



## REKLAOUI KAMAL

- **Directeur**  
École Nationale des Sciences Appliquées de Tétouan
- **Responsable & créateur de la filière d'Ingénieur**  
Intelligence Artificielle et Sciences des Données
- **Chef de Projet de Recherche – Al Khawarazmi**  
Application de l'Intelligence Artificielle  
dans l'industrie : *Maintenance prédictive*
- **Responsable Scientifique**  
Projet ERASMUS+ MarMOOC
- **Titulaire de Deux Brevets**  
Application de l'IA à l'apprentissage en ligne (*E-learning*)  
Maintenance prédictive basée sur l'IA



# Vue d'ensemble du système

Une plateforme d'e-learning intelligente & augmentée qui personnalise en temps réel le parcours pédagogique de chaque apprenant. :

- **Un moteur de génération de contenu dynamique (RAG)** : Il recherche dans une base documentaire des ressources adaptées, puis utilise un modèle de langage local (LLM) pour reformuler et personnaliser les contenus.
- **Des algorithmes d'apprentissage adaptatif** : Des méthodes comme le Bayesian Knowledge Tracing (BKT), l'Item Response Theory (IRT) et le Reinforcement Learning (RL) analysent en continu les performances et ajustent la difficulté ainsi que les recommandations pédagogiques.
- **Une architecture sécurisée et distribuée** : Pour garantir la rapidité, la scalabilité et la confidentialité des données (via des conteneurs Docker, des caches type Redis/Memcached et une orchestration via Kubernetes).



L'objectif est simple : **adapter le parcours d'apprentissage à chaque élève**, selon ses réponses, son niveau, sa progression et ses besoins.

# Fonctionnement global de la plateforme

## 1. Capture des interactions

- Lorsqu'un utilisateur se connecte à Moodle, le plugin intercepte ses actions et enregistre chaque événement (par exemple, le lancement d'un quiz ou la soumission d'un devoir).

