Web sémantiques, arbres et autres ML basés sur les arbres

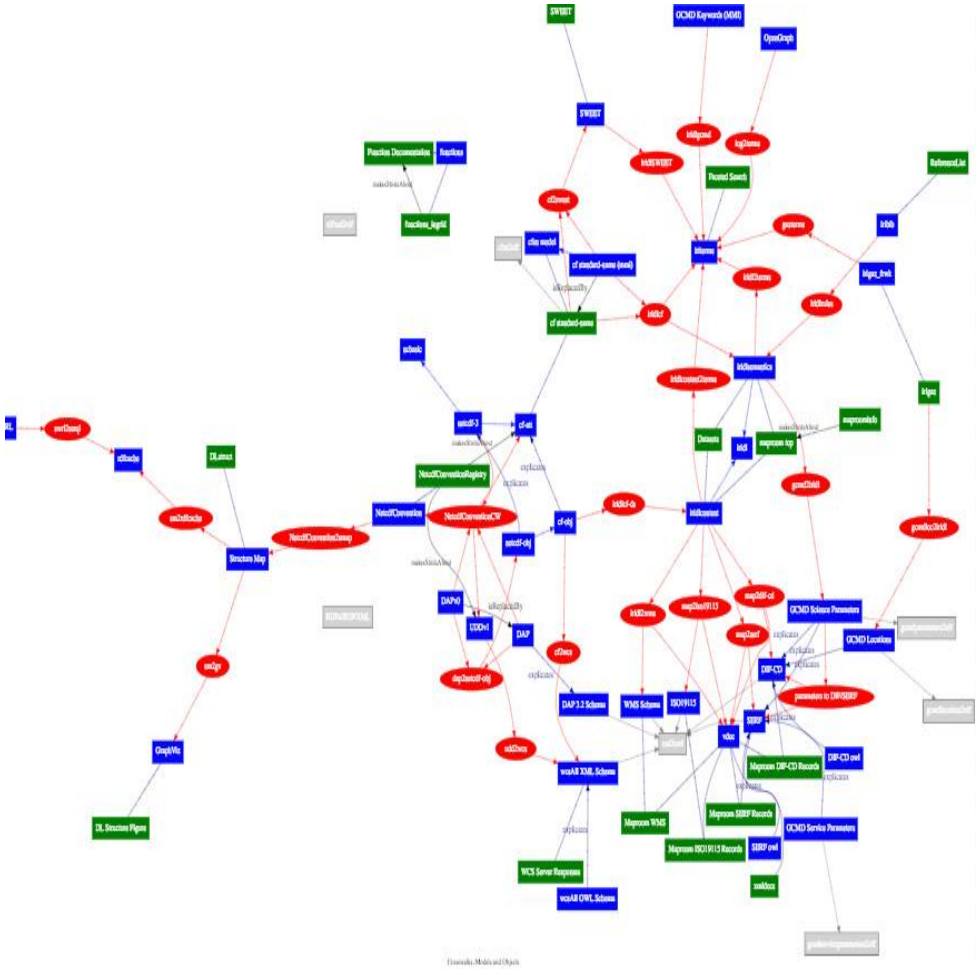
Dr. NSENGE MPIA HERITIER, Ph.D

Précédemment

- Réseaux sémantiques :
  - Facilité de mise en œuvre et d'interprétation
  - Très flexibles
  - Inférence simple grâce à l'héritage
- Cadres :
  - Extension des réseaux sémantiques basée sur le concept de typicité
  - Inférence plus efficace et plus claire
  - Fondement de nombreuses bases de connaissances du monde réel
  - Ni les réseaux sémantiques ni les cadres :
    - n'ont pas de sémantique standard
    - Peuvent limiter les applications et poser des problèmes d'inférence
- Scripts :
  - Étendent les cadres à des séquences de cadres ordonnées dans le temps
- Autres représentations liées aux réseaux sémantiques

Plan de la leçon

- Web sémantique
- Représentation arborescente
  - Arbres de décision
- Autres Machine Learning à base d'arbres



Web sémantique

Systèmes d'organisation des connaissances

- Folksonomie :
  - i.e. étiquetage collaboratif
  - Balises en texte libre

- Vocabulaire contrôlé :
  - Obligation d'utiliser des termes autorisés et acceptés par la communauté
  - Contrairement aux vocabulaires en langage naturel qui n'ont pas de telles restrictions.

- Taxonomie :
  - Arrangement hiérarchique de concepts (souvent utilisé comme "colonne vertébrale" d'une ontologie).

- Thésaurus :
  - Établit les relations entre les mots (souvent combiné à une taxonomie)
    - synonymes, homonymes, antonymes, etc.

- Ontologie :
  - Taxonomie + sémantique des concepts
  - Plus de types de relations, plus spécifiques dans leur fonction

- Web sémantique : (couvert dans ce chapitre)
  - Sémantique des données liées sur le web mondial

Valeur sémantique limitée

Valeur sémantique accrue

World Wide Web 1.0

- C’est un système Hypertexte/hypermédia distribué
- Information accessible par la recherche et la navigation (par mots clés)
- Les outils de navigation restituent l'information pour la consommation humaine.

Limites de la recherche sur le web aujourd'hui

- Les résultats de la recherche sur le Web ont un taux de rappel (recall) élevé et une faible précision
  - C’est-à-dire nous pouvons trouver un certain nombre de choses que nous recherchons mais qui sont accompagnées de beaucoup d'autres choses que nous ne recherchons pas

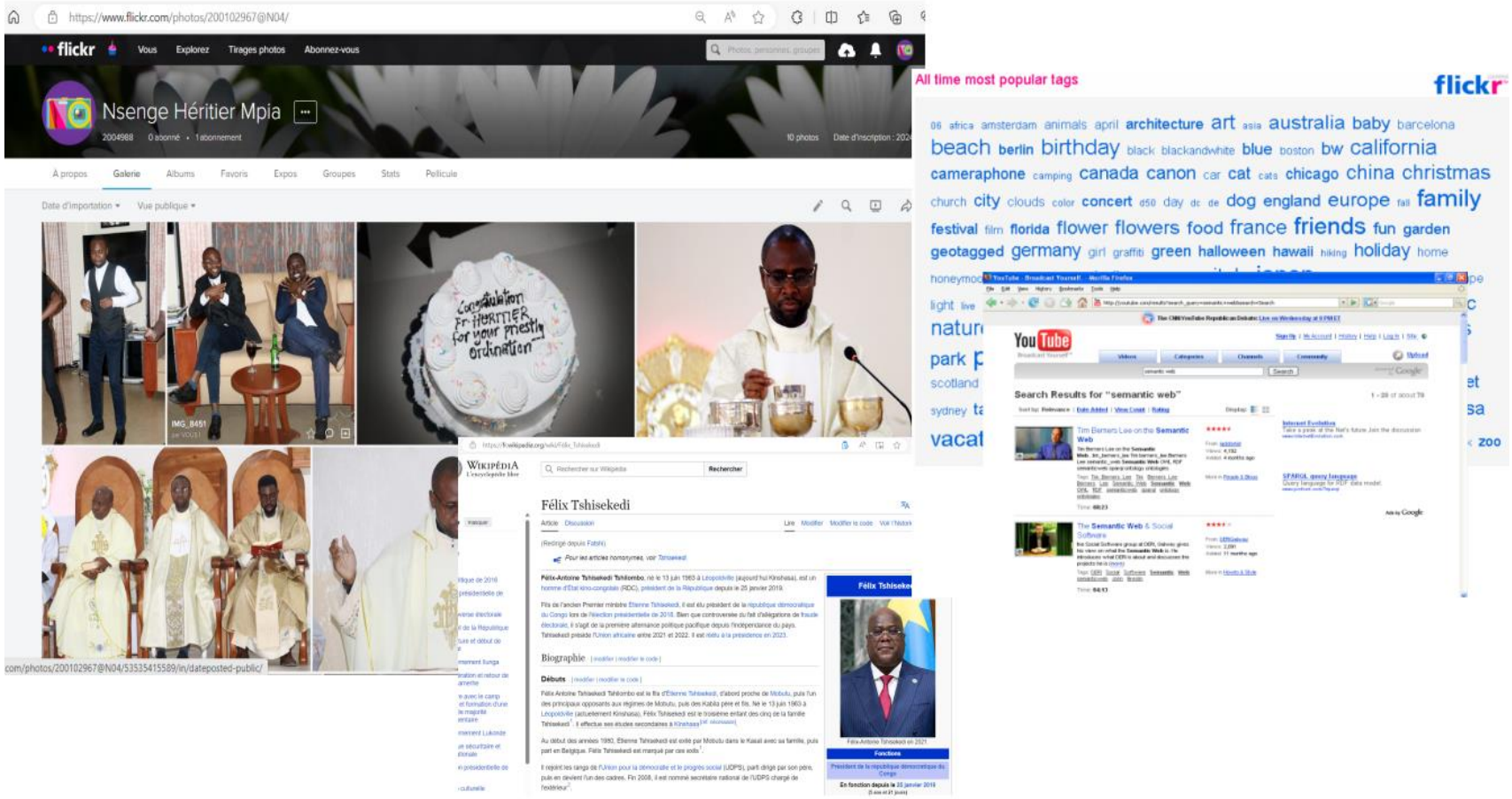
- Les résultats sont très sensibles au vocabulaire

- Les résultats sont des pages Web uniques

- La plupart des contenus de publication ne sont pas structurés de manière à permettre un raisonnement logique et à répondre aux requêtes.

Web 2.0 et Folksonomies

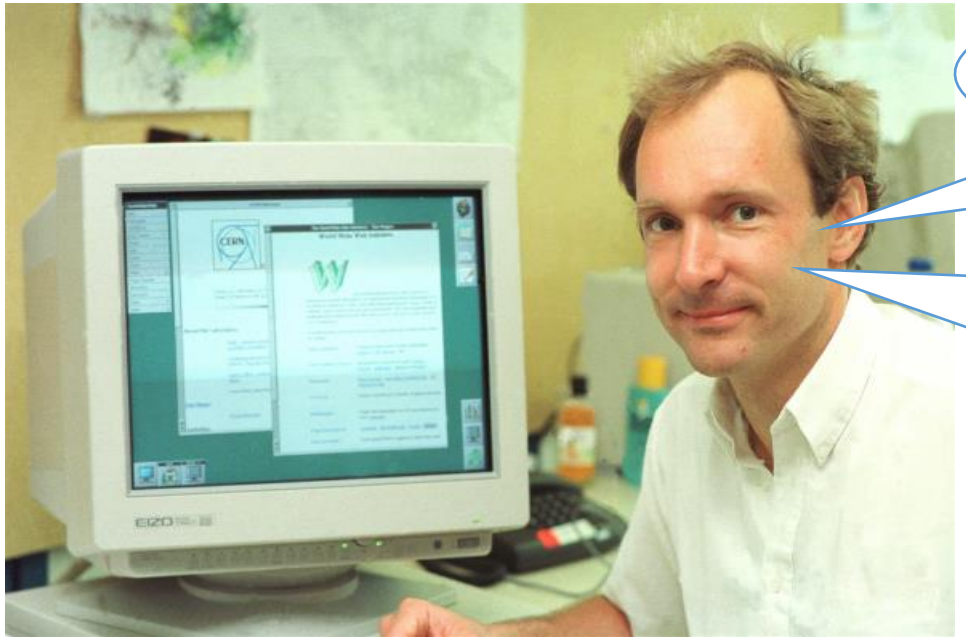
- Toutefois, certains signes indiquent que le web évolue vers quelque chose de nouveau qui pourrait être plus utile à l'avenir.
- L'une de ces étapes est l'inclusion de folksonomies (indexation personnelle).
- Grâce au crowdsourcing, des individus commencent à étiqueter toutes sortes d'entités sur le web afin de les rendre plus faciles à trouver et à consulter ; là encore, il peut s'agir d'images, de vidéos, de discussions, d'articles ou de n'importe quoi.



Qu'est-ce que le web sémantique ?

- Le Web a été "inventé" par Tim Berners-Lee (parmi d'autres), un physicien travaillant au CERN

- Sa vision du Web était beaucoup plus ambitieuse que la réalité du Web (syntaxique) existant :

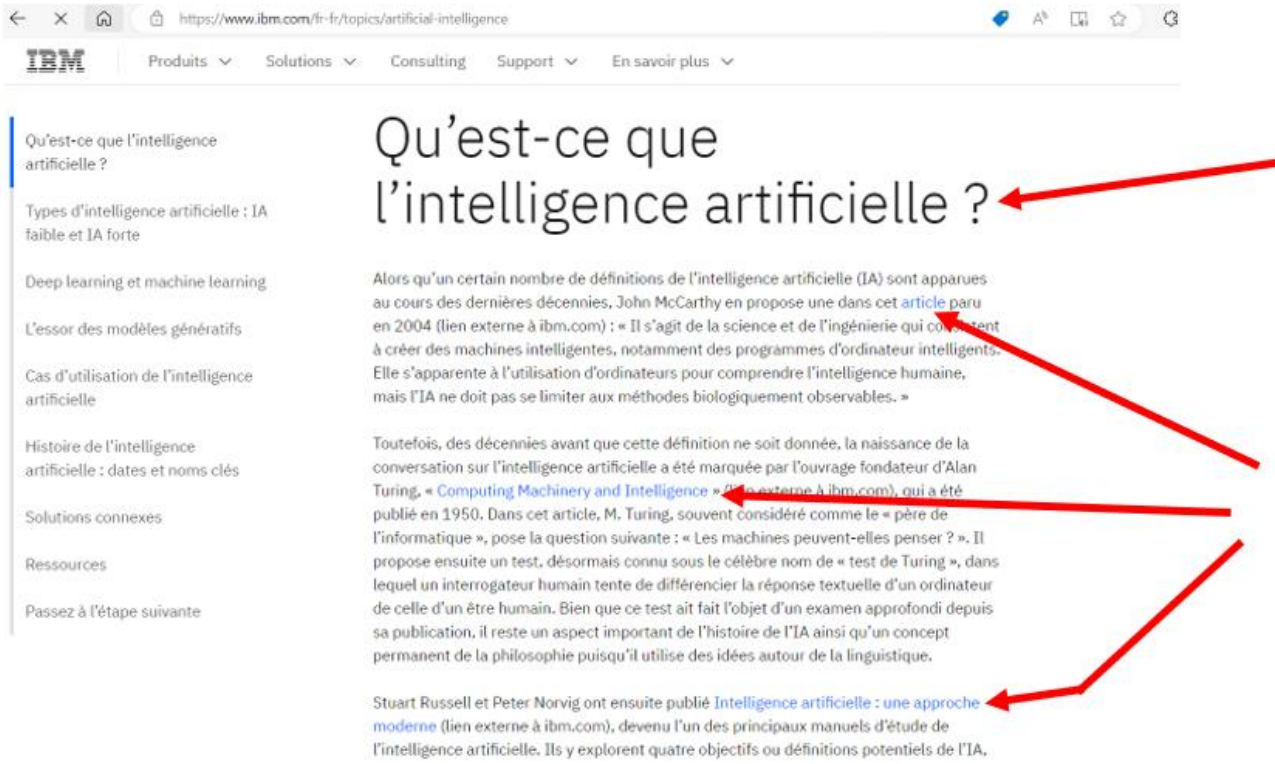


"... un ensemble d'applications connectées ... formant un réseau logique cohérent de données ..."

"... une extension du web actuel dans laquelle l'information est dotée d'une signification bien définie, ce qui permet aux ordinateurs et aux personnes de mieux travailler en coopération..."

- Cette vision du web est connue sous le nom de web sémantique.
- Note : Un web sémantique n'est pas la même chose qu'un réseau sémantique

Quel est le problème avec le Web ?



- Le problème du web, c'est qu'il est aujourd'hui typiquement configuré.
- Considérons une page web typique comme celle-ci-contre :
- Cette page est sous-tendue par un langage de balisage informatique qui se concentre sur le rendu d'informations telles que la taille et la couleur de la police
- et l'inclusion d'hyperliens vers des contenus connexes.
- Cette configuration est réellement conçue pour l'interface et l'interaction humaines, mais elle n'est jamais vraiment conçue pour que les ordinateurs puissent facilement naviguer et utiliser les pages.

