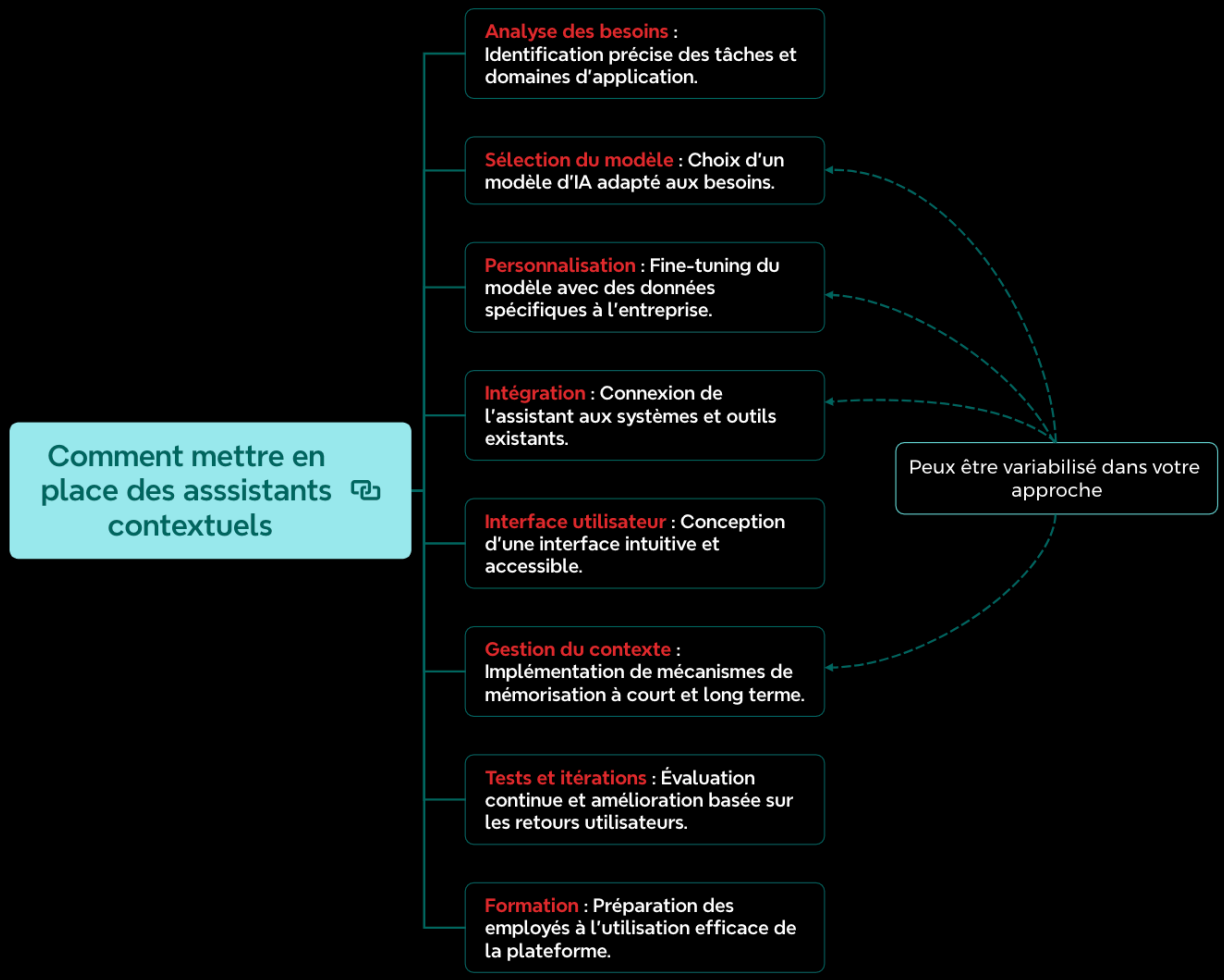
Il ne s'agit pas ici d'agents IA comme nous venons de le voir mais réellement d'une boite à outil très ciblée pour obtenir un résultat très précis sans autre interaction particulière.

Pour créer ce contexte, il va vous falloir le préciser en indiquant ce que votre assistant doit faire, qui il est et de quelle manière il doit répondre.

Cela va en général inclure plusieurs éléments qui sont :

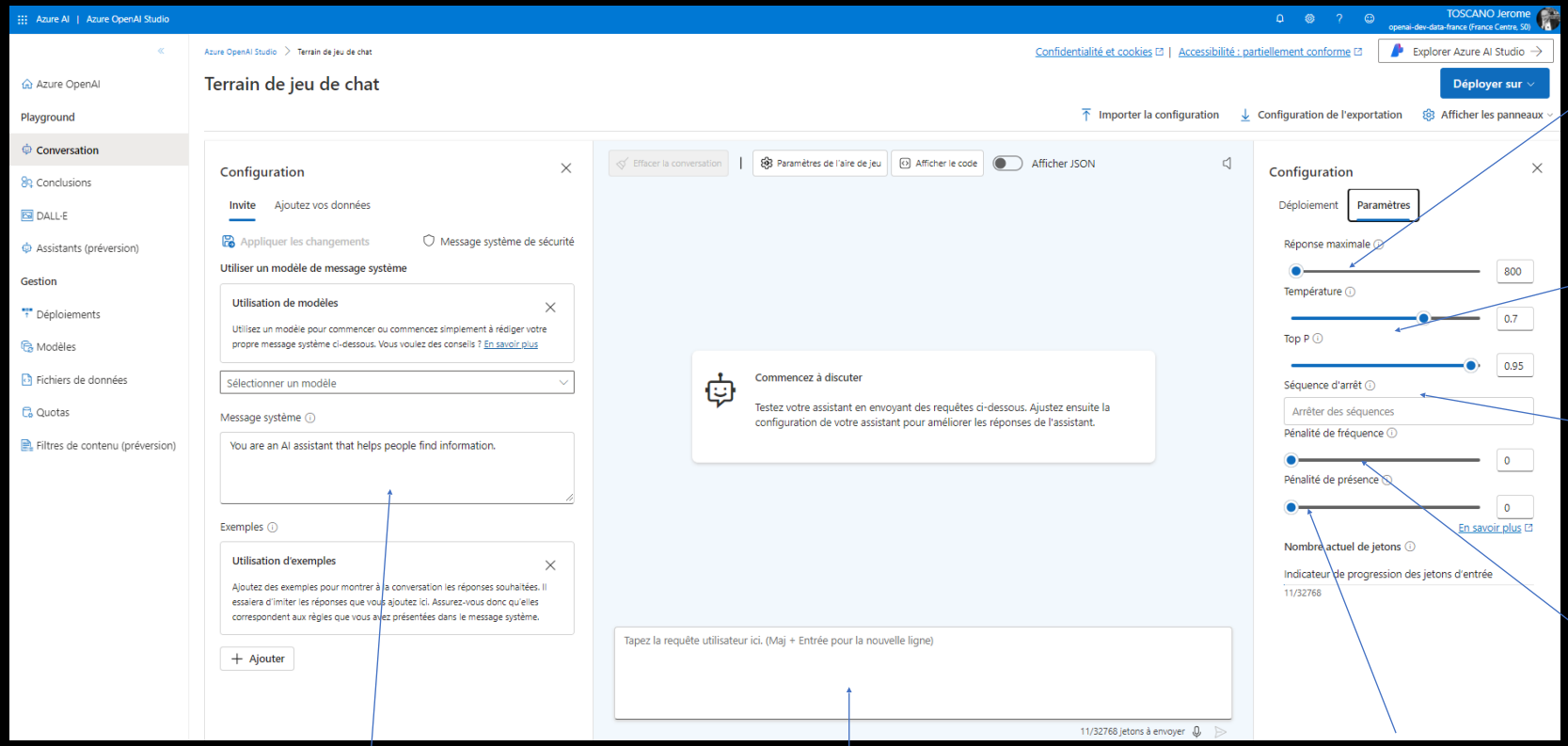
- Le rôle : *tu es un expert en droit des affaires*
- Le contexte : *en France en 2024*
- La précision : *tu te basera sur le code du commerce*
- La tâche à réaliser : *tu répondras toujours dans ce contexte et uniquement ce contexte*.
- Comment la réaliser : *lors de ta réponse, tu indiquera systématiquement le numéro de l'article de loi auquel tu fais référence*

Assemblez les éléments, vous avez votre instruction.