## 2. Analyse en temps réel

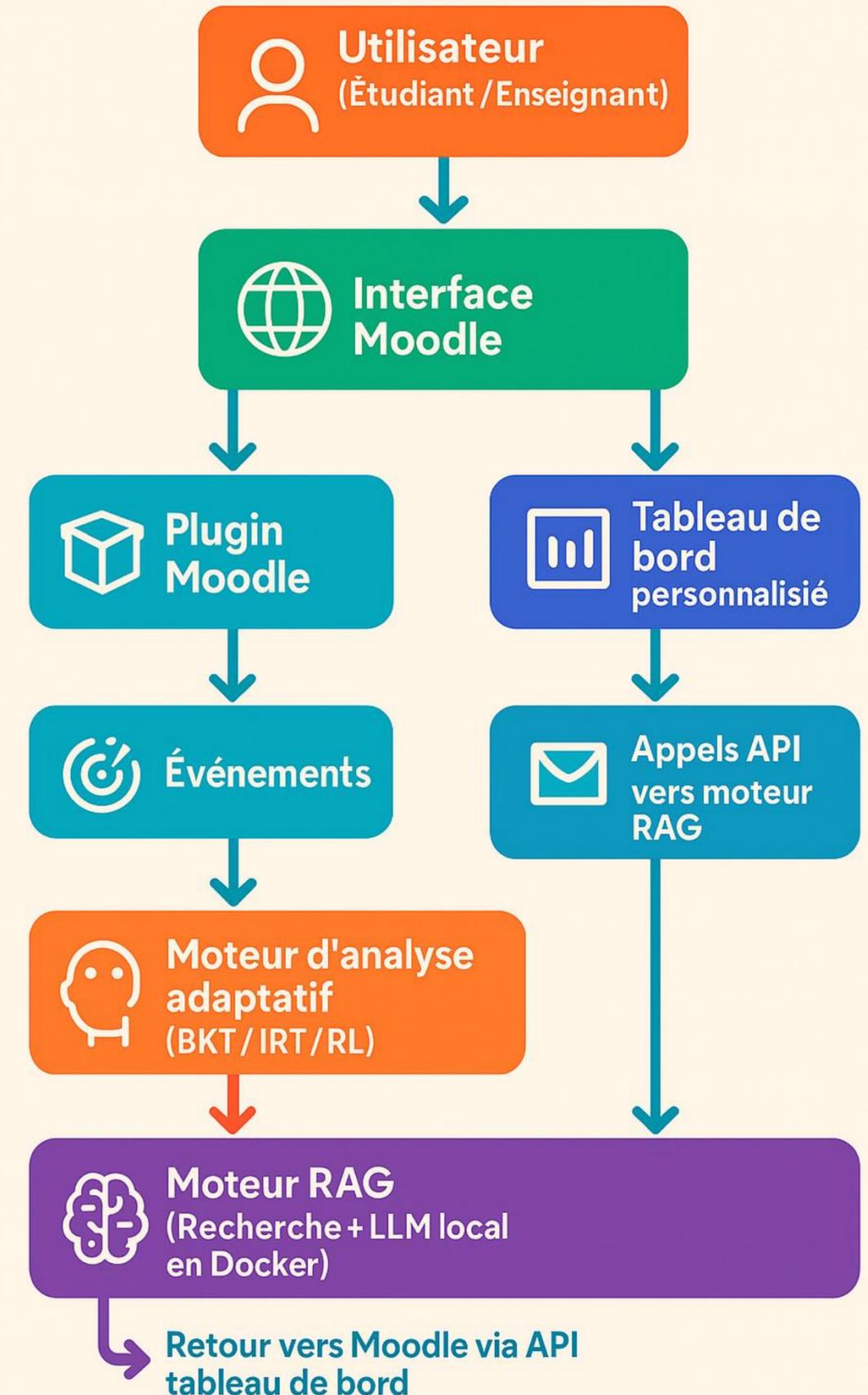
- Les données collectées sont envoyées vers le moteur d'analyse qui utilise les algorithmes d'apprentissage adaptatif (BKT, IRT, RL) pour évaluer les performances de l'apprenant.
- Ces algorithmes ajustent ensuite la difficulté des contenus et préparent des recommandations personnalisées.

## 1. Génération de contenu dynamique

- Si une difficulté est détectée ou si l'apprenant a besoin d'explications supplémentaires, le moteur RAG interroge la base documentaire pour extraire des ressources pertinentes.
- Un LLM local (déployé via conteneurs Docker) reformule ces extraits pour produire une réponse pédagogique adaptée.

## 2. Retour à l'utilisateur

- Les contenus personnalisés, les notifications et les recommandations sont ensuite affichés directement sur l'interface de Moodle (tableaux de bord interactifs, notifications, etc.).



# Intégration de Moodle avec les composants

Moodle sert de point d'entrée pour les apprenants. La plateforme s'intègre à Moodle grâce à un **plugin** spécialement développé qui permet de capter les interactions en temps réel. Voici comment cela fonctionne :