Quelle est la solution (proposée) ?

- La solution proposée vise à aider les ordinateurs à utiliser et à naviguer plus facilement sur le web lorsqu'il s'agit d'ajouter des annotations sémantiques aux ressources du web.

Dr. Nsenge Mpia Héritier, Professeur Associé d'informatique

Université de l'Assomption au Congo, Butembo

Dr.<Personne>Nsenge Mpia Héritier <Personne>, <Job>Professeur Associé d'informatique <Job> <Employeur> Université de l'Assomption au Congo <Employeur>, <Localisation> Butembo <Localisation>

- Par exemple, voici ma photo de profil et quelques informations me concernant.
- Cependant, il ne s'agit que de texte brut, un ordinateur aurait besoin de beaucoup de connaissances de base pour comprendre que Butembo est une *ville*, que l'Université de l'Assomption au Congo est mon *employeur*, que Nsenge Mpia Héritier est mon *nom* et que Dr. est mon *grade* académique.
- Au lieu de cela, nous pouvons prendre ce texte brut et l'enrichir d'étiquettes (tags) sémantiques qui clarifient le sens sous-jacent de chaque morceau de texte.
- Dans le cas présent, nous avons les annotations sémantiques *Personne* encadrant mon nom, qui indiquent que je suis une personne. Le texte décrivant mon poste est entouré de l'annotation Job ; mon employeur est entouré de l'annotation *Employeur* et la *Localisation* est Butembo.
- Ces annotations sont destinées aux machines et non aux êtres humains qui peuvent déduire ce contexte sans aide.

Web sémantique (en quelque sorte)

- Voilà qui devrait éclaircir certains points.

Donner une sémantique aux annotations

- Une autre approche consiste à représenter le contenu du web sous une forme plus facilement accessible aux machines et à utiliser des techniques intelligentes pour tirer parti de ces présentations
- On a besoin d'un accord externe sur la signification des annotations
  - Flexibilité et extensibilité limitées
  - Nombre limité de choses pouvant être exprimées
- On peut utiliser les ontologies pour spécifier la signification des annotations
  - Signification des vocabulaires de termes donnés par les ontologies
    - Doit être normalisée (en fait dans la plupart de cas, les ontologies sont standardisées)
  - Possibilité de combiner/référencer des termes dans plusieurs ontologies

HTML et XML : Un pas en avant

- Nous pouvons utiliser HTML et XML qui sont tous deux des langages de création de pages web.
  - Le HTML décrit la présentation de la page et n'est pas intrinsèquement utilisable par l'homme.
  - En revanche, XML décrit un contenu lisible à la fois par l'homme et par la machine.
- Dans l'exemple HTML ci-dessous, on illustre essentiellement la manière dont ce texte sera formaté sur une page web donnée.
  - Le XML va plus loin que le XTML car il ne se contente pas de présenter le contenu, il le balise également afin que les machines puissent identifier et comprendre ce qu'est réellement ce

HTML :

```html
<h1> Intelligence Artificielle et Systèmes Experts</h1>
    <ul>
        <li>Code: PRSY045</li>
        <li>Etudiants: Master 1</li>
    </ul>
```

XML :

```xml
<cours>
    <titre>Intelligence Artificielle et Systèmes Experts</titre>
    <code>PRSY045</code>
    <etudiants>Master 1</etudiants>
</cours>
```

- Note : Cependant, ni HTML ni XML ne véhiculent de significations sémantiques ou n'utilisent une terminologie standard pour le contenu.

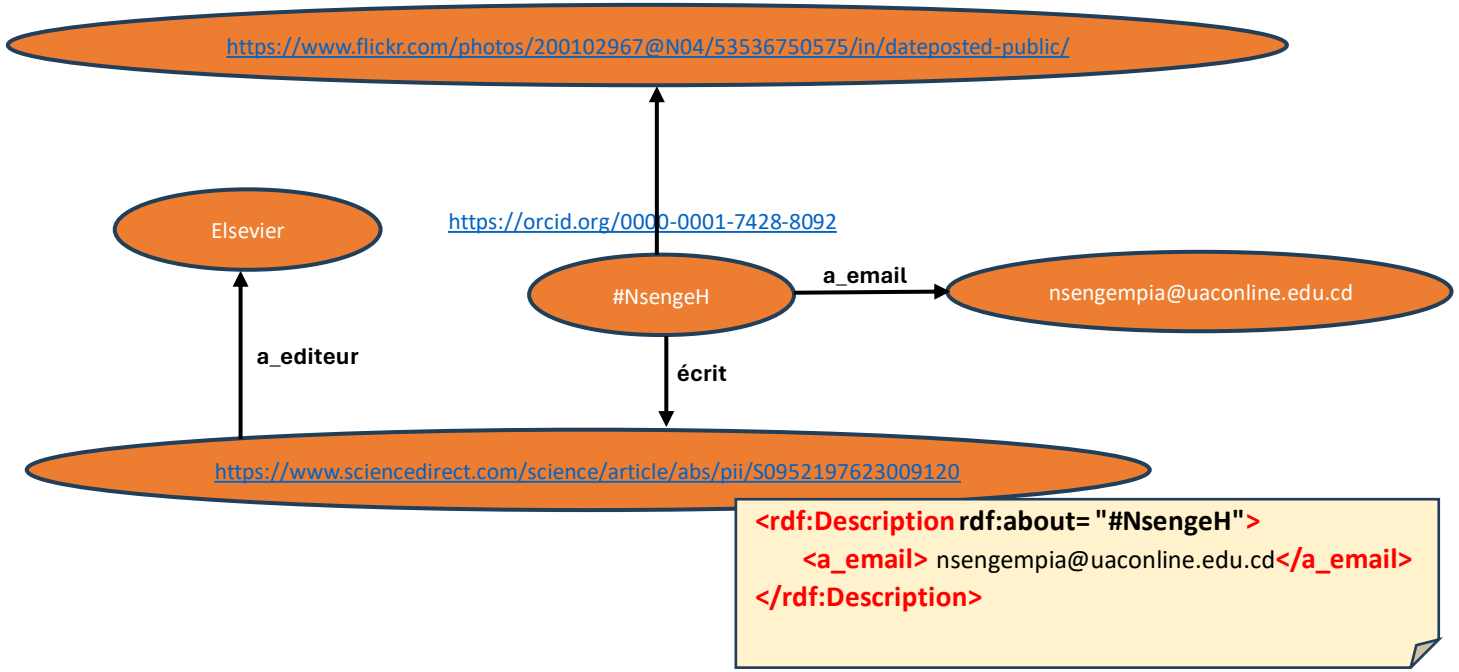
Cadre de description des ressources (RDF)

- Le RDF constitue la deuxième étape dans le processus d'annotation sémantique
- C'est une norme du World Wide Web Consortium (W3C)
  - W3C est la principale organisation internationale de normalisation pour le www
- Ajoute des relations entre les documents
  - Constitué de triples ou de phrases :
    - <sujet, propriété, objet>
    - Exemple: <"André Lufwa", sculpta, "le batteur de tam-tam symbolisant la FIKIN">

- Le schéma RDF étend RDF avec un vocabulaire ontologique standard :
  - Classe est une Propriété
  - Type est le Sous-classe de
  - Domaine spécifie une Plage
- Mais : Pas de signification décrite avec précision, ni de modèle d'inférence

RDF pour l'annotation sémantique

- RDF fournit des métadonnées sur les ressources web. Il utilise les triplets *objet*, *attribut* et *valeur*.
- RDF s'appuie sur la syntaxe XML, et n'importe lequel de ces triplets peut former un graphe.
- Exemple :
  - Commençons par un lien hypertexte vers ma page ORCID. Dans ce cas, un objet représentant mon nom a un email (nsengempia@uaconline.edu.cd) et l'email lui-même est la valeur (NB : ci-dessous dans l'image du rectangle, nous pouvons voir comment cela pourrait être écrit en RDF).
    - Remarquez que cela ressemble un peu à la façon dont les nœuds jouent un rôle dans un réseau sémantique.
  - Nous pouvons également voir d'autres triplets impliqués dans la création d'un graphe sous-jacent : *#NsengeH* a écrit un article et cet article a l'éditeur *Elsevier*.
  - Le RDF peut également être utilisé pour mettre en évidence des relations entre d'autres éléments qui ne sont pas simplement du texte. Par exemple, nous établissons un lien entre mon nom et ma photo, qui se trouve être disponible sur le lien ci-dessus.

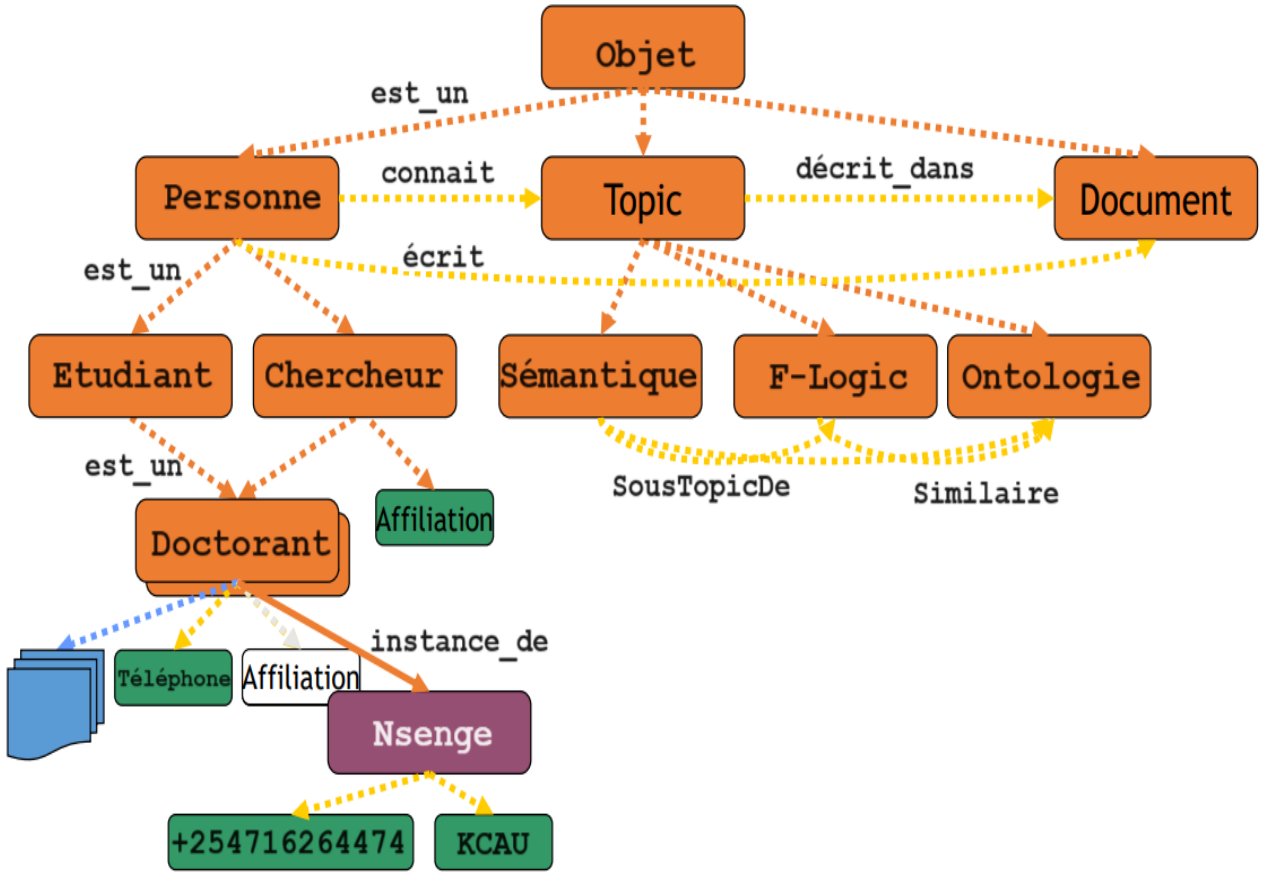


Langage d'ontologie web(OWL) & OWL2

- Prochaine étape dans la notion d'annotation: Ajouter des ontologies pour créer un web sémantique
  - Ajouter des normes terminologiques ontologiques et permettre l'inférence
- OWL est un langage d'ontologie web pour le web sémantique
  - OWL est une recommandation du W3C (c'est-à-dire une norme).
- OWL est basé sur le formalisme de représentation des connaissances de la logique de description (DL).
  - OWL bénéficie de nombreuses années de recherche en DL :
  - Sémantique bien définie
  - Propriétés formelles bien comprises (complexité, décidabilité)
  - Algorithmes de raisonnement connus
  - Systèmes mis en œuvre (hautement optimisés)
- Trois "espèces" d'OWL
  - OWL full est l'union de la syntaxe OWL et de RDF
  - OWL DL limité à un fragment de logique du premier ordre
  - OWL Lite est un sous-ensemble d'OWL DL "plus facile à mettre en œuvre".
- OWL 1 à OWL 2 :
  - Des types de données améliorés pour plus de flexibilité et un développement continu

Exemple d'ontologie à l'origine d'un web sémantique

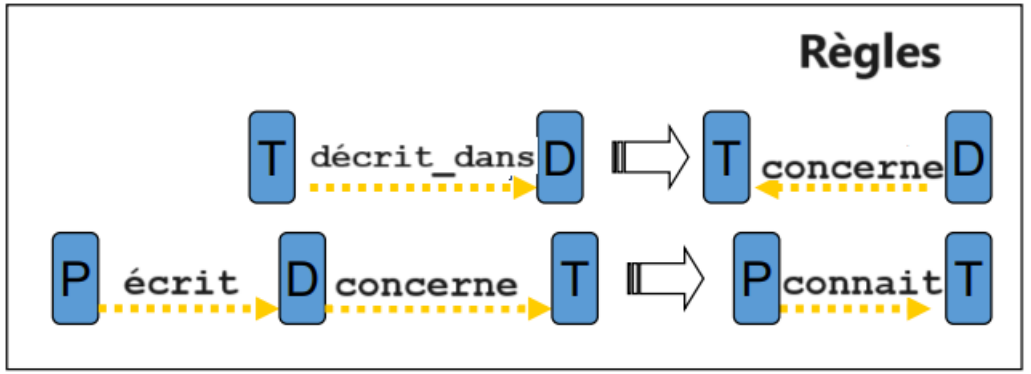
- Dans ce slide, nous essayons d'illustrer un exemple d'ontologie qui pourrait fonctionner dans les coulisses d'un site web sémantique.
  - Les nœuds en orange et les connexions illustrent l'ontologie de base qui fonctionne en coulisses dans cet exemple de web sémantique.
  - Nous avons donc des éléments tels qu'un doctorant est un étudiant mais aussi un chercheur, et un étudiant comme un chercheur sont des personnes ; et une personne est un objet, un *objet étant le type le plus général d'entité dans cette ontologie*.
- Nous pouvons constater qu'il existe un certain nombre d'autres connexions ou relations entre différents types d'objets.
- Nous pouvons également voir que différents objets sont connectés à des éléments qui pourraient être disponibles sur le net. Ici, l'objet "étudiant en doctorat" est généralement associé à des informations spécifiques. Dans ce cas, il peut s'agir d'articles qu'il a publiés, de son numéro de téléphone et de son affiliation.
- Grâce à ces relations normalisées, nous pouvons créer une instance (cf. instance Nsenge) d'un doctorant à laquelle sont associés tous ces éléments, qui peuvent être des éléments disponibles sur le web et liés à des ressources.



Exemple d'ontologie à l'origine d'un web sémantique (Cont.)

- Ce type d'ontologie et de web sémantique (créés dans le slide précédent) peut s'appuyer sur un certain nombre de règles.
  - Ces règles nous permettent d'effectuer certaines déductions automatiques de base.

- Par exemple, si un topic (T) est décrit dans un document (D), nous pouvons également conclure que ce document (D) traite de ce topic (T) ou si une personne (P) écrit un document (D) sur un certain topic (T), nous pouvons conclure que cette personne (P) connaît ce topic (T).