Comment régler un modele d'IA GEN ? Les notion à connaitre

Exemple à travers l'interface d'Azure AI studio :



*Un **jeton** correspond à environ 4 caractères pour un texte anglais typique.*

*L'abaissement de la **température** signifie que le modèle produira des réponses plus répétitives et déterministes. L'augmentation de la température entraînera des réponses plus inattendues ou créatives*

*Abaisser **Top P** réduira la sélection de jetons du modèle à des jetons plus probables. L'augmentation de Top P permettra au modèle de choisir parmi des jetons à probabilité élevée et faible **(soit Top P, soit Temp)***

Réduit les chances de répéter un jeton proportionnellement à la fréquence à laquelle il est apparu dans le texte jusqu'à présent. Cela diminue la probabilité de répéter exactement le même texte dans une réponse

Réduit les chances de répéter tout jeton qui est apparu dans le texte jusqu'à présent. Cela augmente la probabilité d'introduire de nouveaux sujets dans une réponse.

Rédaction du prompt qui fournira le contexte à l'assistant

Zone de « jeu » (test)

LES LIMITES DE L'IA GÉNÉRATIVE

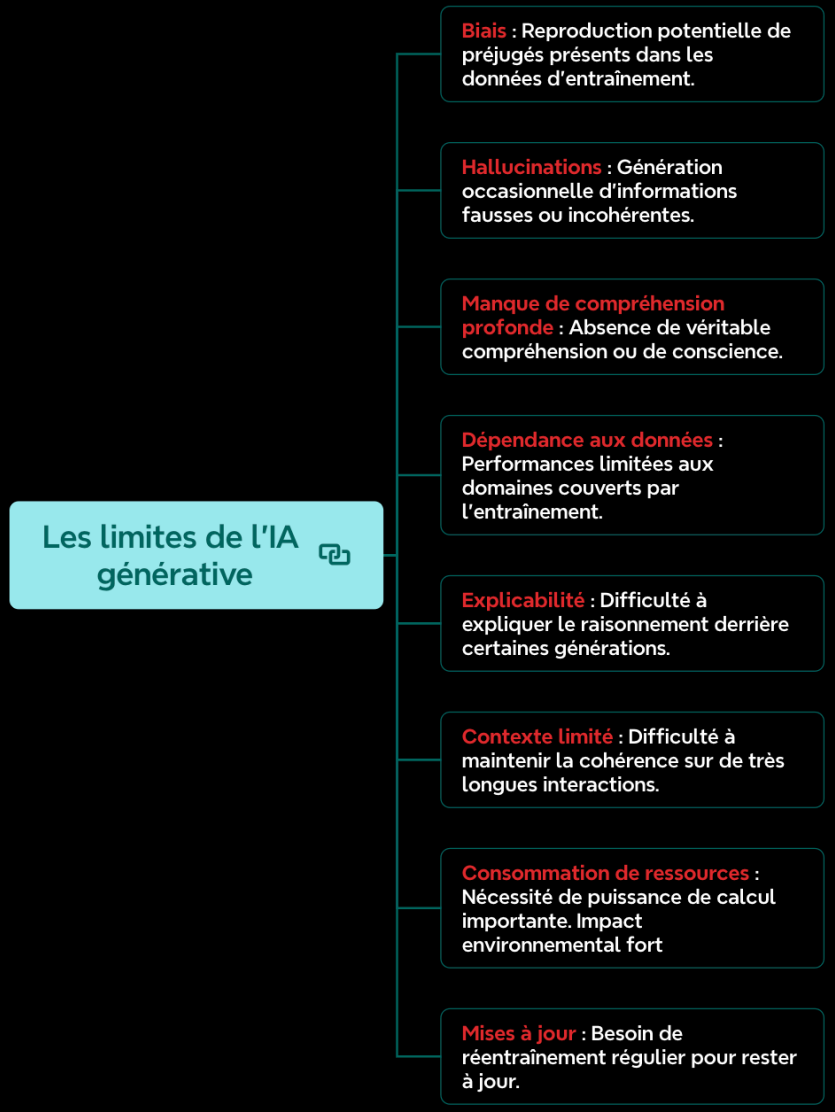
C'est un peu paradoxal, mais la première limite de l'IA générative est tout simplement son coté « boite noire ». En général on ne comprend pas trop sur la base de quels éléments les réponses sont générées(sauf dans le cadre du RAG), voir pire, on ne cherche pas à le comprendre.

Les LLM sont entrainés sur des bases qui sont l**imitées dans le temps**. si par exemple vous interrogez chat GPT sur ce qu'il s'est passé la veille, il y a peu de chance que l'IA vous réponde convenablement. Cela est lié à la dépendance aux données que j'évoque ci-contre.

Une des limites évidente aujourd'hui (je ne sais pas ce qu'il en sera demain) est également **la non gestion des biais**. C'est à dire un axe de décision qui commence à pencher un peu trop dans un sens en commettant des erreurs. Le biais peut également être originel : par exemple les modèles peuvent être entrainés initialement sur un ensemble d'information se limitant à un contexte particulier.

Une limite forte également est le fait que l'usage des LLM et donc de l'IA générative est **extrêmement consommateur d'énergie** et de ressources. Est-ce toujours bien utile ? Pas forcement. On parle de plus en plus de la notion d' « IA frugale » , des petits modèles pour des tâches bien précises et bien moins couteux. (ahhh la sobriété !)

Une dernière limite que je vais évoquer ici est celle de **notre propre imagination**. C'est en même temps ce qui fait que ce sujet est passionnant et un peu vertigineux . Faites les choses par étape.



LES RISQUES D'UN MAUVAIS DÉPLOIEMENT DE L'A GÉNÉRATIVE DANS L'ENTREPRISE

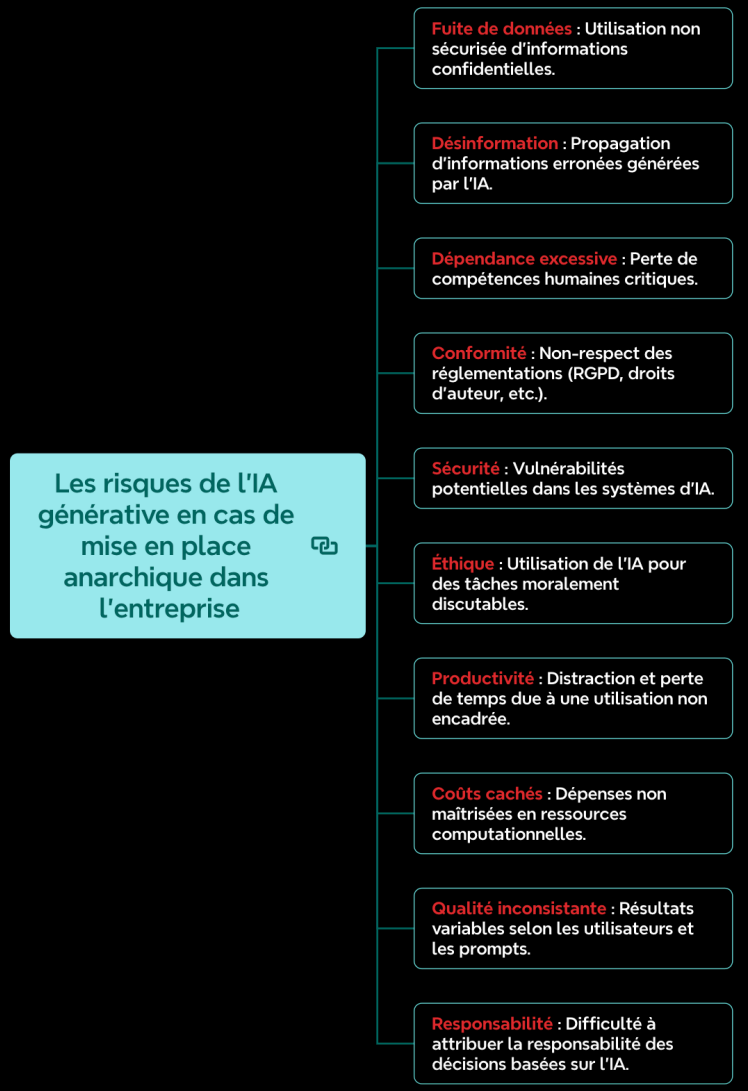
Penser que déployer un tel outil au sein de votre organisation sans vous prémunir des risques qui lui sont liés est une totale hérésie.