- Cet exemple illustre essentiellement la manière dont nous pouvons utiliser une ontologie avec des relations.
  - Nous pouvons l'utiliser pour créer un réseau sémantique reliant les ressources sur le web et, grâce à cette ontologie formellement établie avec des règles d'inférence prédéfinies, nous pouvons automatiquement tirer des conclusions sur certaines des ressources présentes sur le web.
  - Cela peut nous aider à trouver plus facilement les informations que nous recherchons et à le faire avec une plus grande précision.
- Les principaux paradigmes d'inférence sur le web sémantique sont issus de *la programmation logique* et de la *logique de description* et reposent sur des normes telles que RDF et OWL.

Logique de description (DL)

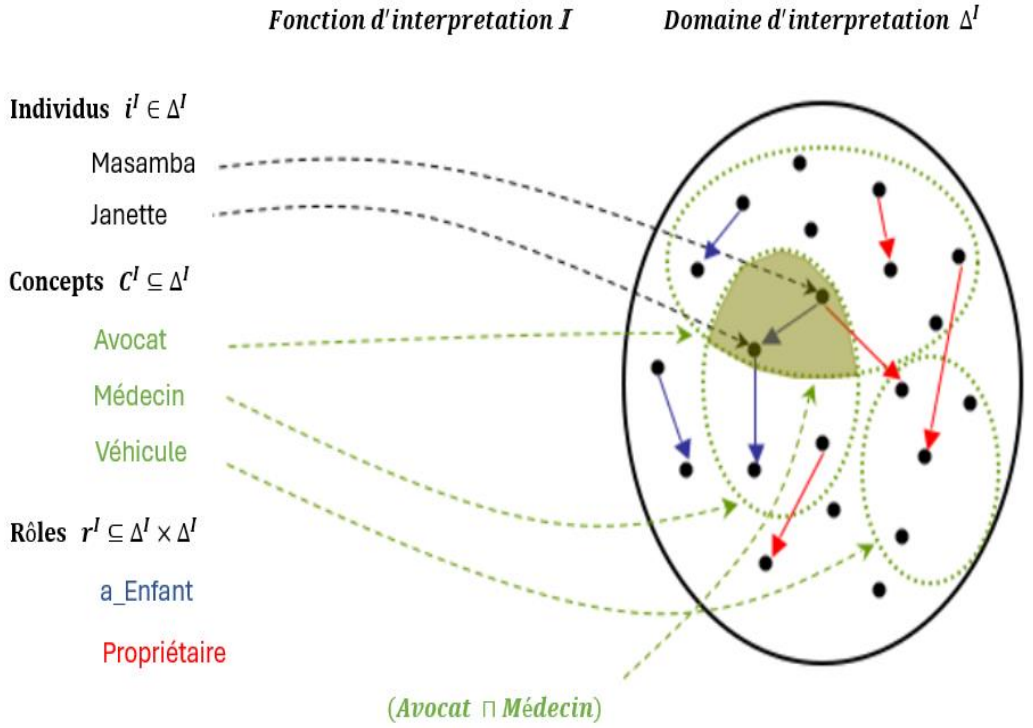
- C’est une famille de langages formels de représentation des connaissances
  - Plus expressif que la logique propositionnelle mais moins expressif que la logique du premier ordre (FOL)
- Modélise les concepts, les rôles et les individus, ainsi que leurs relations
- Le concept de modélisation fondamental d'un DL est l'axiome - une proposition logique reliant les rôles et/ou les concepts

- Utilise une terminologie différente

FOL	OWL	DL
Prédicat unaire	Classe	Concept
Constante	Individu	Individu
Prédicat binaire	Propriété	Rôle

Logique de description (DL)(Cont.)

- Le DL est un outil de raisonnement ontologique dont *l'objectif est de répondre à des requêtes portant sur des classes et des instances d'ontologie*.
  - Par exemple, pour trouver des classes plus générales ou plus spécifiques ou pour retrouver des individus ou des tuples correspondant à une requête donnée.
- L'image ci-contre illustre cette idée :
  - nous avons différents concepts d'avocat, de médecin et de véhicule, qui sont représentés de manière bidimensionnelle par ces ovales verts en pointillés.
  - Les individus sont ici des points (noirs dans le cercle) qui illustrent les personnes, dans ce cas Masamba et Janette.
  - Les rôles sont les relations entre les individus. Dans ce cas, une flèche bleue correspond à «a enfant» et une flèche rouge à «propriétaire».
  - Imaginons donc que ce grand cercle représente notre web sémantique basé sur la logique de description et que nous voulions rechercher des individus qui soient à la fois avocats et médecins, nous pourrions appliquer la logique de description dans ce cas pour trouver les individus Masamba et Janette qui se trouvent être à la fois avocats et médecins.



Avantages des logiques de description

- DL possèdent deux tâches d'inférence
  - la subsomption: c'est-à-dire vérifier si une catégorie est un sous-ensemble d'une autre sur la base de leurs descriptions
  - la classification: c'est-à-dire vérifier si un objet appartient à une catégorie
- Toutes ces deux tâches ont une complexité polynomiale
  - Cela reflète essentiellement la complexité de la recherche nécessaire pour obtenir une réponse et la complexité polynomiale est beaucoup moins coûteuse que la complexité exponentielle.
- Les DL ont une sémantique claire, ce qui en fait un bon formalisme de représentation des connaissances pour les applications à forte composante déclarative
- Le langage de Web sémantique OWL est basé sur les DL.

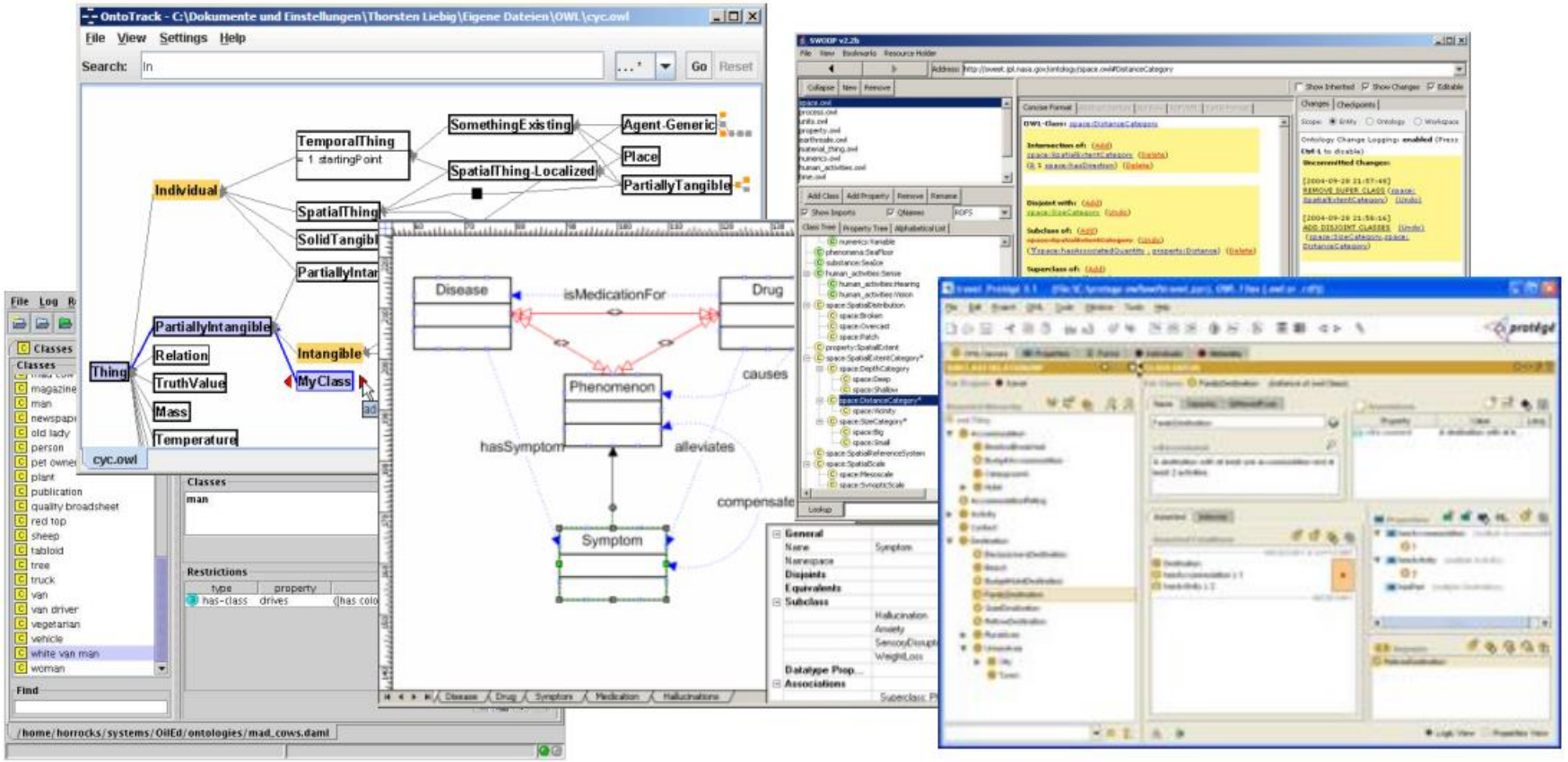
Inconvénients des logiques de description

- L'inférence polynomiale est assurée par les définitions des catégories
  - Si elles sont petites et correctement définies, les tâches d'inférence sont faciles à réaliser
  - Mais si les catégories sont difficiles à définir de manière concise, leurs descriptions peuvent devenir (exponentiellement) grandes $O(2^n)$.

- Des capacités d'inférence faibles (pour préserver la décidabilité), qui ne peuvent pas gérer des concepts spécifiés de manière imprécise et des relations entre eux qui ne sont pas entièrement appliquées

Outils et infrastructure

- Si vous souhaitez en savoir plus sur le développement du web sémantique ou des logiques de description, il existe un certain nombre d'éditeurs ou d'environnements différents dans lesquels vous pouvez les construire et les modifier.
- Le plus populaire est probablement Protégé, mais il en existe d'autres, notamment Oiled, Swoop, Ontotrack Construct, etc.
- Certains de ces outils sont également parfaits pour la construction d'otologies.



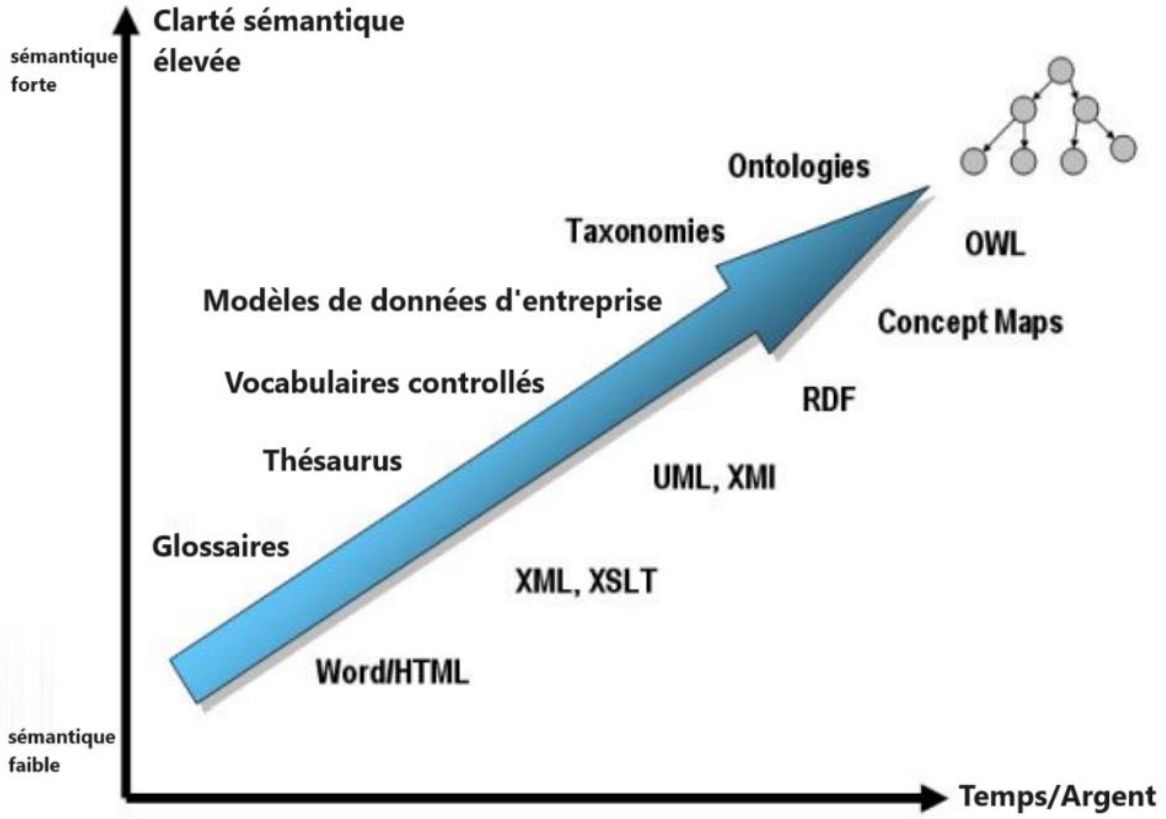
Protégé : Editeur d'ontologie

- Un éditeur d'ontologie et un cadre de travail libres et gratuits pour la construction de systèmes intelligents

- https://protege.stanford.edu/
- https://www.youtube.com/watch?v=uxR-bmCE_eU

Coût de la clarté sémantique

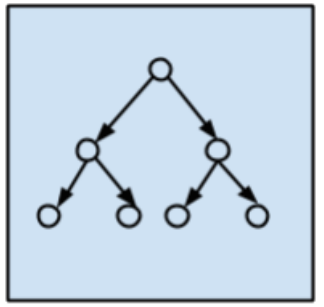
- La clarté du web sémantique implique des compromis.
- La figure très approximative ci-contre illustre le compromis entre une clarté sémantique élevée et le temps et l'argent consacrés à la création et à la maintenance.
- Il est évident qu'un web sémantique ou une ontologie bien développés demandent beaucoup de temps, d'efforts, d'organisation et d'expertise.



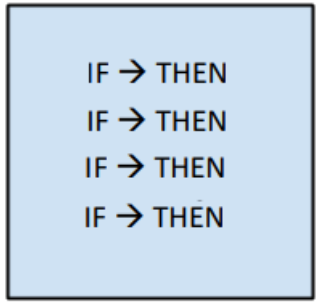
Représentation arborescente avec arbres de décision

Représentations d'arbres et de règles

- Différentes représentations des connaissances sont utilisées dans l'IA
- Nous examinerons ensuite deux représentations qui sont également utilisées couramment dans l'apprentissage automatique (ML)
  - les arbres (par exemple les arbres de décision)
  - les règles (par exemple les systèmes basés sur des règles) (prochain chapitre)
- Ces représentations peuvent stocker des "connaissances d'expert" existantes pour les utiliser dans des systèmes de raisonnement tels que les systèmes experts
- ou l'apprentissage inductif peut être appliqué pour former ces représentations à stocker des relations probabilistes à partir d'exemples spécifiques.

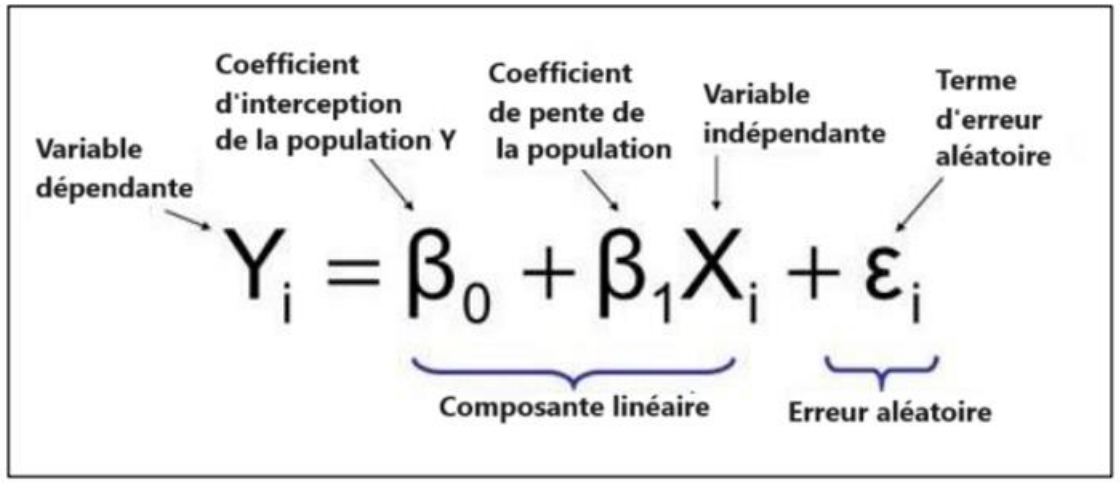


Arbre de décision

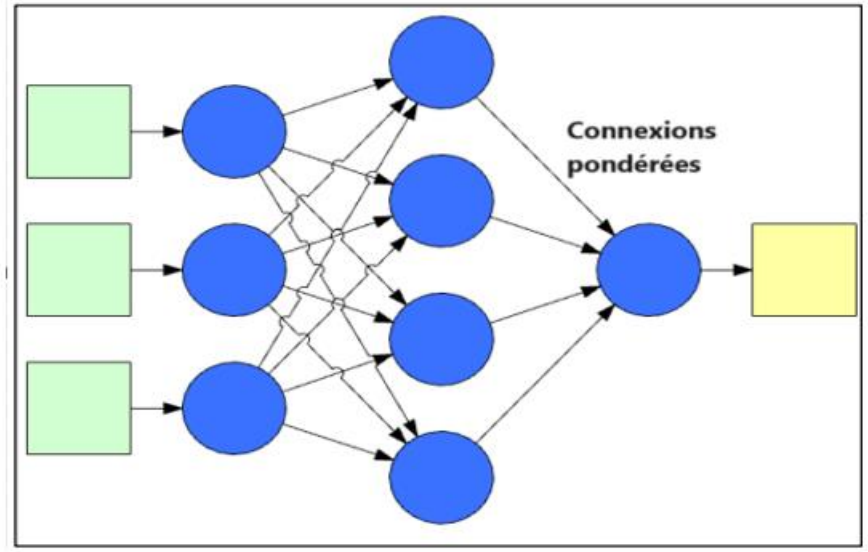


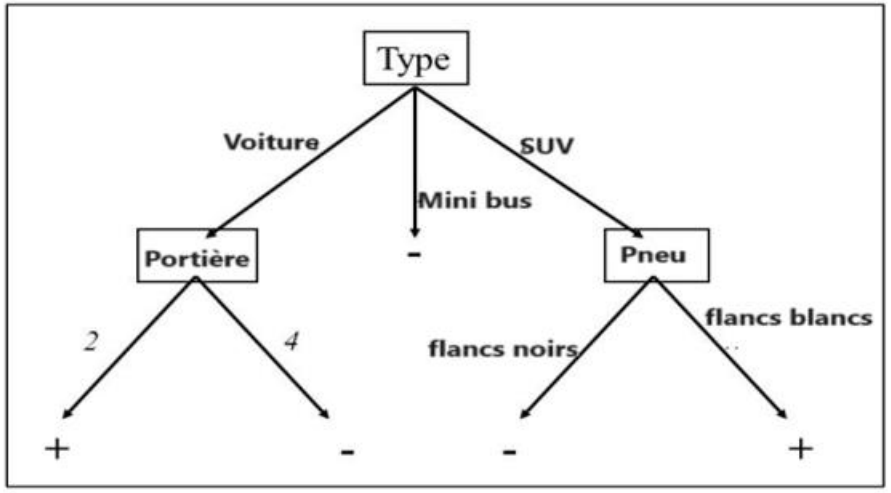
Basé sur les règles

Apprentissage automatique (ML) : Représentation

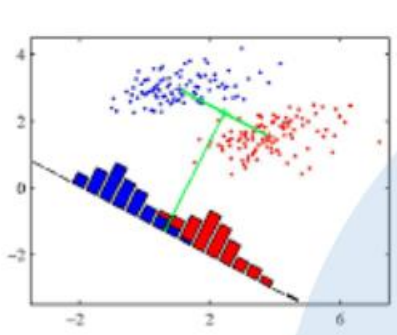


- **R1:** IF l'animal a des poils
  THEN c'est un mammifère
- **R2:** IF l'animal donne du lait
  THEN c'est un mammifère
- **R3:** IF l'animal a des plumes
  THEN c'est un oiseau
- **R4:** IF l'animal vole
  l'animal pond des œufs
  THEN c'est un oiseau
- **R5:** IF l'animal est un mammifère
  l'animal mange de la viande
  THEN c'est un carnivore

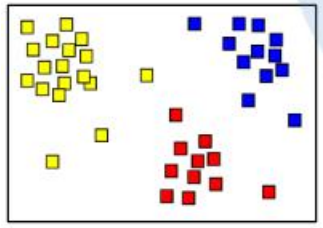


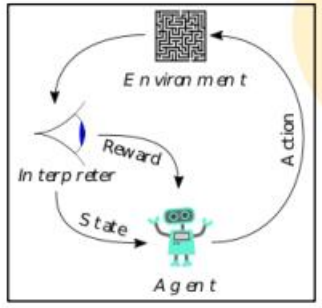


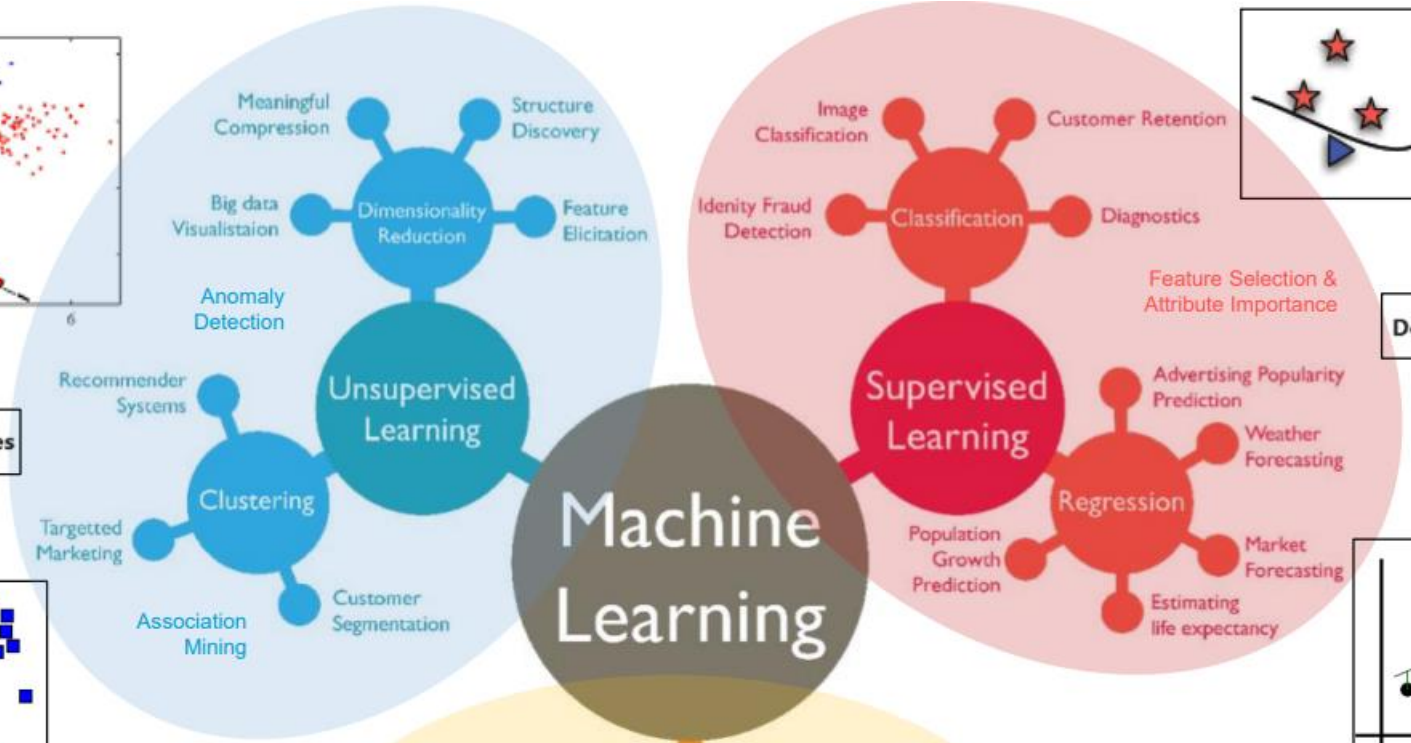
Types de Machine Learning



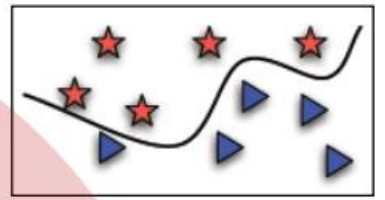
Données non étiquetées



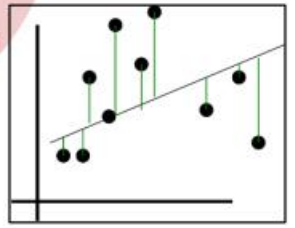




Problèmes à plusieurs étapes

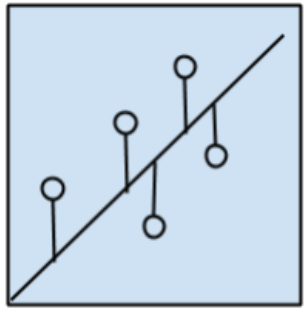


Données étiquetées

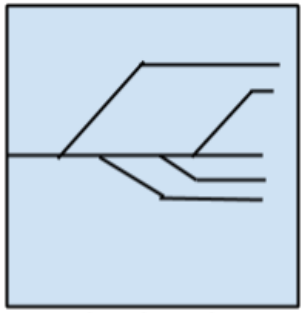


Adapté de : *https://www.pinterest.com/pin/786792997374742269/*

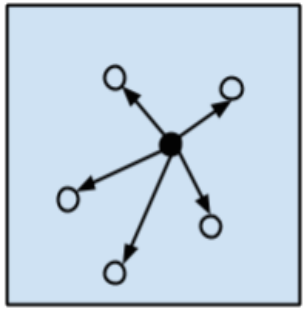
Familles d'algorithmes d'apprentissage automatique



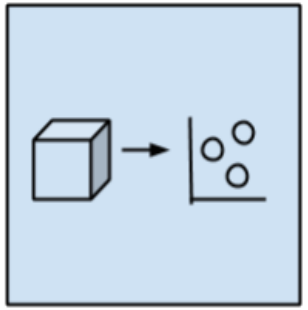
Algorithmes de regression



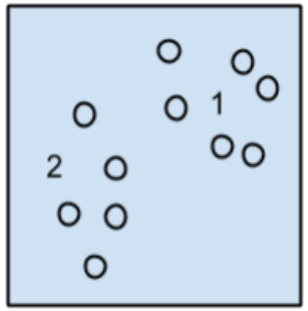
Algorithmes de régularisation



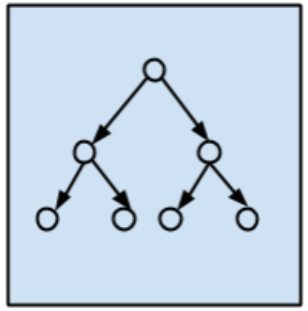
Algorithmes basés sur des instances



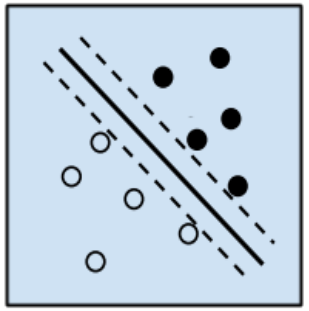
Algorithmes de réduction dimensionnelle



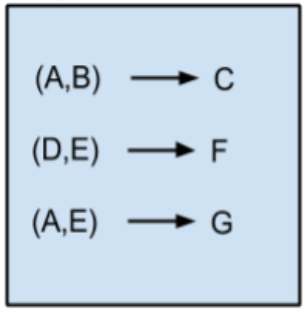
Algorithmes de Clustering



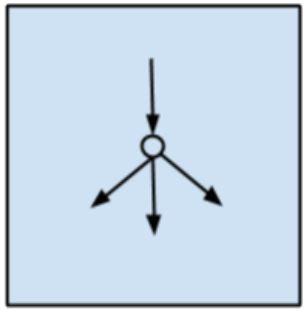
Arbres de décision



Machines à vecteurs de support

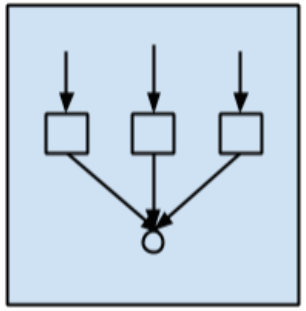


Algorithmes d'apprentissage de règles d'association



Algorithmes de réseaux de neurones artificiels

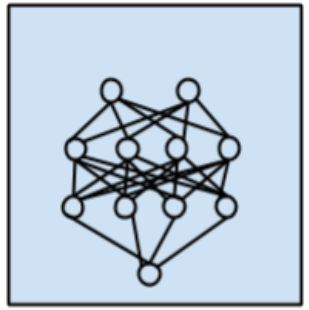
Algorithmes Bayésiens



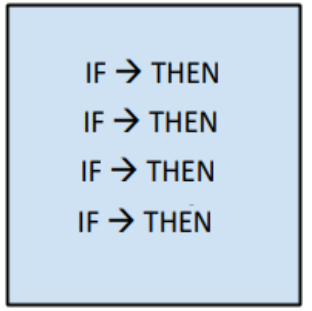
Algorithmes d'ensemble

Algorithmes évolutionnaires

Liste non exhaustive des familles ML



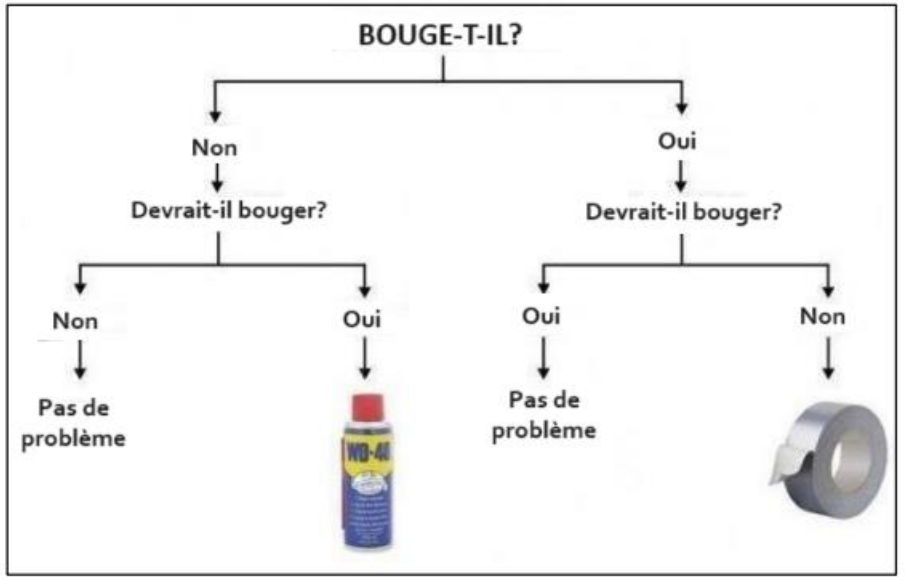
Algorithmes de Deep Learning



Apprentissage des systèmes de classification

Arbre de décision : Qu'est-ce que c'est ?