Si les collaborateurs ne sont pas au fait des limites et des bonnes pratiques liées à un l'usage de l'IA générative, vous allez rencontrer de grandes déconvenues.

Il convient donc, comme toujours, de se prémunir des risques potentiels, tant en termes d'usages que de conséquences.

Formez vos utilisateurs, expliquer le fonctionnement, les limites et le cadre d'utilisation. Ne les laissez pas seuls en face d'un outil dont les éléments générés pourraient être pris comme argent comptant.

D'autre part, monitorer l'ensemble de l'activité et placez des gardes fous sur les termes sensibles ou inappropriés, tant en termes d'usage que de gestion des accès. Vous devez maitriser ce qui se passe, et comprendre ce que les utilisateurs font. Il en va de la valeur que vous apporterez à terme à vos utilisateurs et donc, à votre organisation.



Troisième partie :
Description et fonctionnement des principaux algorithmes de Machine Learning

Les modèles ont des finalités différentes et répondent donc, à des objectifs différents. Il est possible de les utiliser de manière isolé pour répondre à une problématique en particulier, mais ils peuvent également être combinés pour obtenir des résultats plus probants.

Je vous ai ici listé les principaux algorithmes utilisés communément, et en complément pour chacun d'entre eux, je me suis posé les questions suivantes :

- Comment fonctionne-t-il ?
- Quels sont ses avantages ?
- Quels sont ses inconvénients ?
- Quels sont les principaux cas d'utilisation ?
- Comment est-il construit ?
- Comment s'utilise-t-il dans un contexte Big Data ?
- Quels sont ses limites ?
- Quel est son rôle dans l'IA ?

En complément, je vous décris également comment évaluer la bonne performance d'un modèle à travers des notions communes.

LES 3 FAMILLES DE MODÈLE

Je vais ici décrire ce qu'est un modele supervisé, non supervisé et semi-supervisé. Une petite analogie s'impose :

1. **Modèles supervisés** : Imagine que tu apprends à dessiner des animaux avec ton professeur. Ton professeur te montre comment dessiner un chat, un chien, un oiseau, etc., et te dit exactement ce que chaque dessin représente. Plus tard, quand tu vois un animal que tu n'as jamais dessiné auparavant, tu peux utiliser ce que tu as appris pour deviner quel animal c'est. C'est ce qu'on appelle l'apprentissage supervisé - tu as des exemples étiquetés (les dessins d'animaux avec leurs noms) pour t'aider à apprendre.

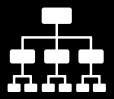
2. **Modèles semi-supervisés** : Maintenant, imagine que ton professeur te montre comment dessiner certains animaux, mais pas tous. Par exemple, il te montre comment dessiner un chat et un chien, mais pas un oiseau. Plus tard, quand tu vois un oiseau, tu dois deviner ce que c'est en te basant sur ce que tu as appris et sur ce que tu observes. C'est ce qu'on appelle l'apprentissage semi-supervisé - tu as quelques exemples étiquetés (les dessins de chats et de chiens avec leurs noms) et tu utilises ces informations pour faire des suppositions éclairées sur les exemples non étiquetés (le dessin de l'oiseau).

3. **Modèles non supervisés** : Enfin, imagine que ton professeur te donne une pile de dessins d'animaux sans te dire ce qu'ils sont. Il te demande de les trier en groupes qui se ressemblent. Tu pourrais finir par mettre tous les dessins de chats dans un groupe, tous les dessins de chiens dans un autre groupe, etc., même si personne ne t'a dit quels dessins représentaient quels animaux. C'est ce qu'on appelle l'apprentissage non supervisé - tu n'as pas d'exemples étiquetés pour t'aider, tu dois donc trouver des modèles et faire des suppositions par toi-même.

De manière générale, pour chaque modèle, je commencerais toujours par vous donner une petite analogie simple qui vulgarisera son fonctionnement et son utilité comme si vous aviez 5 ans ;)

LES MODÈLES SUPERVISÉS

Les Forets aléatoires

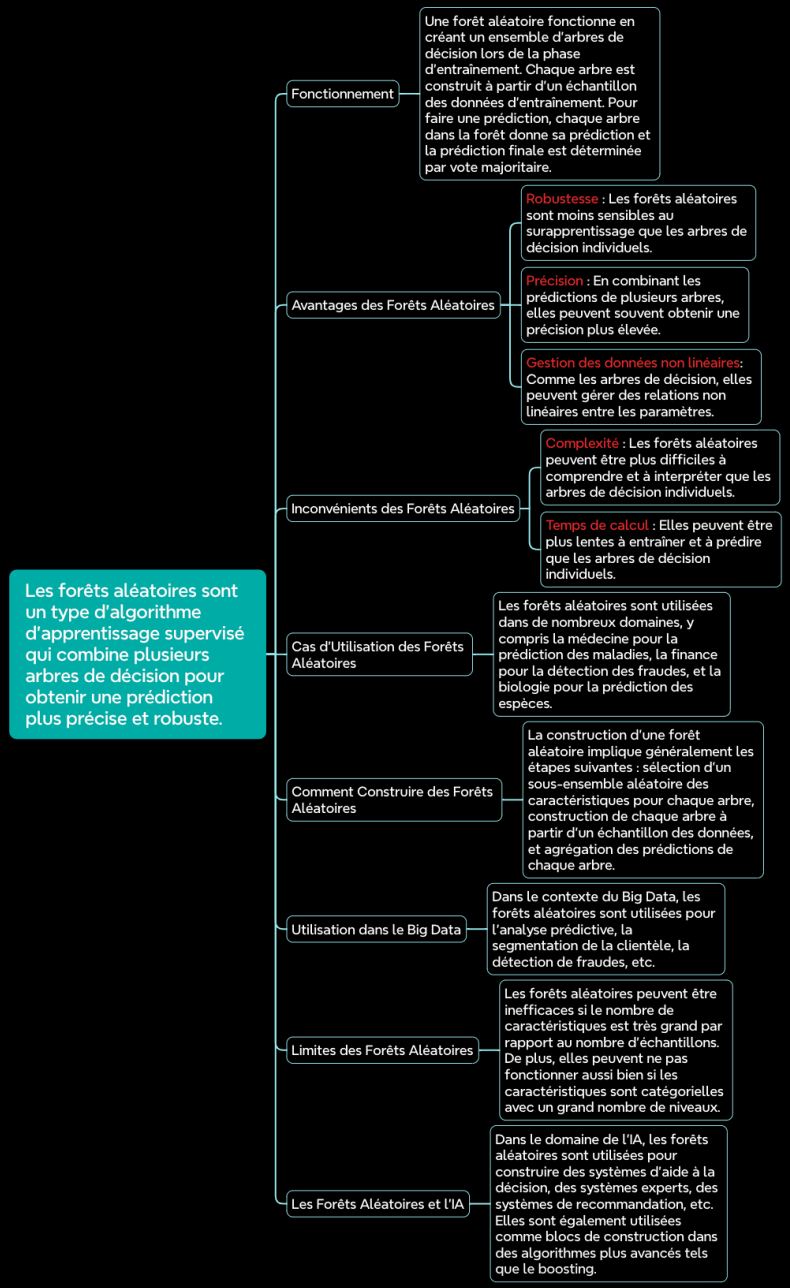


Imagine que tu joues à un jeu de devinettes avec tes amis. Chaque ami te pose une question pour essayer de deviner ce que tu penses. Par exemple, ils pourraient te demander : “Est-ce que c’est un animal ?” ou “Est-ce que ça se mange ?”. Chaque ami fait sa propre supposition sur ce que tu penses en fonction de tes réponses à leurs questions.

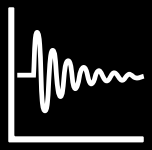
Maintenant, imagine que tu combines toutes les suppositions de tes amis pour faire une supposition finale. C’est un peu comme ça que fonctionne l’algorithme des forêts aléatoires.

Dans une forêt aléatoire, chaque “ami” est un arbre de décision - un modèle qui fait des suppositions en posant une série de questions. Chaque arbre de décision dans la “forêt” fait sa propre supposition. Ensuite, l’algorithme des forêts aléatoires combine toutes ces suppositions pour faire une supposition finale.

C’est pourquoi on appelle cela une “forêt” aléatoire - c’est comme si tu avais une forêt d’arbres de décision qui travaillent ensemble pour faire une supposition!