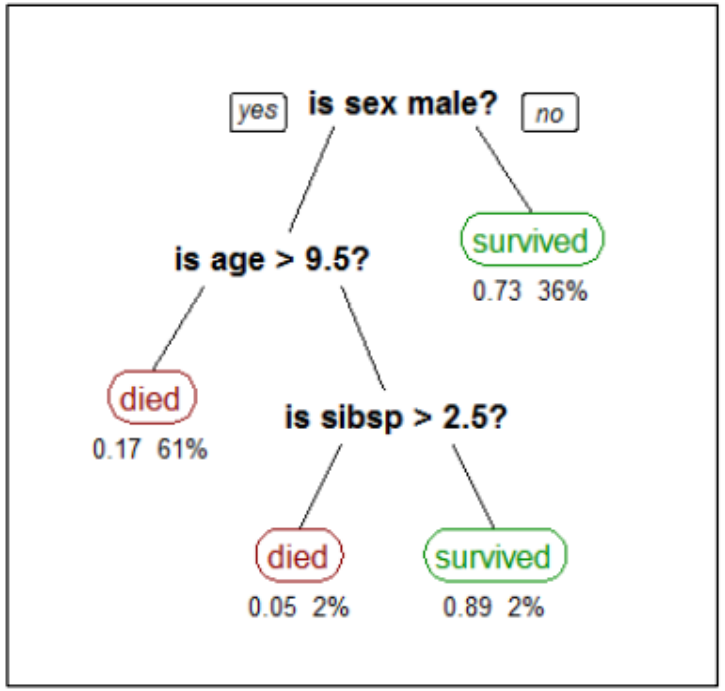
- Lorsque nous parlons d'arbres de décision, la première chose à noter est qu'il s'agit à la fois d'une représentation et d'une stratégie d'apprentissage automatique.
- La définition la plus élémentaire d'un arbre de décision est
  - une structure graphique de type organigramme où chaque nœud interne représente un test dans un attribut, chaque branche représentant le résultat du test.
- En d'autres termes, le test (à partir de l'image ci-contre) pourrait être "bouge-t-il ?". "oui ou non ?"
  - Chaque branche représente le résultat du test d'un attribut.
    - Vous pouvez faire le lien avec la proposition de vérité ou de fausseté de la logique propositionnelle.
  - Plus loin dans l'arbre, chaque nœud inférieur ou feuille représente un conséquent.
  - Au bas de l'arbre, nous avons donc différents nœuds feuilles et une série de questions :
    - est-ce qu'il bouge ? Oui ;
    - devrait-il bouger ? Non ;
  - et notre conséquence est d'utiliser du ruban adhésif (en arbre de décision, la décision est prise après avoir calculé tous les attributs dans un chemin vers le bas de l'arbre).



- Un arbre de décision peut être utilisé comme
  - outil d'aide à la décision
  - ou comme moyen de représenter des informations déjà connues
  - ou de les présenter en vue d'une prise de décision et d'en évaluer les conséquences ou le coût.
- Ainsi, dans l'arbre très simple ci-contre, nous avons pris nos propres connaissances préalables et nous avons créé un arbre à partir de celles-ci.
- N.B:
  - Il n'y a pas eu d'apprentissage.
  - Nous avons simplement pris nos connaissances
  - et un processus de prise de décision les a structurées sous la forme d'un arbre de décision.

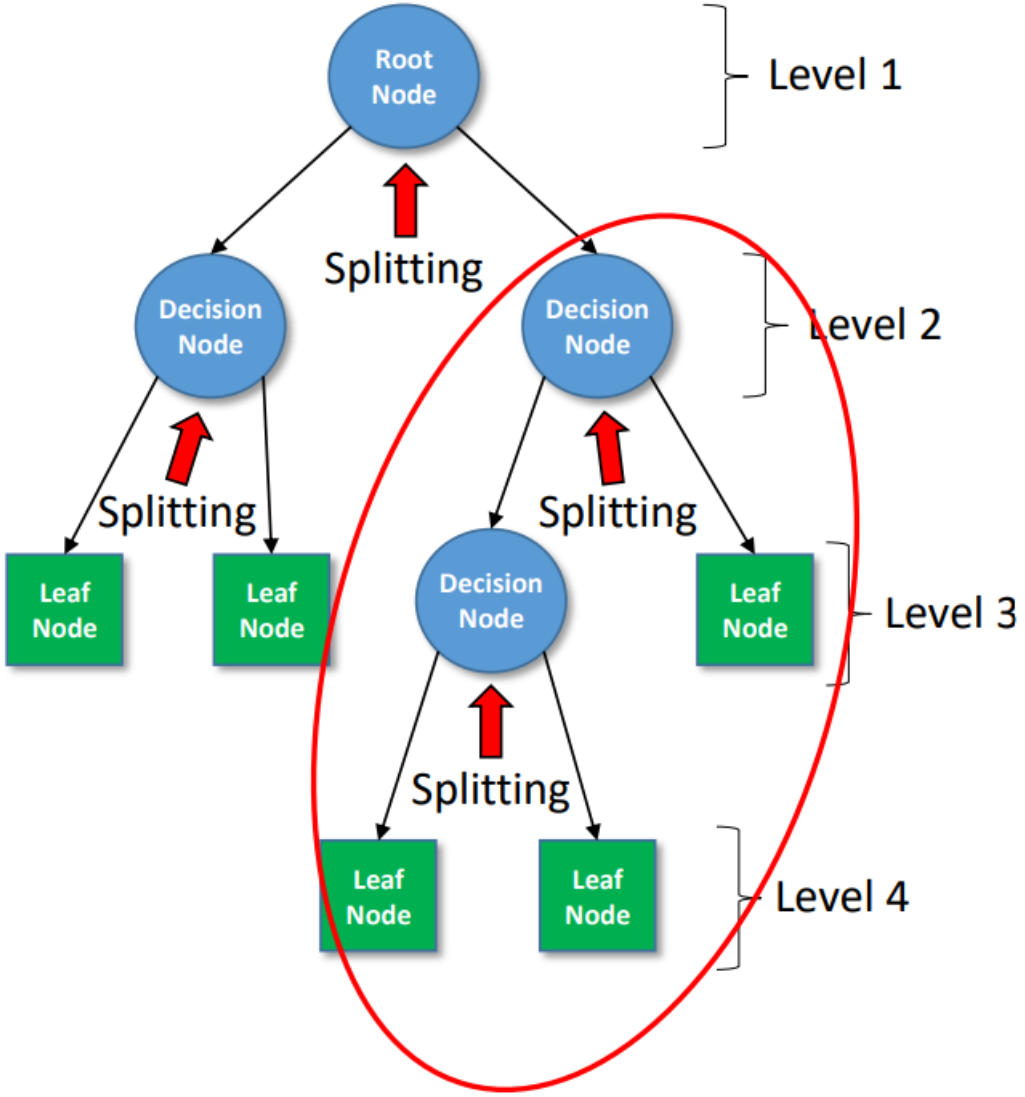
Arbre de décision : Qu'est-ce que c'est ? (Cont.)

- Les arbres de décision peuvent également être appliqués en tant qu'algorithme d'apprentissage automatique supervisé pour *modéliser* et *prédire les résultats*.
- Dans cette situation, nous pourrions donc construire un arbre en séquence en utilisant un ensemble de données de formation.
- Par exemple, vous pouvez former un arbre qui ressemble à notre image ci-contre.
  - Le sexe est-il masculin ? Oui ou non ?
  - Si ce non, la majorité des gens ont survécu.
  - S'ils sont de sexe masculin et que leur âge est supérieur à 9.5 ans, la plupart d'entre eux sont décédés (dans ce cas, les conséquences sont survivants et décédés).
- Les pourcentages ci-dessous indiquent la proportion d'individus décédés dans un état donné qui ont été sauvés dans la branche partant de ce nœud foliaire jusqu'au sommet de l'arbre.



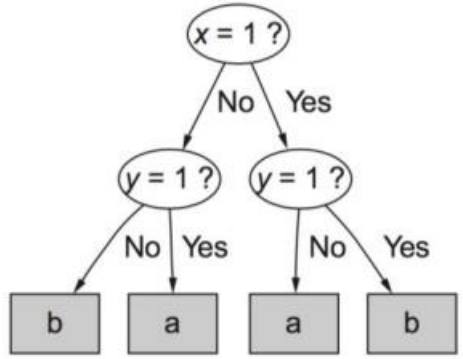
Arbre de décision : Terminologie

- **Nœuds** (nodes) :
  - Racine (root) : elle représente l'ensemble de la population ou de l'échantillon. Il sera divisé en deux ou plusieurs ensembles homogènes.
  - Décision : Lorsqu'un sous-nœud se divise en d'autres sous-nœuds, il est appelé nœud de décision (Aussi connu comme : nœud secondaire, interne, de division ou de hasard)
  - Feuille (leaf): Nœuds qui ne se divisent pas.
    - Donne la classe ou la valeur moyenne (Aussi connue comme : nœud terminal ou nœud de résultat)
  - Parent et enfant : Le nœud parent se divise en nœuds descendants.
- **Splitting** : Il s'agit d'un processus de division d'un nœud en deux ou plusieurs sous-nœuds.
- **Branche / sous-arbre** : Une sous-section de l'arbre entier est appelée branche ou sous-arbre.
- **Niveaux/profondeur**(Level/Depth) : Le nombre de divisions par un chemin donné dans l'arbre.



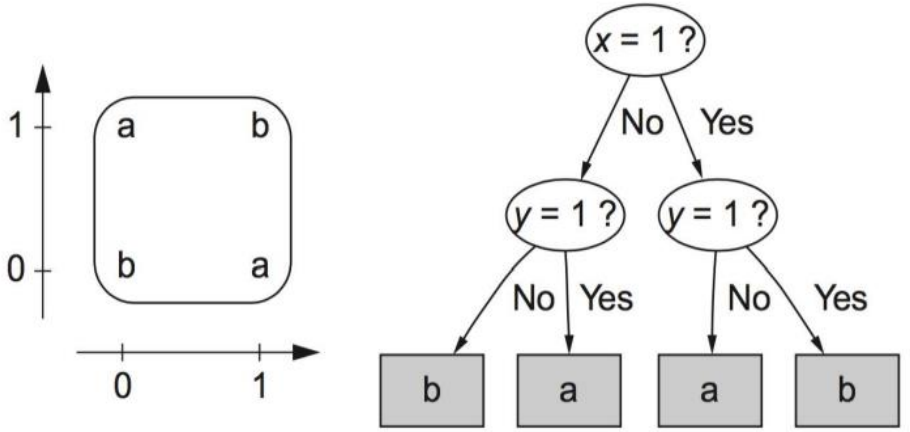
Arbres de décision

- À partir de maintenant, nous allons considérer les arbres de décision du point de vue de l'apprentissage automatique, c'est-à-dire que nous essayons de former un arbre à partir de zéro plutôt que d'en coder un à l'aide de nos propres connaissances d'expert.
- La formation des arbres adopte une approche de type "diviser pour mieux régner".
  - Ainsi, lorsque nous essayons de former un arbre, nous commençons par choisir le meilleur nœud racine possible.
  - Nous essayons de trouver un attribut qui nous permet de diviser au mieux les instances de notre ensemble de données. Le test d'un nœud pourrait être une valeur d'attribut comparée à une constante.
  - Par exemple, *x* est égal à 1?. C'est la question que nous posons ou le test que nous posons à ce nœud.
- Cependant, vous pouvez également comparer une fonction d'un ou plusieurs attributs comme étant la meilleure à un nœud.
- Les feuilles représentent les conséquences et peuvent produire une conclusion, une classe, une action ou un ensemble de ces éléments.
- En ce qui concerne la classification, ces conséquences ne donnent pas seulement la prédiction de la classe, mais elles représentent une *distribution de probabilités* spécifiant le soutien ou la certitude d'un résultat de classe donné.
- Nous *étudierons plus en détail cette ligne de pensée lorsque nous aborderons* le "raisonnement avec l'incertitude" dans un chapitre ultérieur.
- Chaque fois que nous avons une nouvelle instance pour laquelle nous voulons faire une prédiction ou utiliser notre arbre, elle arrive par le nœud racine et, sur la base des valeurs de cette instance, nous répondons à chaque test jusqu'à ce que nous arrivions au bon conséquent, qui est notre nœud de décision ultime dans l'arbre de décision.
  - En d'autres termes, si l'instance qui arrive n'a pas la valeur *x*, nous ne pouvons pas continuer.
- On peut considérer que ce processus est *similaire à l'instanciation de la logique.*



Les arbres de décision en termes de logique

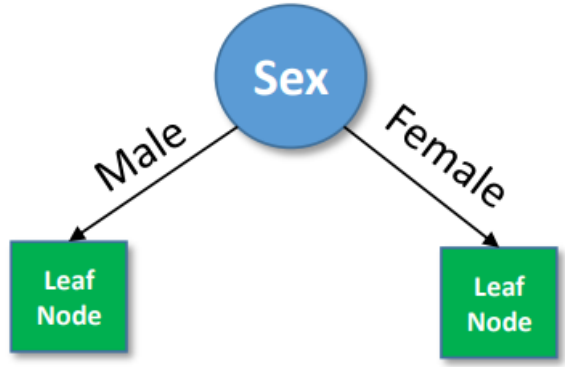
- Pour rester dans le domaine de la logique, on peut considérer les arbres de décision comme une hiérarchie d'implications.
- Par exemple, si $x = 1 \wedge y = 0 \rightarrow a$.
  - Cette implication décrit un chemin spécifique à travers l'arbre jusqu'à la conclusion *a*.
- Ci-contre, nous avons les quatre chemins différents à travers cet arbre de décision.
- En effet, du point de vue de la logique, un arbre de décision peut être *considéré comme une disjonction de conjonctions de contraintes sur les valeurs des attributs d'une instance*.
  - $Action\ if (A \wedge B \wedge C) \vee (A \wedge \neg B \wedge D) \vee (\ldots) \ldots$
    - La conjonction de contraintes sur les valeurs d'attribut fait allusion au fait que nous attribuons des valeurs spécifiques à ces variables.
    - La composante disjonction de cette définition fait référence au fait qu'il existe de multiples façons d'aboutir à *a* ou d'aboutir à *b*.



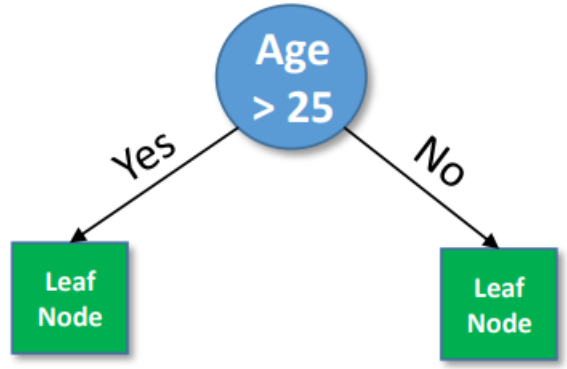

```
If x=1 and y=0 then class = a
If x=0 and y=1 then class = a
If x=0 and y=0 then class = b
If x=1 and y=1 then class = b
```

Arbre de décision : Fractionnement par…

- Caractéristiques catégorielles (c'est-à-dire nominales)
  - Revenons rapidement en arrière et examinons comment les arbres de décision peuvent être divisés selon que l'on considère une caractéristique catégorielle ou numérique.
  - Dans l'exemple en haut, nous avons une caractéristique catégorielle, à savoir le sexe.
  - Pour les besoins de cette illustration, imaginons que le sexe soit uniquement masculin ou féminin et que nous ayons donc une division définie par une valeur discrète. Dans ce cas, notre test est ses et nous pouvons passer d'une branche à l'autre.

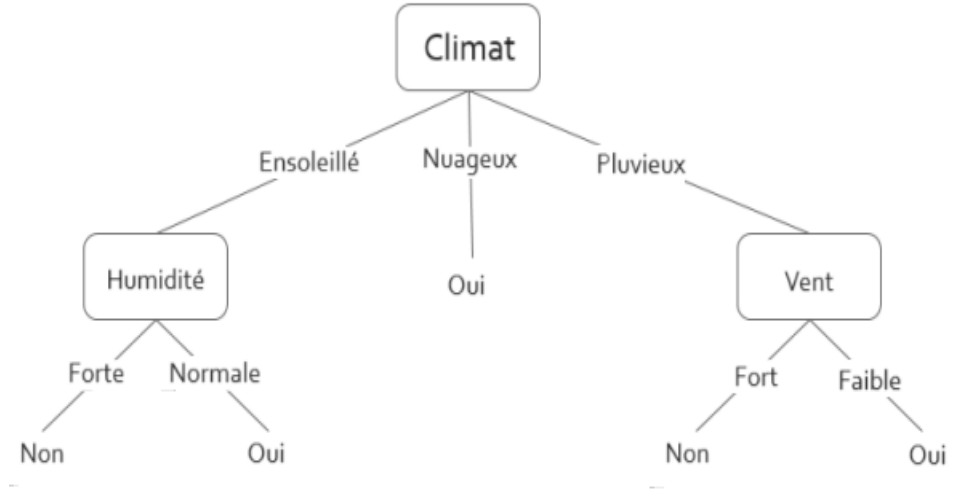


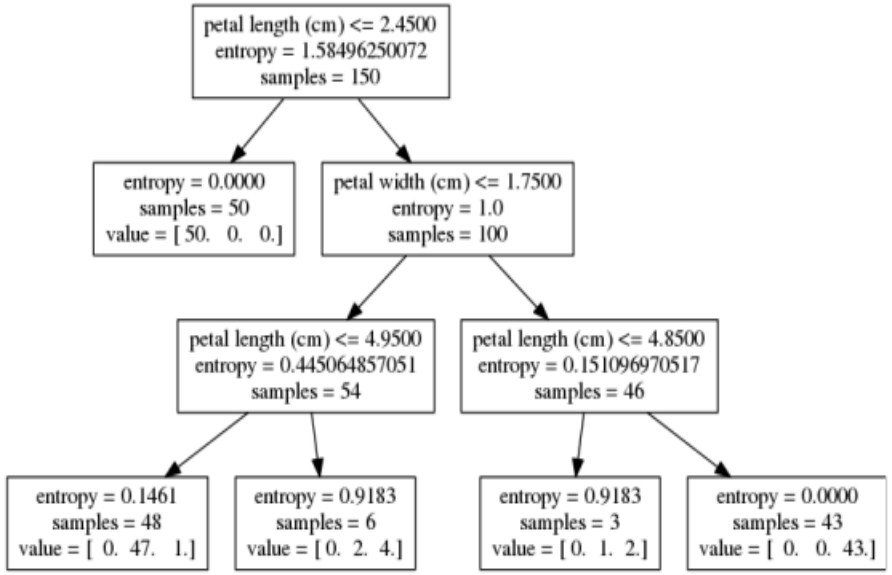
- Caractéristiques à valeur continue (c'est-à-dire numériques)
  - Pour les caractéristiques à valeur continue telles que l'âge, nous définirons cette division par une sorte de seuil, par exemple un âge supérieur à 25 ans. Là encore, nous disposons d'un nombre discret de choix pour continuer à descendre dans l'arbre.



Attributs nominaux et numériques dans les arbres

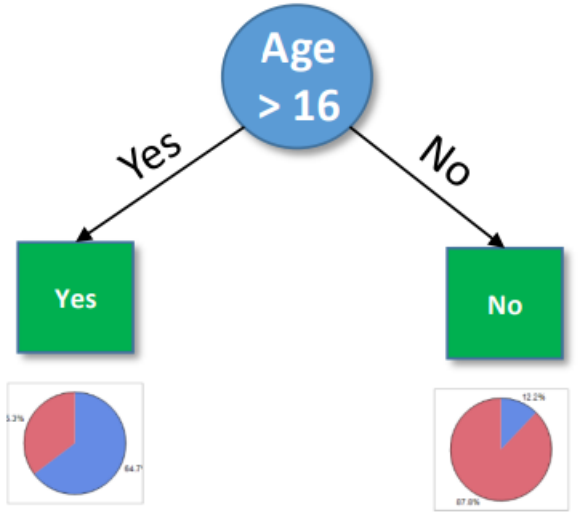
- Nominal :
  - Le nombre de nœuds enfants est généralement égal au nombre de valeurs d'attributs possibles
  - L'attribut ne sera pas examiné plus d'une fois
  - Autre possibilité : division en deux sous-ensembles de valeurs
- Numérique :
  - Test visant à déterminer si la valeur est supérieure ou inférieure à une constante
  - L'attribut peut être examiné plusieurs fois
  - Autre possibilité : division en trois (ou division multiple)



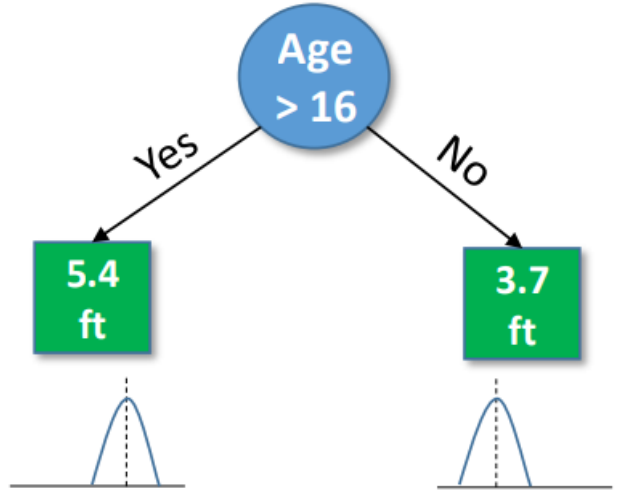


Arbres de décision : Par résultat

- Arbre de classification :
  - Résultat catégorique
    - par exemple, une personne peut-elle conduire ?
  - Les proportions des classes indiquent la probabilité de la classe
  - La prédiction correspond généralement à la majorité de la classe
  - Peut être "multi-classes".



- Arbre de régression :
  - Résultat continu
    - par exemple Taille (en ft)
  - Taille moyenne des individus dans un nœud foliaire donné



Construction d'arbres de décision

- Les arbres de décision peuvent être construits à la main
  - Représenter la logique de décision dans les branches d'un arbre
  - Présenter les connaissances connues (pour le raisonnement déductif)
- Généralement, les arbres de décision sont appris/entraînés (induction) :
  - Fournir des exemples : de nombreux ensembles de valeurs d'attributs et les classes/actions qui en résultent
  - L'algorithme construit ensuite un arbre à partir des exemples
  - Généralise à partir des exemples
  - L'induction ne garantit pas des arbres de décision corrects
  - L'apprentissage est non incrémental
  - Nécessité de stocker tous les exemples
  - ID3 est l'algorithme d'apprentissage de base
  - C4.5 est une version mise à jour et élargie

Induction d'un arbre

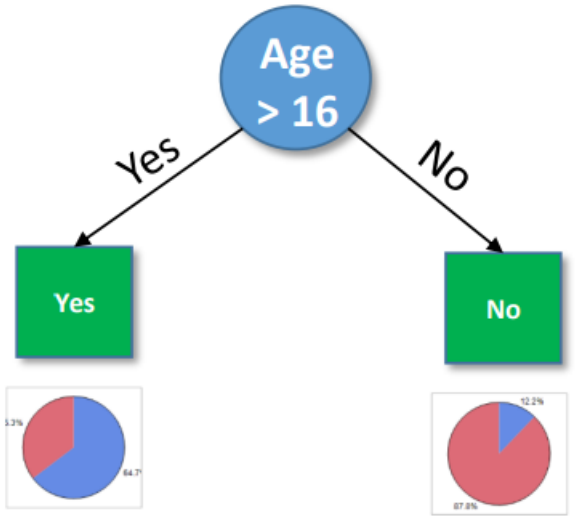
- Si X est vrai dans chaque exemple qui aboutit à l'action A, alors X doit toujours être vrai pour l'action A
  - Plus il y a d'exemples (dans le dataset), mieux c'est
  - Les erreurs dans les exemples causent des difficultés
    - Si X est vrai dans la plupart des exemples, X doit toujours être vrai
    - ID3 gère bien les erreurs (bruit) dans les exemples
  - Notez que l'induction peut entraîner des erreurs
    - Il peut s'agir d'une simple coïncidence si X est vrai dans tous les exemples
- L'apprentissage par arbre de décision typique tente de déterminer quels tests (c'est-à-dire les vérifications de la valeur des attributs) sont toujours vrais pour chaque action
  - On suppose que si ces choses sont vraies à nouveau, alors la même action devrait en résulter.

L'induction requiert des exemples

- D'où viennent les exemples ?
  - Le programmeur/concepteur fournit des exemples
  - Capturez les actions d'un joueur expert et l'état du jeu pendant qu'il joue
- Le nombre d'exemples nécessaires dépend de la difficulté du concept
  - Difficulté : Nombre de tests nécessaires pour déterminer l'action
  - Plus d’exemples, c'est toujours mieux
- Ensemble d'entraînement vs. Ensemble de test
  - Entraînement sur la plupart (par exemple 75%) des exemples
  - Utilisation du reste pour valider les arbres de décision appris en estimant l'efficacité de l'arbre sur des exemples qu'il n'a pas vus.

Apprendre les algorithmes

- Algorithmes récursifs
  - Trouver un test d'attribut qui sépare les actions
  - Diviser les exemples sur la base du test
  - Recourir à la récursivité (c'est-à-dire répéter la procédure) sur les sous-ensembles.



- Que signifie séparer ?
  - Idéalement, aucune action n'a d'exemples dans les deux ensembles
  - A défaut, la plupart des actions ont la plupart des exemples dans un ensemble
  - Ce qu'il faut mesurer, c'est l'entropie - le degré d'homogénéité (ou d'absence d'homogénéité) d'un ensemble
    - L'entropie est également importante pour la compression

Arbre de décision : Formation du modèle

- Commence par le nœud racine
- Recherche récursivement une variable qui divise au mieux les données en résultats. Algorithme de Hunt
- La "meilleure" variable est déterminée de manière heuristique
  - Indice de Gini (par ex. CART)
  - Gain d'information (par ex. ID3, C4.5)
  - Chi Square
  - Réduction de la variance (par ex. régression CART)
- Heuristique : produire des spits aussi homogènes (purs) que possible en termes d'étiquettes de résultats
- Critère d'arrêt : profondeur maximale, pureté des étiquettes dans chaque feuille.

Avantages de l'arbre de décision

- Représentation plus simple et plus compacte
- Facile à créer, à comprendre et à interpréter
  - Peut également être représentée sous forme de règles
- Stratégie de visualisation intuitive des connaissances et du raisonnement en tant que processus de prise de décision
- Reflète la prise de décision humaine plus fidèlement que d'autres algorithmes/représentations
- Les arbres de décision peuvent être appris ou "conçus".

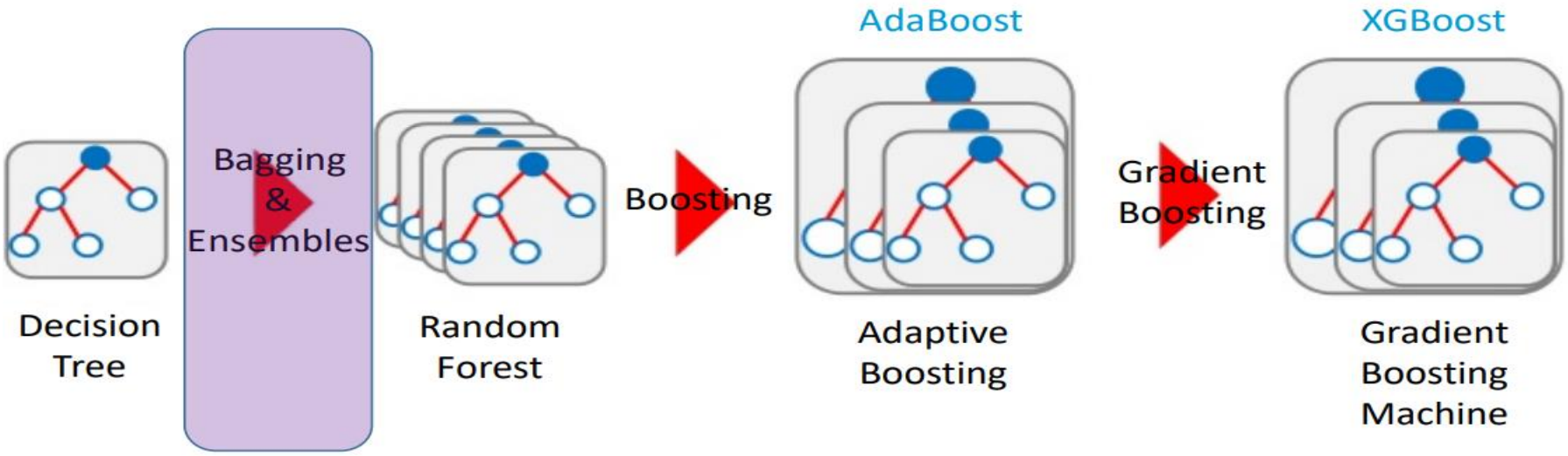
Inconvénients de l'arbre de décision

- Peut être difficile à concevoir plutôt qu'à former
- Très difficile à mettre à jour avec de nouvelles connaissances
  - Peut devoir construire un nouvel arbre à partir de zéro
- Peut ne pas convenir pour représenter de grandes bases de connaissances
- Apprentissage par induction :
  - Nécessite autant d'exemples que possible pour l'apprentissage
  - Les arbres de décision appris peuvent contenir des erreurs
  - Susceptible d'être surajusté (overfitting) lors de l'apprentissage
  - Seul un attribut à la fois est testé pour prendre une décision
  - Ne convient pas aux associations complexes (c-à-d les interactions)

Autres ML à base d'arbres

Dans la modélisation arborescente

- L'arbre de décision que nous venons d'étudier a donné lieu à un certain nombre d'autres stratégies d'apprentissage automatique basées sur des arbres plus avancés.
- Nous commencerons par l'idée des ensembles et du bagging et la manière dont ils ont conduit au développement des forêts aléatoires.
- Nous discuterons ensuite du boosting et de la manière dont il a été formé,
  - ce que l'on appelle le boosting adaptatif et le boosting de gradient, qui nous ont permis d'obtenir certains des apprentissages automatiques basés sur les arbres les plus sophistiqués à ce jour, y compris l'algorithme XGBoost.



Ensembles

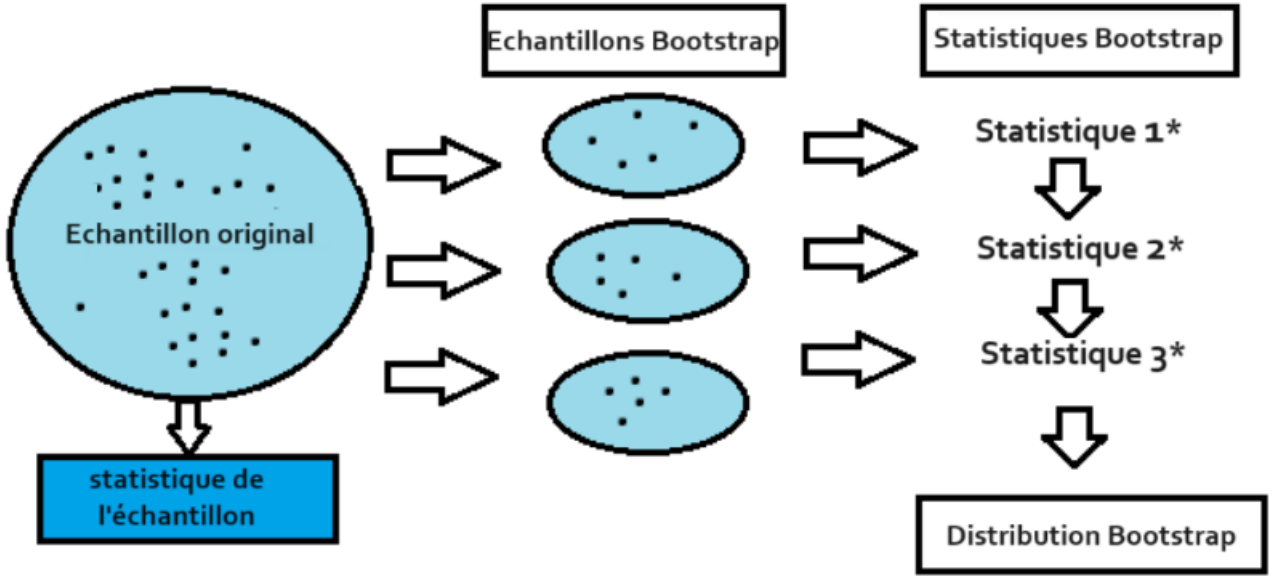
- Un ensemble est simplement une collection (c'est-à-dire un "groupe") de modèles entraînés sur la même tâche
- Plusieurs versions d'un même modèle ou différents types de modèles
- Résultat final : *moyenne pondérée ou vote*
- Un ensemble de différents modèles qui atteignent des performances de généralisation similaires *est souvent plus performant que n'importe quel modèle individuel*.
  - Comment ?