La régression logistique



Imagine que tu joues à un jeu où tu dois deviner si un fruit est une pomme ou une orange en fonction de sa couleur et de sa taille. Tu pourrais dire que si le fruit est rouge et petit, alors c'est probablement une pomme. Si le fruit est orange et gros, alors c'est probablement une orange.

La régression logistique fonctionne de manière similaire. Elle prend différentes caractéristiques (comme la couleur et la taille du fruit) et utilise ces informations pour prédire à quelle catégorie appartient quelque chose (comme si le fruit est une pomme ou une orange).

La différence est que la régression logistique ne se limite pas à deux options (pomme ou orange). Elle peut prédire la probabilité d'appartenance à plusieurs catégories différentes.

Par exemple, elle pourrait prédire la probabilité qu'un fruit soit une pomme, une orange, une banane, etc.

La régression logistique est une méthode d'apprentissage supervisé utilisée pour résoudre des problèmes de classification. Elle est particulièrement utile lorsque la variable cible est binaire.

- Fonctionnement
  - La régression logistique utilise la fonction logistique pour transformer sa sortie en une probabilité qui peut être utilisée pour prédire une classe. Plus précisément, elle modélise la probabilité que la variable cible appartienne à une classe particulière.
- Avantages de la Régression Logistique
  - Facilité d'interprétation : Les coefficients de la régression logistique peuvent être interprétés en termes de chances (odds).
  - Robustesse: Elle est moins sensible aux valeurs aberrantes et à la multicollinéarité que la régression linéaire.
  - Efficacité : Elle est rapide à calculer et ne nécessite pas de ressources informatiques importantes.
- Inconvénients de la Régression Logistique
  - Linéarité : Elle suppose une relation linéaire entre les caractéristiques et le logarithme des chances.
  - Problèmes de séparation : Si une caractéristique peut parfaitement prédire la classe cible, la régression logistique peut avoir du mal à estimer les coefficients.
- Cas d'Utilisation de la Régression Logistique
  - La régression logistique est largement utilisée dans divers domaines tels que la médecine, l'économie, le marketing, etc. Par exemple, elle peut être utilisée pour prédire si un patient a une maladie en fonction de ses symptômes, ou si un client va acheter un produit en fonction de son historique d'achats.
- Comment Construire la Régression Logistique
  - La construction d'un modèle de régression logistique implique généralement plusieurs étapes : la sélection des caractéristiques, l'estimation des coefficients à l'aide de la méthode du maximum de vraisemblance, et l'évaluation du modèle à l'aide de techniques comme la validation croisée.
- Utilisation dans le Big Data
  - Dans le contexte du Big Data, la régression logistique peut être utilisée pour l'analyse prédictive, la segmentation de la clientèle, la détection de fraudes, etc. Elle est particulièrement utile lorsque les données sont volumineuses et que la variable cible est binaire.
- Limites de la Régression Logistique
  - La régression logistique peut avoir du mal à traiter des problèmes non linéaires et à gérer des caractéristiques corrélées. De plus, elle ne peut pas être utilisée pour prédire des variables cibles continues.
- La Régression Logistique et l'IA
  - Dans le domaine de l'IA, la régression logistique est souvent utilisée comme bloc de construction dans des algorithmes plus avancés tels que les réseaux de neurones. Elle est également utilisée pour construire des systèmes d'aide à la décision, des systèmes experts, des systèmes de recommandation, etc.

L’arbre de décision

Imagine que tu joues à un jeu de “20 questions”.

Tu penses à quelque chose, et tes amis doivent deviner ce que c’est en te posant des questions auxquelles tu peux répondre par oui ou non. Par exemple, ils pourraient te demander : “Est-ce que c’est un animal ?” ou “Est-ce que ça se mange ?”. En fonction de tes réponses, ils posent des questions de plus en plus précises jusqu’à ce qu’ils puissent deviner ce à quoi tu penses.

Un arbre de décision fonctionne de la même manière. Il pose une série de questions pour diviser les données en groupes de plus en plus petits et de plus en plus spécifiques.

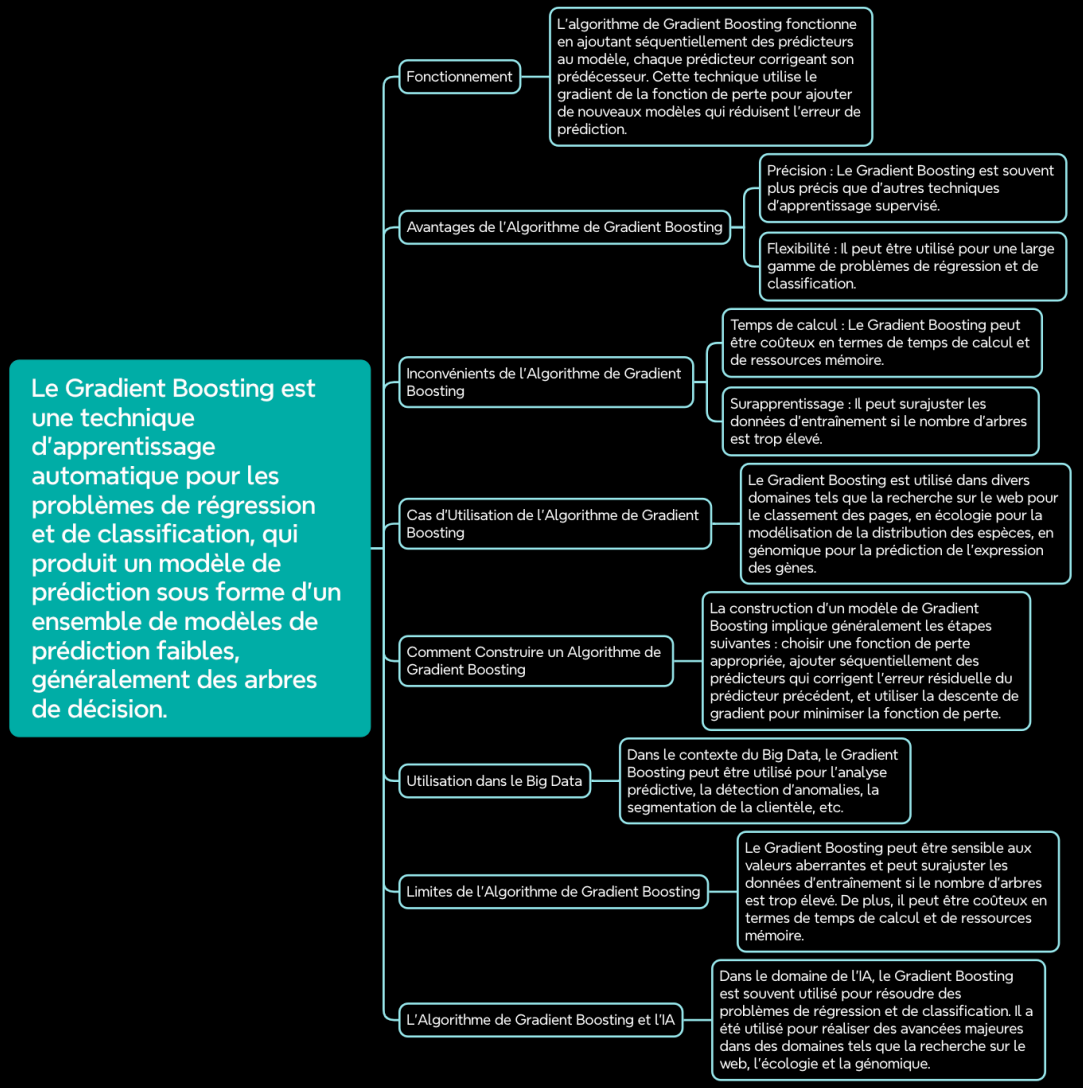
Par exemple, dans un arbre de décision qui essaie de prédire si un fruit est une pomme ou une orange, la première question pourrait être : “Est-ce que le fruit est rouge ?”. Si la réponse est oui, l’arbre pourrait prédire que le fruit est une pomme. Si la réponse est non, l’arbre pourrait poser une autre question, comme “Est-ce que le fruit est rond ?”, et ainsi de suite.