Ensembles - Pourquoi ça marche ?

- Supposons que nous disposions d'un ensemble de classificateurs binaires
- Chaque classificateur a la *même erreur moyenne* qui est meilleure qu'une supposition aléatoire
- Supposons que les erreurs qu'ils commettent sont indépendantes
- **Intuition** : la *majorité des classificateurs seront corrects sur de nombreux exemples où un classificateur individuel commet une erreur*
- Un simple vote majoritaire peut améliorer la performance de la classification en diminuant la variance dans ce contexte
- Comment former un tel ensemble ?

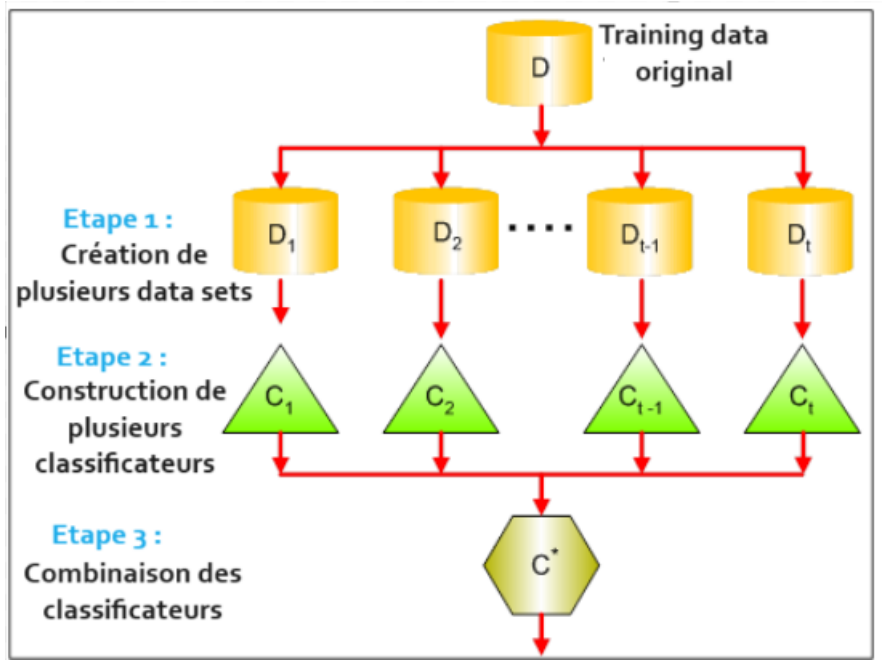
Précurseur : Bootstrapping

- Largement utilisé pour quantifier l'incertitude associée au modèle
  - Erreur standard
  - Intervalle de confiance pour le coefficient
- Obtenir de nombreux ensembles de données de la même taille que l'original par échantillonnage avec remplacement
- Obtenir des statistiques d'intérêt et examiner leur distribution pour estimer la variabilité



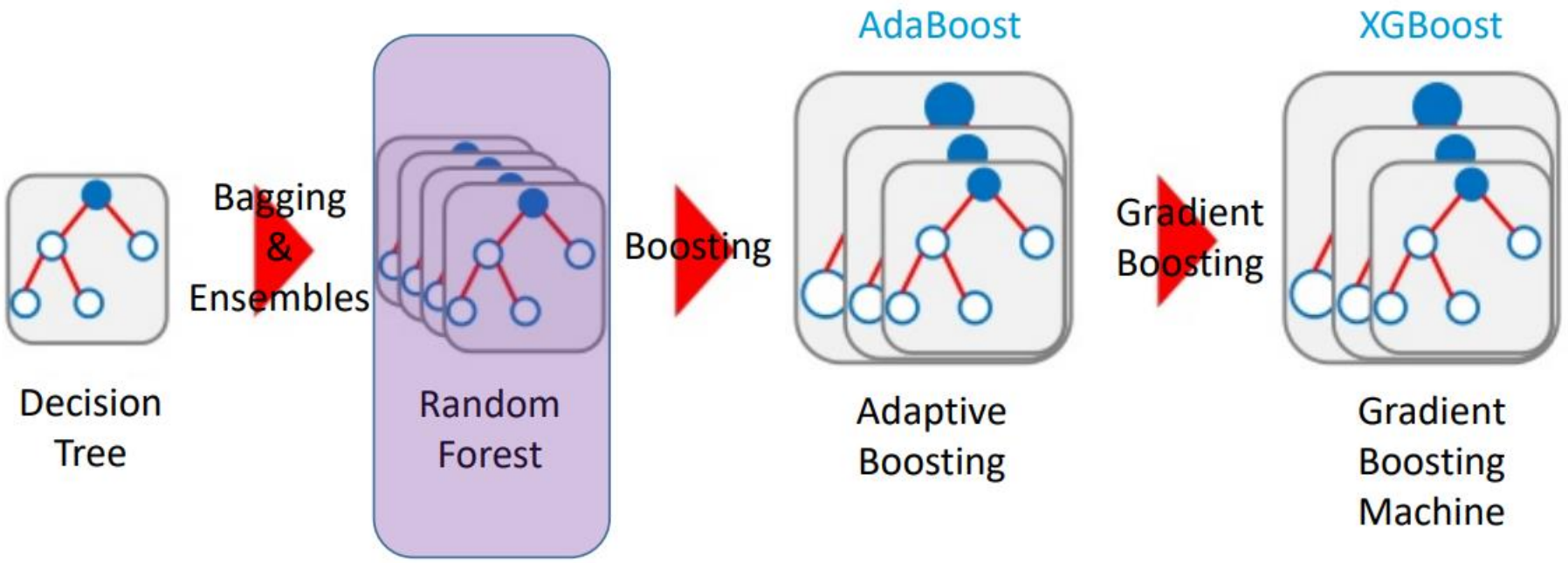
Bagging / Bagged Trees

- Aussi connu comme : Bootstrap aggregation
- Objectif : réduire la variance
- Tente de former des classificateurs indépendants en échantillonnant l'ensemble d'apprentissage
- Échantillonne *k* fois avec remplacement
- Forme (train) *k* classificateurs sur des sous-ensembles
- Utile pour les modèles à haute variance et à haute capacité (c'est-à-dire les arbres de décision)
- Un plus grand nombre de modèles est toujours préférable (compromis en termes de temps d'exécution)
- Estimation de l'erreur interne : utilise la partie des données qui n'a pas été construite pour créer le modèle comme ensemble de test (données hors du sac).



Suite sur la modélisation arborescente (Random Forest)

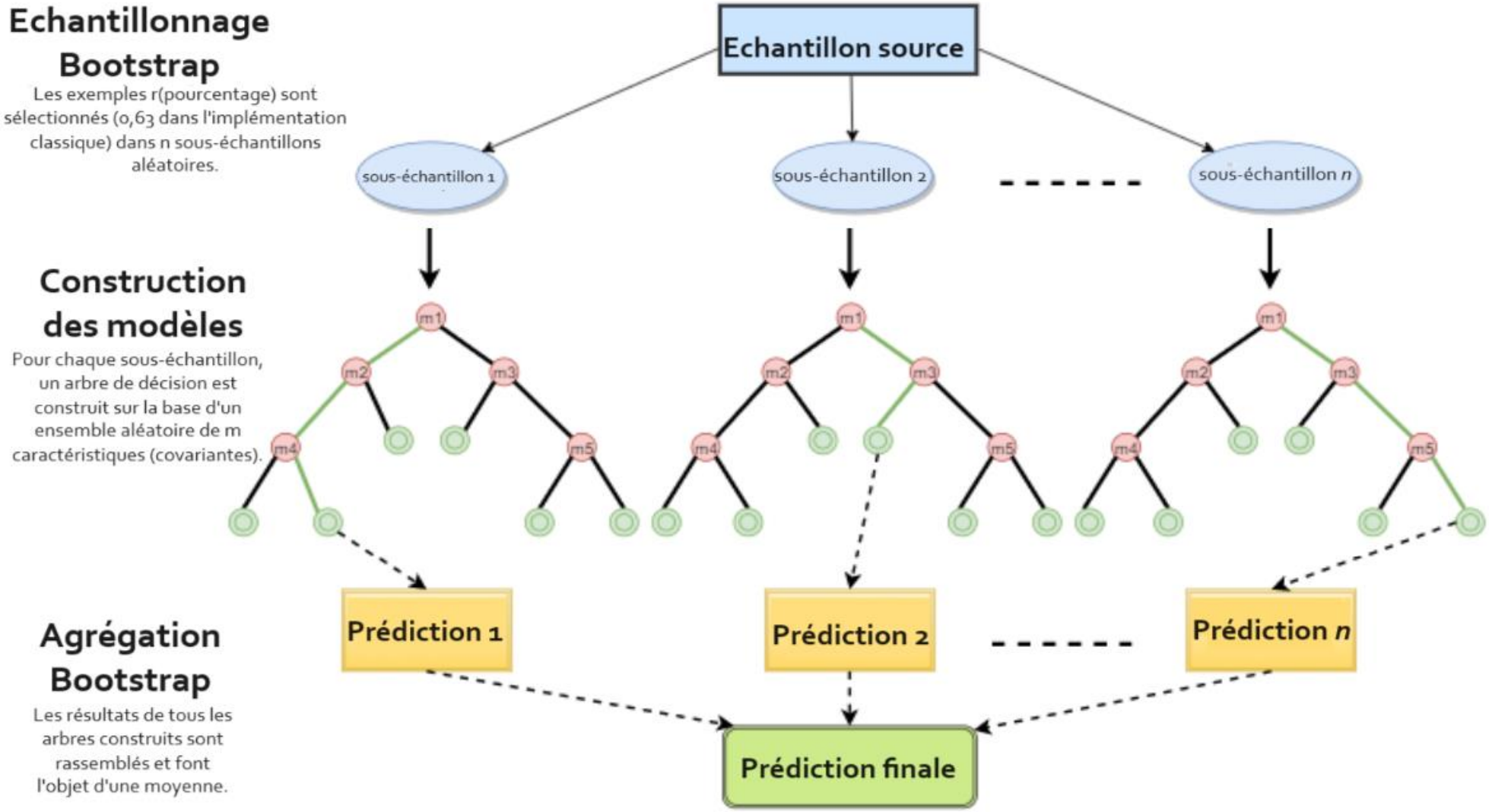
- On vient de voir ce que le bagging et la méthode Ensemble
- On peut alors introduire l'algorithme de Random Forest



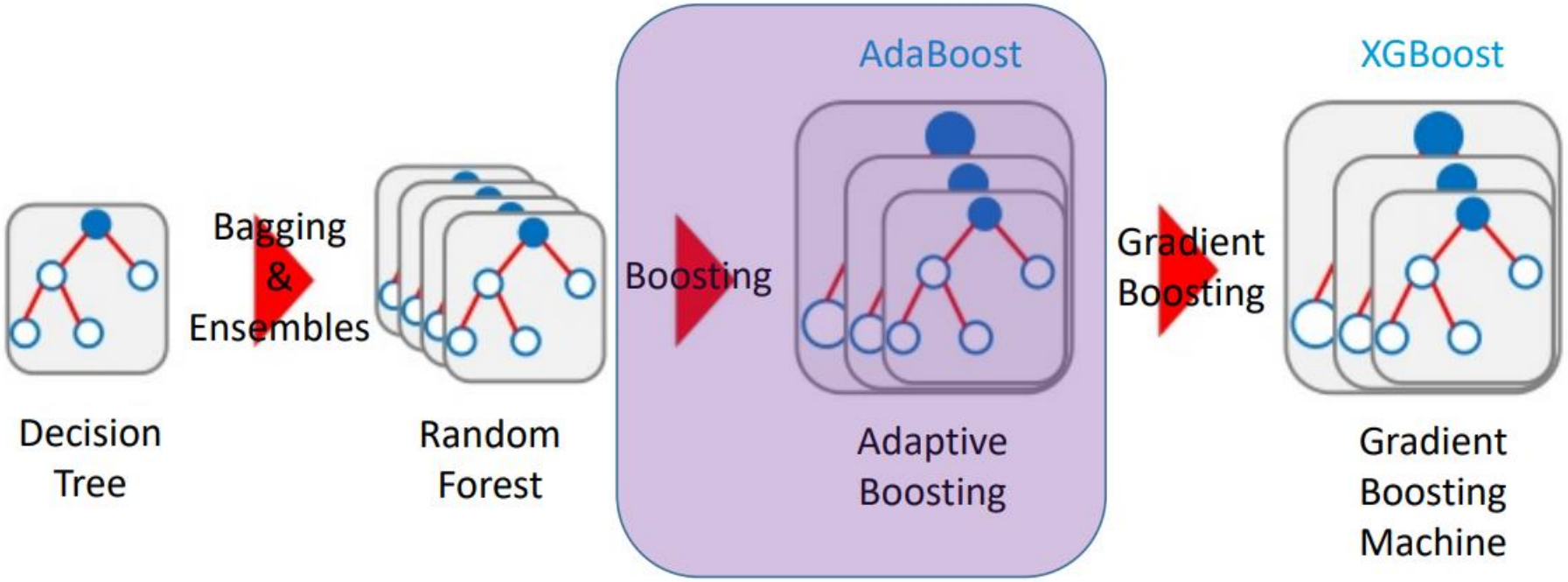
Random Forests

- Les forêts aléatoires sont une extension réussie des bagged trees
- Considérées comme la panacée pour tous les problèmes de ML.
  - Algorithme de référence
  - Fonctionne généralement assez bien pour la plupart des problèmes.
- Extension de la méthode de bagging: Prend en compte un sous-ensemble aléatoire de caractéristiques lorsqu'il s'agit de décider des variables à diviser.
- Lorsque de nouvelles données sont disponibles, elles sont transmises à tous les arbres de la forêt et la classe est estimée sur la base du résultat le plus populaire.
- Importance de la caractéristique : estimation de la contribution d'une seule variable à la classification.