Le grandiant boosting

Tu essaies de résoudre un puzzle complexe, mais tu as du mal à le faire tout seul. Alors, tu demandes de l'aide à tes amis. Le premier ami vient et résout une partie du puzzle. Le deuxième ami vient ensuite et corrige certaines des erreurs faites par le premier ami tout en résolvant une autre partie du puzzle. Le troisième ami fait de même, et ainsi de suite. À la fin, tu combines toutes les pièces résolues par tes amis pour obtenir le puzzle complet. C'est un peu comme ça que fonctionne le gradient boosting.

Dans le gradient boosting, chaque "ami" est un modèle de prédiction simple, appelé un "apprenant faible". Chaque apprenant faible fait des prédictions sur les données, puis les erreurs de ces prédictions sont utilisées pour former le prochain apprenant faible. Ce processus est répété plusieurs fois, et à la fin, toutes les prédictions des apprenants faibles sont combinées pour obtenir la prédiction finale.

C'est pourquoi on appelle cela "boosting" - chaque apprenant faible "booste" les performances du suivant en lui indiquant où se concentrent les erreurs.



LES MODÈLES SEMI-SUPERVISÉS

L’apprentissage multi-instance

Nous avons un sac plein de bonbons. Certains bonbons sont sucrés et d’autres sont amers. Mais il y a un hic - tu ne peux pas goûter les bonbons individuellement. Au lieu de cela, tu dois déterminer si un sac entier de bonbons est principalement sucré ou amer.

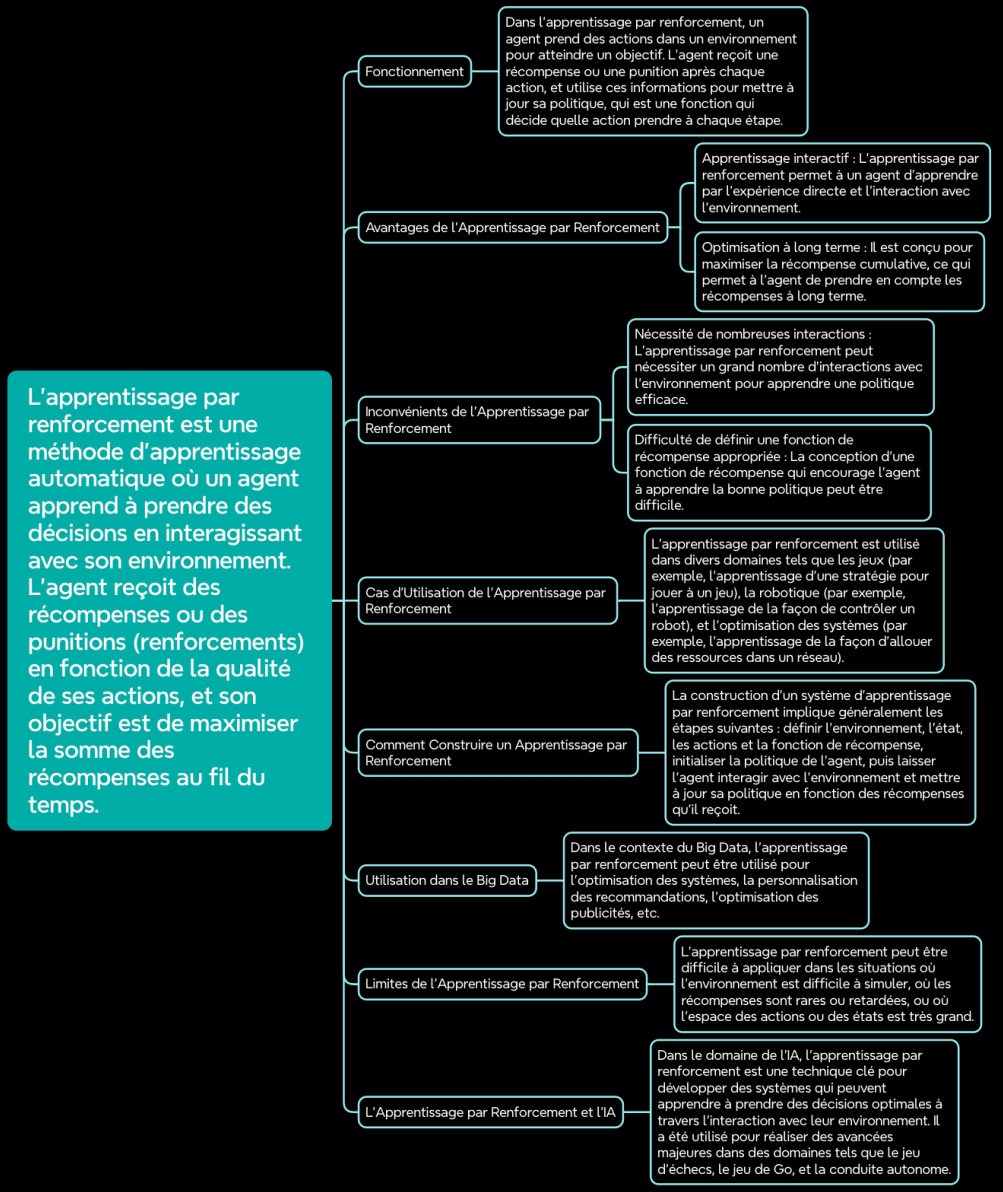
C’est un peu comme ça que fonctionne l’apprentissage multi-instance. Dans l’apprentissage multi-instance, tu as des “sacs” de données, et chaque sac contient plusieurs “instances” (comme les bonbons dans le sac). Mais tu ne connais pas les étiquettes des instances individuelles (tu ne sais pas quels bonbons sont sucrés ou amers). Au lieu de cela, tu connais seulement l’étiquette du sac entier (si le sac est principalement sucré ou amer).

Ton objectif est d’apprendre à prédire l’étiquette d’un nouveau sac de données, même si tu ne connais pas les étiquettes des instances individuelles.

L’apprentissage par renforcement

Tu joues à un jeu vidéo pour la première fois. Au début, tu ne sais pas comment jouer, alors tu essaies différentes choses pour voir ce qui fonctionne. Parfois, tu gagnes des points, ce qui te donne envie de répéter ce que tu as fait. D’autres fois, tu perds des points, alors tu essaies d’éviter de refaire la même chose. Avec le temps, tu apprends à jouer au jeu de mieux en mieux en renforçant les actions qui te donnent plus de points et en évitant celles qui te font perdre des points.

C’est un peu comme ça que fonctionne l’apprentissage par renforcement. Dans l’apprentissage par renforcement, un agent (comme toi dans le jeu vidéo) apprend à effectuer des actions dans un environnement (comme le jeu vidéo) pour maximiser une certaine récompense (comme les points dans le jeu). L’agent ne sait pas au départ quelles actions mèneront à la plus grande récompense, il doit donc explorer l’environnement et apprendre de ses erreurs et de ses succès. En quelque sorte le modele qui imite l’expérience :)

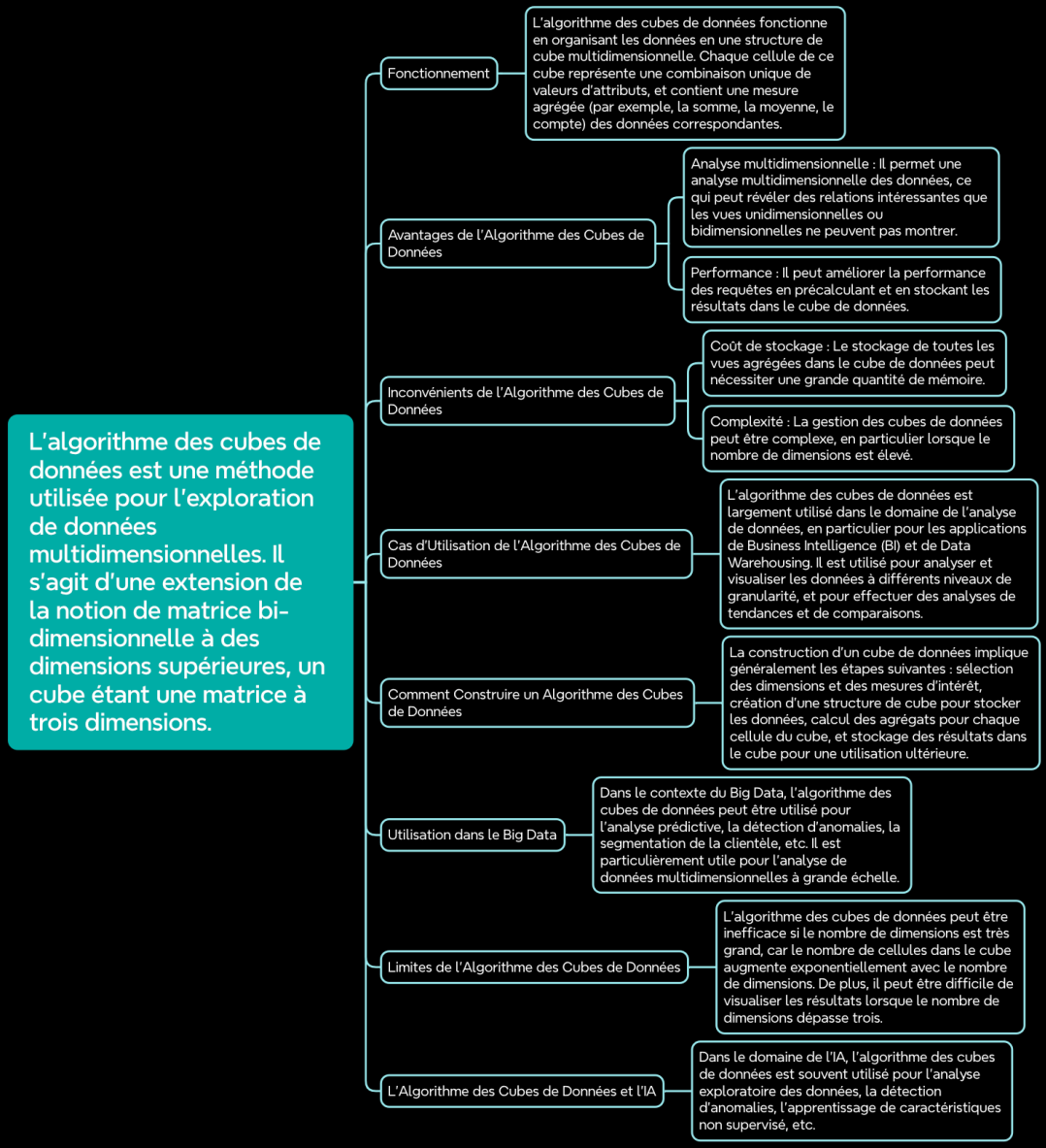


L’algorithme des cubes de données

Dans un grand bac à sable se trouvent plein de jouets différents. Tu décides de trier tous les jouets par type, couleur et taille. Tu commences par faire un tas pour chaque type de jouet (par exemple, les voitures, les poupées, les blocs). Ensuite, pour chaque tas, tu fais des sous-tas basés sur la couleur (par exemple, les voitures rouges, les voitures bleues, etc.). Enfin, pour chaque sous-tas, tu fais des piles plus petites basées sur la taille (par exemple, les petites voitures rouges, les grandes voitures rouges, etc.).

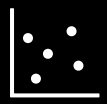
C’est ainsi que fonctionne l’algorithme des cubes de données.

Dans un cube de données, tu as des données sur plusieurs dimensions (comme le type, la couleur et la taille des jouets). L’algorithme des cubes de données te permet de résumer et d’organiser ces données de manière à ce que tu puisses facilement les analyser sous différents angles (par exemple, combien de petites voitures rouges tu as, combien de grandes poupées bleues tu as, etc.).



LES MODÈLES NON SUPERVISÉS

L’algorithme de clustering K-Means



Tu as un de nouveau un grand sac de bonbons de différentes couleurs : rouge, bleu, vert et jaune. Tu veux séparer les bonbons par couleur, mais tous les bonbons sont mélangés et tu ne peux pas les voir tous en même temps. Alors, comment fais-tu ?

Tu pourrais commencer par prendre un bonbon au hasard et dire : “C’est un bonbon rouge”. Ensuite, tu prends un autre bonbon. Si ce bonbon est proche de la couleur du bonbon rouge, tu le mets dans le même tas. Sinon, tu commences un nouveau tas. Tu répètes ce processus jusqu’à ce que tous les bonbons soient triés en tas de couleurs similaires.

C’est un peu comme ça que fonctionne l’algorithme de clustering K-means. Dans le clustering K-means, tu commences par choisir un certain nombre de “centres” au hasard (comme les premiers bonbons que tu as choisis). Ensuite, pour chaque point de données, tu le mets dans le groupe dont le centre est le plus proche. Tu répètes ce processus jusqu’à ce que les centres ne bougent plus beaucoup (ce qui signifie que les points de données ne changent plus de groupe).

Le clustering K-Means est une méthode non supervisée d’apprentissage automatique utilisée pour classer des éléments dans un ensemble de données en groupes ou “clusters”. Les éléments dans le même cluster sont plus similaires entre eux qu’avec ceux d’autres clusters.

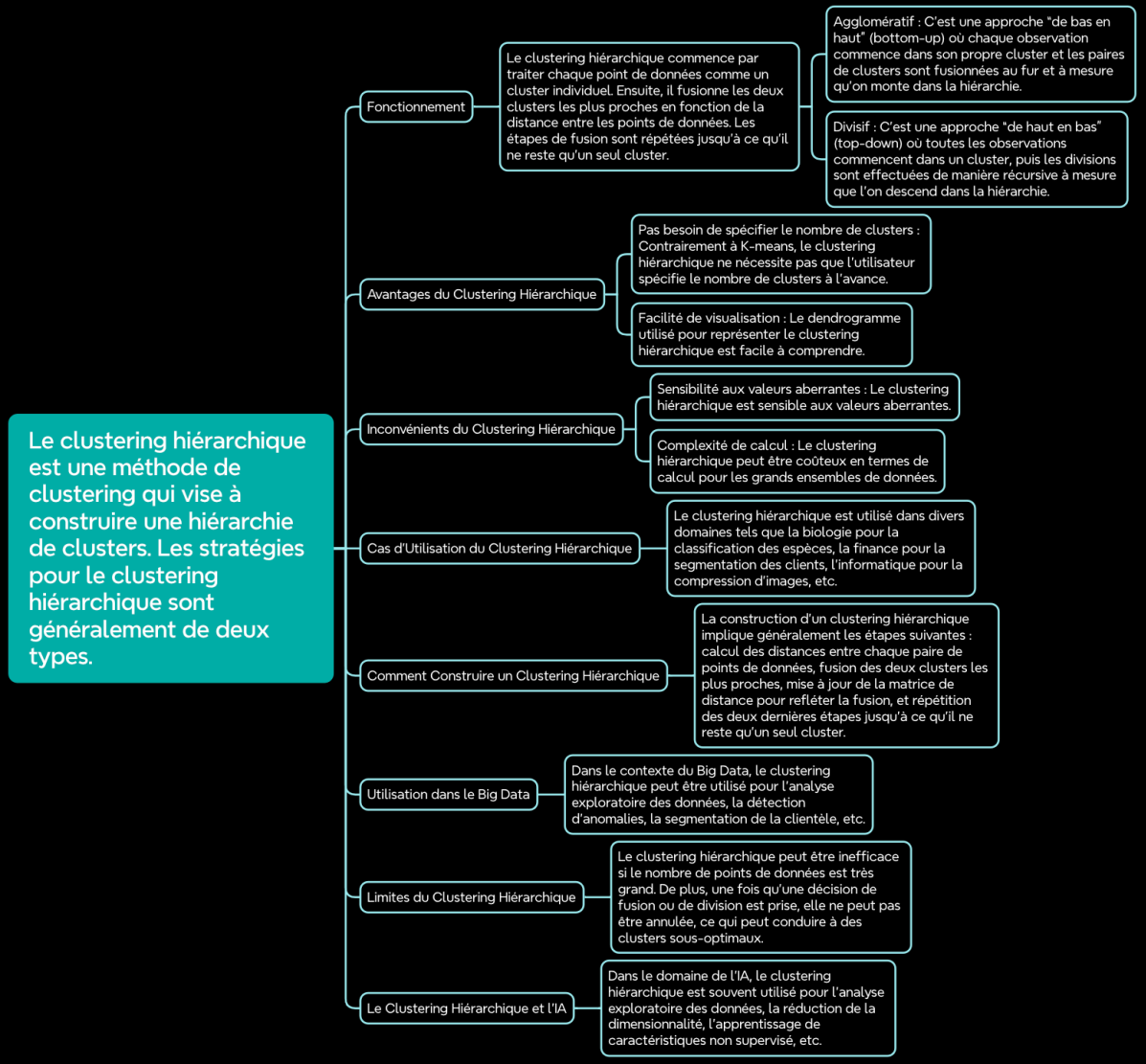
- Fonctionnement
  - L'algorithme K-Means fonctionne en initialisant aléatoirement “K” centres de clusters. Chaque point de données est ensuite attribué au centre de cluster le plus proche. Les centres de clusters sont recalculés comme le centre (moyenne) des points attribués et le processus est répété jusqu'à ce que les centres de clusters ne changent plus ou que le nombre maximum d'itérations soit atteint.
- Avantages du Clustering K-Means
  - Simplicité : L'algorithme K-Means est simple à comprendre et à mettre en œuvre.
  - Efficacité : Il est généralement rapide et efficace en termes de calcul.
- Inconvénients du Clustering K-Means
  - Initialisation : Le résultat peut varier en fonction de l'initialisation des centres de clusters.
  - Forme des clusters : K-Means a du mal avec les clusters non sphériques et de tailles différentes.
- Cas d'Utilisation du Clustering K-Means
  - Le clustering K-Means est utilisé dans divers domaines tels que le marketing pour la segmentation de la clientèle, l'informatique pour la compression d'images, la finance pour la détection de fraudes, etc.
- Comment Construire un Clustering K-Means
  - La construction d'un clustering K-Means implique généralement les étapes suivantes : sélection du nombre de clusters “K”, initialisation aléatoire des centres de clusters, attribution des points de données aux clusters, recalcul des centres de clusters, et répétition des deux dernières étapes jusqu'à convergence.
- Utilisation dans le Big Data
  - Dans le contexte du Big Data, le clustering K-Means peut être utilisé pour l'analyse exploratoire des données, la détection d'anomalies, la segmentation de la clientèle, etc.
- Limites du Clustering K-Means
  - K-Means peut être sensible à l'initialisation et aux valeurs aberrantes. De plus, il suppose que tous les clusters ont une variance égale, ce qui n'est pas toujours le cas dans les données réelles.
- Le Clustering K-Means et l'IA
  - Dans le domaine de l'IA, le clustering K-Means est souvent utilisé pour l'analyse exploratoire des données, la réduction de la dimensionnalité, l'apprentissage de caractéristiques non supervisé, etc.