Random Forest - Illustration

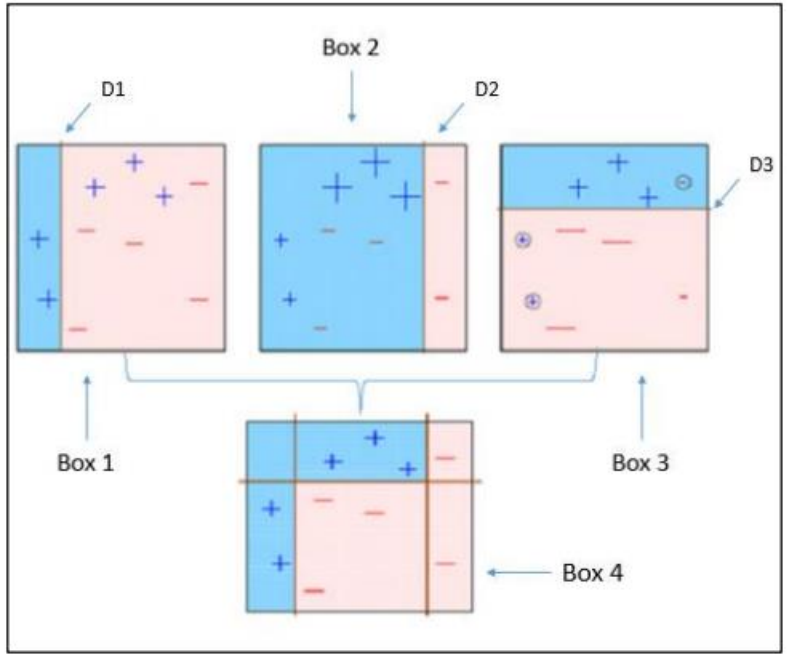


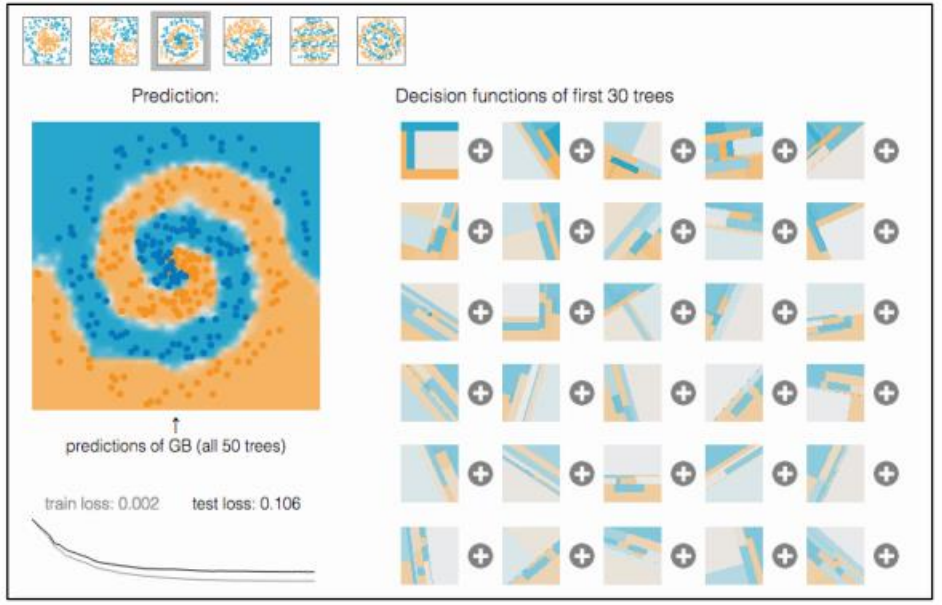
Suite sur la modélisation arborescente (Boosting)



Boosting

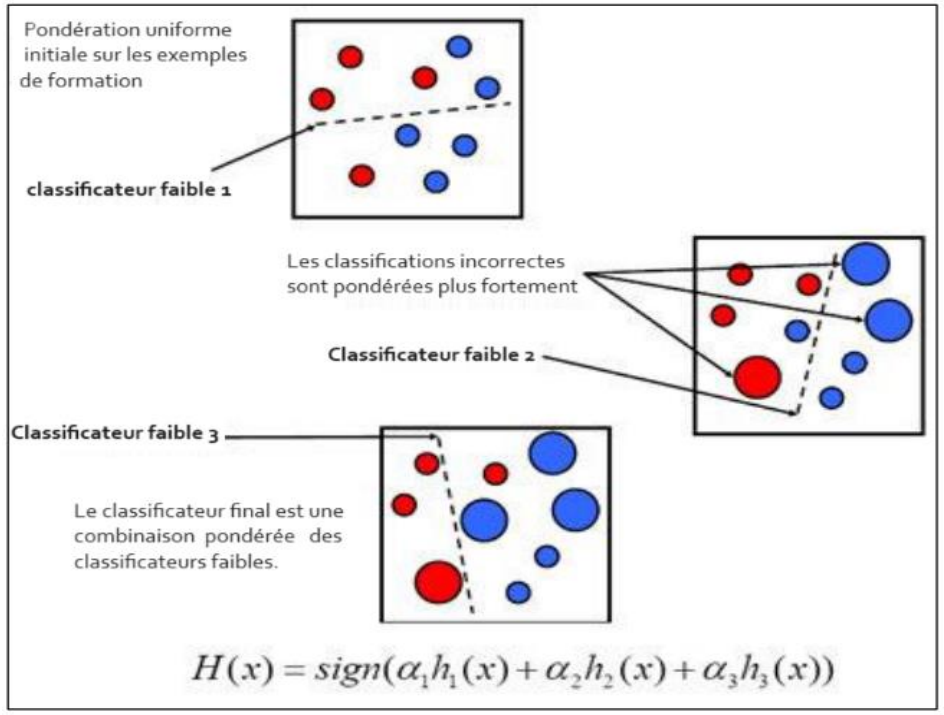
- Méta-algorithme d'ensemble pour *combiner des apprenants faibles en un apprenant fort*.
- Apprentissage itératif des apprenants faibles (par exemple, arbre de décision à un niveau), de manière séquentielle
- À chaque itération, les poids des données sont mis à jour pour se concentrer davantage sur les instances mal classées
- Facilement mis en échec par les données bruitées et les valeurs aberrantes.
- Sensible aux problèmes d'overfitting

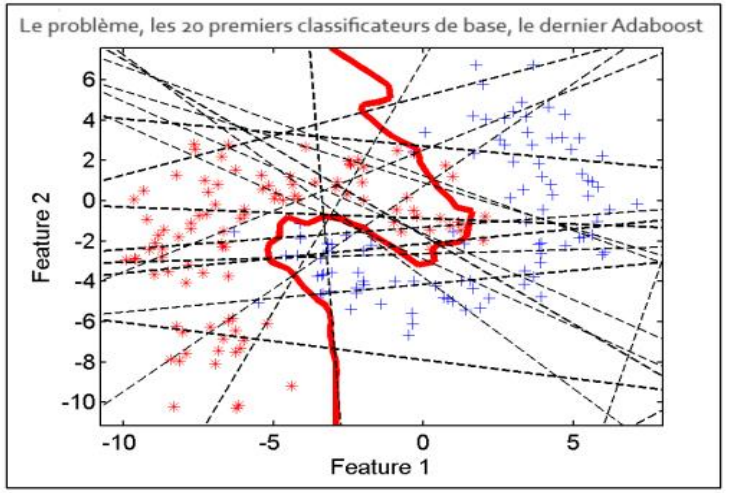




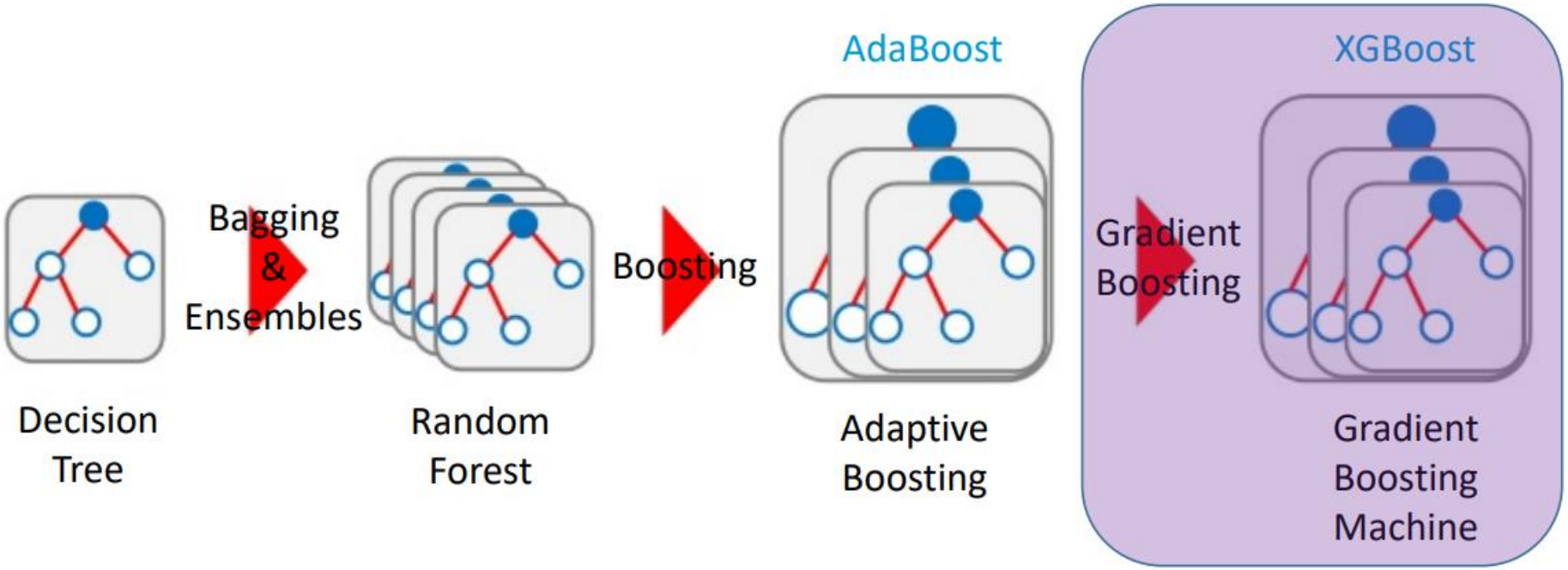
ADABoost – Adaptive Boosting

- L'une des versions les plus connues d'un algorithme basé sur un arbre de boosting est ADABoost.
- Dans l'image à droite, nous pouvons voir comment un groupe d'apprenants faibles est combiné dans un modèle de prédiction très complexe.



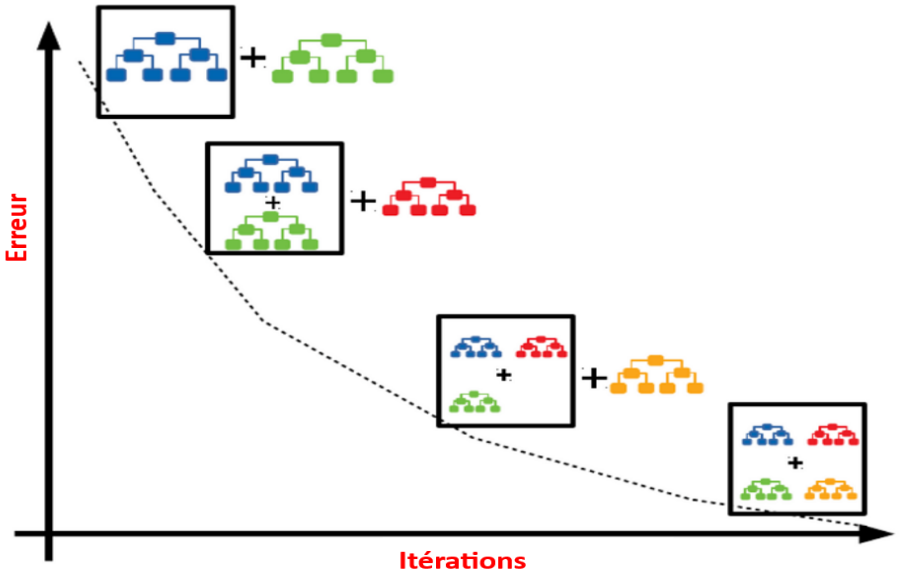


Suite sur la modélisation arborescente (Gradient Boosting)



GBM - Gradient Boosting Machine

- Exemples:
  - MART (Multiple Additive Regression trees)
  - GBRT (Gradient Boosted Regression Trees)
  - XGBoost



- Au lieu de s'entraîner sur des instances pondérées, l'apprenant faible s'entraîne sur les erreurs restantes (appelées pseudo-résidus) de l'apprenant fort et méta.
- Une autre façon de donner de l'importance aux instances difficiles
- La contribution de l'apprenant faible à l'apprenant fort n'est pas calculée en fonction de sa performance sur le nouvel échantillon de distribution, mais en utilisant un processus d'optimisation par *descente de gradient*
- La contribution calculée est celle qui minimise l'erreur globale de l'apprenant fort.

XGBoost – Extreme Gradient Boosting

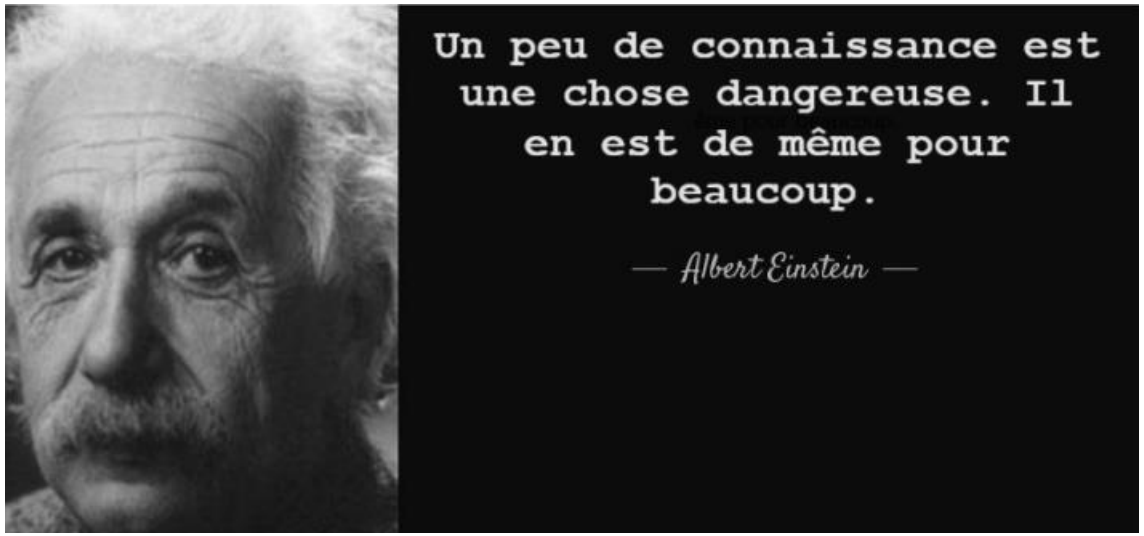
- L'une des méthodes les plus connues et les plus efficaces de ces arbres de renforcement du gradient est appelée xgboost.

- Elle utilise une méthode de renforcement du gradient régularisée (pour éviter l'overfitting) et le calcul parallèle (10 fois plus rapide que le renforcement du gradient) pour la rendre plus rapide et plus efficace.

- Il existe également une autre variante appelée LGBoost qui est devenue très populaire ces dernières années.
- Dans l'ensemble, cette section du chapitre a pour but de vous présenter certaines des connexions entre les différents algorithmes d'apprentissage automatique basés sur les arbres.

Synthèse

- Web sémantique
  - Terminologie et normalisation/connexion sémantique
  - Liaison entre bases de données, ontologies, données, recherche sur le web
  - HTML, XML - RDF, OWL
  - Logiques de description
- Arbres
  - Représentation des connaissances d'experts pour l'induction via l'apprentissage automatique
  - Terminologie, structure, liens avec la logique
  - Construction d'arbres de décision / induction
- Autres ML basés sur les arbres
  - Ensembles
  - Bagging
    - Random Forrests
  - Boosting
    - GBM
    - XGBoost


Un peu de connaissance est une chose dangereuse. Il en est de même pour beaucoup.
— Albert Einstein —

Travail de mi-chemin: Rappel

- **Objectifs :**

1. Entreprendre une recherche documentaire (revue de littérature).
2. Identifier une poignée (par exemple 3-5) d'articles de recherche originaux qui illustrent des exemples des sujets d'IA couverts dans ce cours appliqués au domaine d'expertise tel que la médecine, le droit… (même ceux qui n'ont pas encore été abordés). Ces articles devraient idéalement présenter toutes (ou certaines) les propriétés suivantes (a) clairement rédigés, (b) ayant un impact, par exemple très cités, (c) des figures intuitives, (d) une revue claire de la recherche pertinente/reliée, et (e) une bonne compréhension de ce qui se passe dans le domaine de système expert en médecine ou d'autres domaine tel que le droit. (e) indiquer clairement les avantages et les inconvénients du travail proposé.
3. Préparez un bref synopsis/évaluation (1 paragraphe) de chaque publication ainsi que les informations de citation pour chaque article et sauvegardez chaque article au format PDF. Sauvegardez vos 3 à 5 synopsis au format PDF en classant les articles par ordre décroissant d'intérêt pour la présentation de l'article en question. Placez votre synopsis et les PDF des articles dans un dossier zippé.
4. Remettre le dossier zippé ci-dessus dans les délais impartis. A partir de là, j'identifierai un document unique pour chaque équipe/individu à présenter.
5. Préparer une présentation orale au format journal (diapositives Powerpoint facultatives). Les présentations en classe dureront 10 minutes chacune et 5 minutes de discussion. (Le temps exact dépend de la taille de la classe)

Exemples de sujets des TP (non limités à ceux-ci) :

- Systèmes experts, outils d'aide à la décision clinique ou juridique, ontologies, harmonisation des données, l'analyse collaborative des données, le raisonnement probabiliste, l'infrastructure de l'information sanitaire, l'éducation sanitaire, les systèmes de surveillance des patients, la recherche d'informations et les bibliothèques numériques.