L’algorithme de clustering hiérarchique

Imagine que tu as une boîte pleine de jouets différents et tu veux les organiser. Au lieu de les trier par une seule caractéristique (comme le type de jouet), tu décides de créer une structure d’organisation plus complexe.

D’abord, tu sépares les jouets en deux groupes principaux : les jouets mous (comme les peluches) et les jouets durs (comme les blocs de construction). Ensuite, tu divises chaque groupe en sous-groupes plus spécifiques. Par exemple, tu pourrais diviser les jouets mous en animaux en peluche et poupées. Tu pourrais diviser les jouets durs en blocs de construction en bois et blocs de construction en plastique. Tu continues à diviser chaque groupe jusqu’à ce que chaque jouet ait sa propre catégorie spécifique.

Dans le clustering hiérarchique, tu commences par regrouper les données les plus similaires. Ensuite, tu crées des groupes de plus en plus grands en combinant les groupes qui sont similaires entre eux. À la fin, tu obtiens une structure d’arbre, ou une hiérarchie, qui montre comment les données peuvent être regroupées à différents niveaux de similarité.



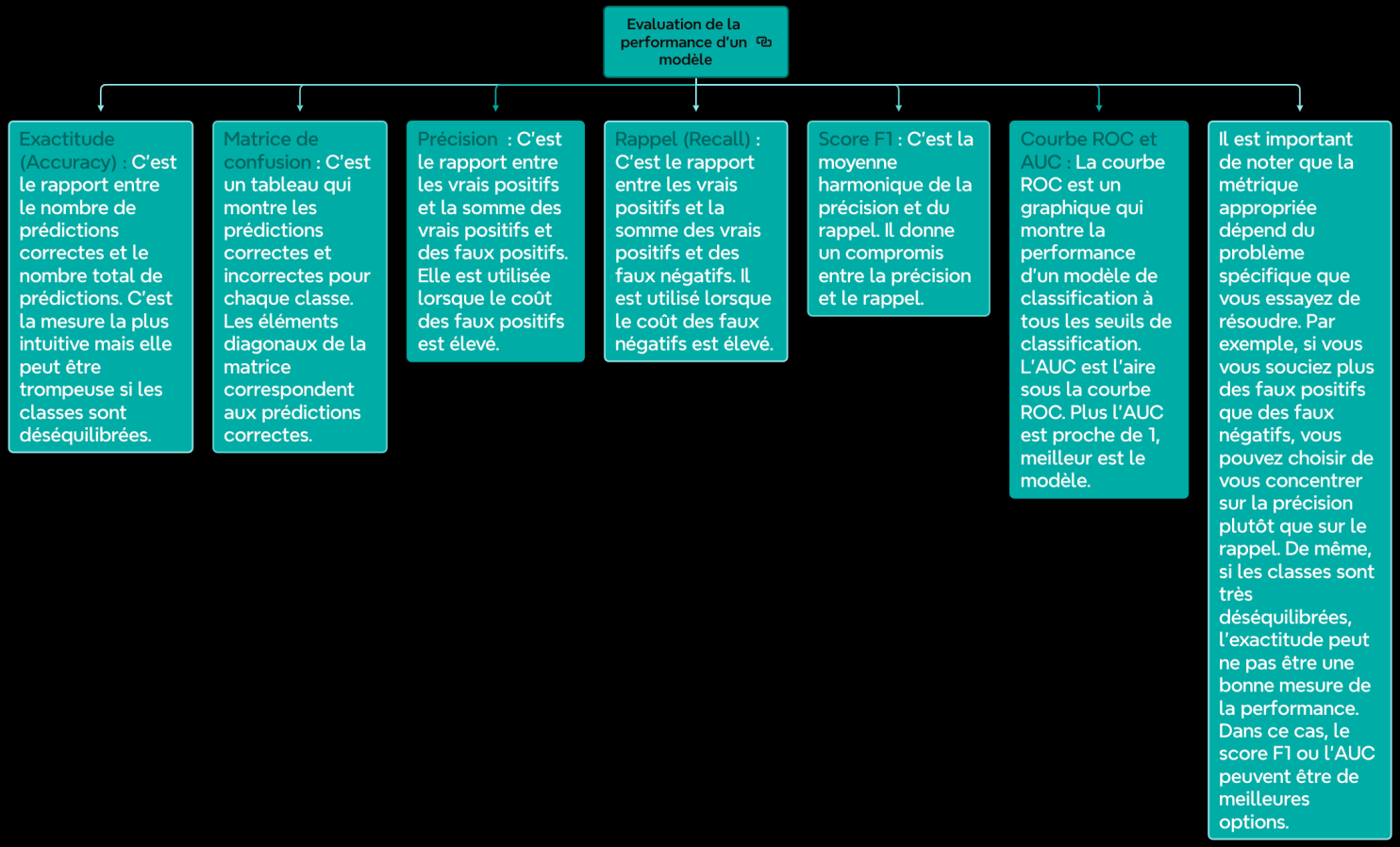
L’algorithme de clustering basé sur la densité

Nous sommes dans une pièce sombre avec une lampe de poche. Tu ne peux voir que les objets qui sont proches de toi lorsque tu pointes la lampe de poche vers eux. Si tu vois un groupe d’objets proches les uns des autres, tu peux supposer qu’ils forment un groupe. Si tu vois un objet isolé, loin des autres, tu peux supposer qu’il n’appartient à aucun groupe. Tu continues à explorer la pièce, en formant des groupes d’objets qui sont proches les uns des autres.

Avec le clustering basé sur la densité, tu commences par un point de données (comme tu commences par un objet dans la pièce). Ensuite, tu cherches d’autres points de données qui sont proches (comme tu cherches d’autres objets proches avec ta lampe de poche). Si tu trouves suffisamment de points proches, tu les considères comme faisant partie du même groupe (ou “cluster”). Si un point est trop éloigné des autres, tu le considères comme du « bruit » ou comme appartenant à un autre groupe.

L'ÉVALUATION DE LA PERFORMANCE D'UN MODELE

Maintenant que vous comprenez quel modele fait quoi, et comment, il est primordial de savoir évaluer leur performance. alors comment cela fonctionne-t-il ?



L’accuracy :

Nous jouons à un jeu de fléchettes. Chaque fois que tu lances une fléchette, tu essaies de la faire atterrir au centre de la cible (c’est un peu le but, on va pas se mentir). Parfois tu réussis, parfois tu manques. À la fin du jeu, tu comptes combien de fois tu as réussi à atteindre le centre de la cible et tu divises ce nombre par le nombre total de fléchettes que tu as lancées. Cela te donne une idée de ta **précision** - plus tu es précis, plus tu as de chances d’atteindre le centre de la cible.

L’accuracy d’un modèle de machine learning fonctionne de manière similaire. Chaque fois que le modèle fait une prédiction, il essaie de “toucher la cible” - c’est-à-dire de faire une prédiction correcte. L’accuracy est le nombre de fois où le modèle a fait une prédiction correcte divisé par le nombre total de prédictions. Plus l’accuracy est élevée, plus le modèle est bon pour faire des prédictions correctes.

La matrice de confusion :

Tu dois deviner le type de fruit dans un sac en te basant uniquement sur le toucher. Parfois, tu devines correctement, parfois tu te trompes. À la fin du jeu, tu décides de faire un tableau pour voir combien de fois tu as deviné correctement et combien de fois tu t’es trompé.

Dans ce tableau, tu mets les vrais types de fruits en colonnes et tes devinettes en lignes. Chaque case du tableau montre combien de fois tu as deviné un certain type de fruit quand le vrai fruit était d’un certain type. Par exemple, la case en haut à gauche pourrait montrer combien de fois tu as deviné “pomme” quand le vrai fruit était une pomme (ce sont les vrais positifs). La case en bas à droite pourrait montrer combien de fois tu as deviné “orange” quand le vrai fruit était une orange (ce sont aussi des vrais positifs). Les autres cases montrent combien de fois tu t’es trompé.

Une matrice de confusion est un tableau qui montre les prédictions correctes et incorrectes d’un modèle de classification. Elle peut t’aider à comprendre où ton modèle fait des erreurs et comment tu peux l’améliorer.

La précision :

Cette fois, tu dois lancer des anneaux sur des bouteilles. Chaque fois que tu lances un anneau, tu essaies de le faire atterrir sur une bouteille. Parfois tu réussis, parfois tu loupes. À la fin du jeu, tu comptes combien de fois tu as réussi à faire atterrir un anneau sur une bouteille et tu divises ce nombre par le nombre total de fois où tu as pensé avoir réussi. Cela te donne une idée de ta précision - plus tu es précis, plus tu as de chances de faire atterrir l'anneau sur la bouteille.

Chaque fois que le modèle fait une prédiction positive (par exemple, prédit qu'un email est un spam), il essaie de "toucher la cible" - c'est-à-dire de faire une prédiction correcte. La précision est le nombre de fois où le modèle a fait une prédiction positive correcte (vrais positifs) divisé par le nombre total de prédictions positives (vrais positifs et faux positifs). Plus la précision est élevée, plus le modèle est bon pour faire des prédictions positives correctes.

Le recall :

Jouons cette fois-ci à cache-cache avec nos amis. Certains de tes amis sont très bons pour se cacher et tu ne les trouves pas tous. À la fin du jeu, tu comptes combien de tes amis tu as réussi à trouver et tu divises ce nombre par le nombre total d'amis qui jouaient. Cela te donne une idée de ton "rappel" - plus ton rappel est élevé, plus tu es bon pour trouver tes amis.

Le Recall d'un modèle de machine learning fonctionne de manière similaire. Chaque fois que le modèle fait une prédiction, il essaie de "trouver" les vrais positifs (par exemple, les emails qui sont réellement du spam). Le Recall est le nombre de vrais positifs que le modèle a réussi à trouver divisé par le nombre total de vrais positifs. Plus le Recall est élevé, plus le modèle est bon pour trouver les vrais positifs.

Score F1 :

À présent, nous devons attraper des balles. Certaines balles sont rouges et d'autres sont bleues. Ton objectif est d'attraper toutes les balles rouges et d'éviter toutes les balles bleues. À la fin du jeu, tu veux savoir à quel point tu as bien joué. Pour cela, tu utilises deux mesures : la précision (combien de balles rouges tu as attrapées parmi toutes celles que tu as essayé d'attraper) et le rappel (combien de balles rouges tu as attrapées parmi toutes les balles rouges qui étaient dans le jeu).

Cependant, il est difficile de comparer ta performance en utilisant deux mesures séparées. Alors, tu décides de combiner la précision et le rappel en une seule mesure, appelée le score F1. Le score F1 est la moyenne harmonique de la précision et du rappel. Cela signifie qu'il tient compte à la fois de la précision et du rappel et donne un score élevé seulement si les deux sont élevés.

Le score F1 est utilisé pour évaluer la performance d'un modèle de classification lorsque les classes sont déséquilibrées (c'est-à-dire lorsqu'il y a beaucoup plus d'exemples de l'une classe que de l'autre).

Courbe ROC et AUC :

1.**Courbe ROC** : Continuons à attraper les balles rouges et éviter les balles bleues. Plus tu es bon dans le jeu, plus tu attrapes de balles rouges et moins tu touches de balles bleues. Si tu dessines un graphique montrant combien de balles rouges tu as attrapées (taux de vrais positifs) par rapport à combien de balles bleues tu as touchées (taux de faux positifs), tu obtiens ce qu'on appelle une courbe ROC (Receiver Operating Characteristic).

2.**AUC** : Maintenant, imagine que tu veux comparer ta performance dans le jeu avec celle de tes amis. Une façon de le faire est de regarder l'aire sous la courbe ROC (Area Under the Curve, ou AUC). Plus l'aire sous ta courbe ROC est grande, mieux tu es dans le jeu par rapport à tes amis. L'AUC est un nombre entre 0 et 1 - un AUC de 1 signifie que tu es parfait dans le jeu (tu attrapes toutes les balles rouges et tu évites toutes les balles bleues), tandis qu'un AUC de 0,5 signifie que ta performance n'est pas meilleure que le hasard (comme si tu fermes les yeux et attrapes les balles au hasard).

J’espère que désormais, vous y voyez plus clair sur ce que sont les modèles et à quoi ils servent.

Dans le cadre de la résolution d’un probleme complexe, combiner et enchainer les modèles est une impérieuse nécessité car l’usage d’un seul d’entre eux n’est pas forcement une fin en soit.

Je n’aborde pas ici les thématiques (pourtant passionnantes) du DEEP Learning et des réseaux neuronaux. Mais je vous invite à creuser ces sujets si cela vous intéresse.

CONCLUSION

Nous sommes déjà au terme de ce second livret du data framework lié au sujet de l'IA été de son déploiement dans le cadre d'une organisation.

Je vous remercie de l'avoir parcouru et j'espère qu'il vous a permis d'éclaircir les zones d'ombres qui pouvaient exister sur ces sujets avant que vous ne le lisiez. Je l'avais mentionné en préambule, l'objectif de ce livret n'est pas de faire de vous ds experts absolus sur ce sujet, mais de vous éclairer sur les notions fondamentales et les étapes nécessaires pour mettre en place sereinement votre stratégie IA dans votre organisation.

Que vous soyez dans le cadre d'une petite structure ou d'une vaste organisation, le tronc commun des connaissances nécessaire à faire de votre approche d'IA une réussite est désormais à votre portée.

Sachez vous armer des ressources et du recul nécessaire et ne sous estimez jamais le point le plus important de l'ensemble de cette démarche : savoir pendre du recul et ne pas se laisser engouffrer dans les modes et la fausse idée du résultat immédiat.

Ce sujet fait partie d'une stratégie DATA globale de votre organisation. Apres le livret relatif à la gouvernance, vous tenez désormais la seconde pièce du puzzle qui fera de votre stratégie data un succès.

A bientôt pour le prochain livret.

Jérôme

Dans la même série « Les cahiers data de Jérôme » :

- Livret 1 : Mode opératoire d'un processus de gouvernance de la donnée_ ***déjà disponible***
- Initier un projet data d'entreprise réussi _ *À paraitre*
- La data-visualisation ou comment rendre accessible l'information_ *À paraitre*
- Les data plateformes et leurs mises en oeuvre dans le cadre d'une organisation_ *À paraitre*
- La stratégie d'acculturation Data et sa mise en oeuvre_ *À paraitre*

« Les cahiers Data » sont des éléments du **data-framework**.