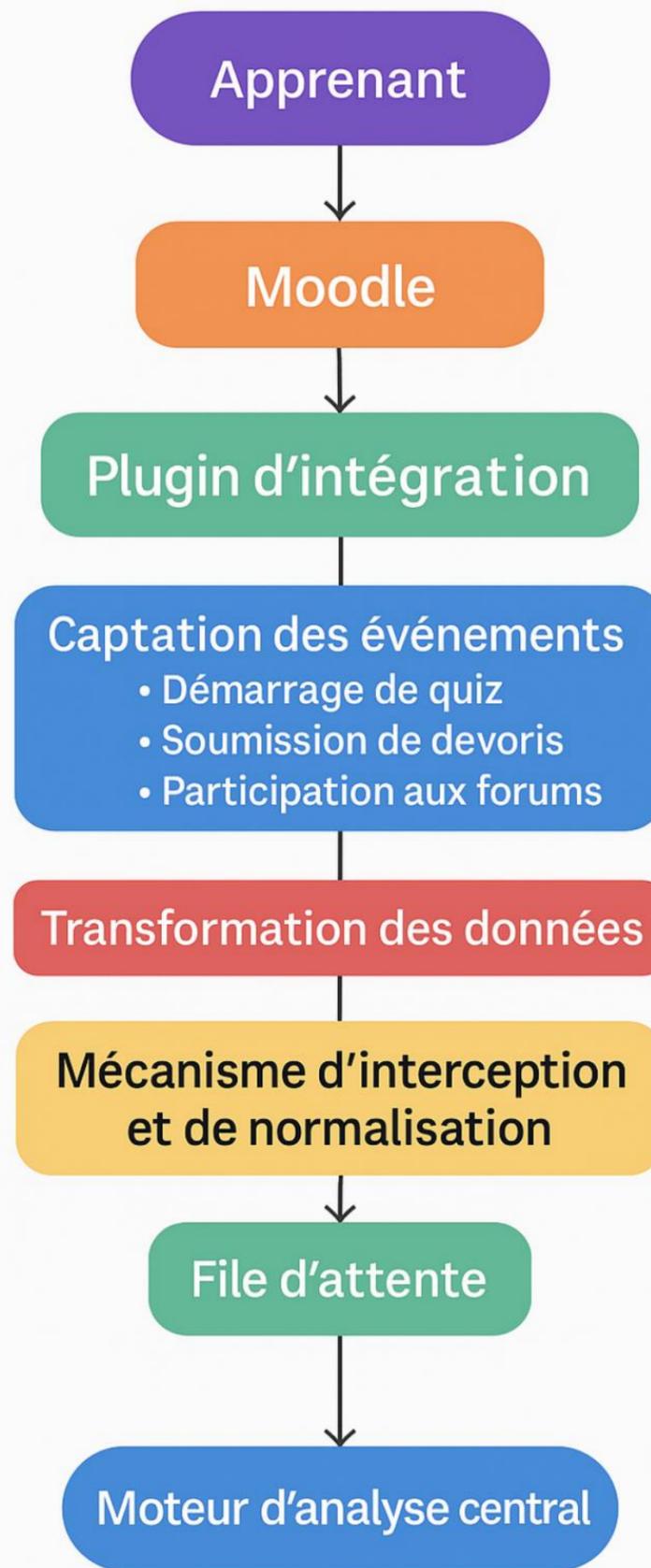
## . Plugin d'intégration dans Moodle

- **Captation des événements** : Le plugin exploite les API et hooks de Moodle pour intercepter des actions clés (démarrage de quiz, soumission de devoirs, participation aux forums, etc.).
- **Transformation des données** : Chaque événement est converti en un objet structuré (par exemple, un objet JSON) contenant les informations essentielles (ID utilisateur, type d'action, timestamp, etc.).
- **Transmission sécurisée** : Les données sont envoyées via des protocoles sécurisés (API RESTful avec TLS/SSL) à un moteur d'analyse central.

## . Mécanisme d'interception et de normalisation

- Un module dédié surveille en continu les activités des apprenants, formate ces informations et les place dans une file d'attente (en utilisant RabbitMQ) pour un traitement asynchrone.

Cette intégration permet à la plateforme d'obtenir en temps réel des informations précieuses sur le comportement et les performances de l'apprenant, lesquelles serviront à adapter dynamiquement les contenus pédagogiques.



# Apprentissage Adaptatif en E-learning

**L'IRT et le BKT** (Bayesian Knowledge Tracing (BKT), l'Item Response Theory (IRT) ),fournissent une base solide pour évaluer et suivre les compétences et la progression des apprenants.

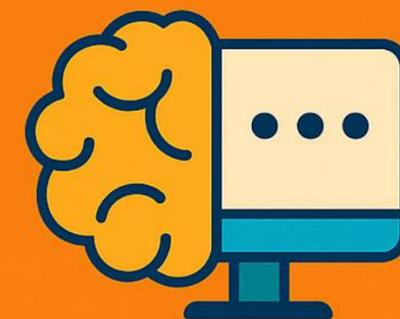
**L'apprentissage par renforcement RL** va optimiser les parcours d'apprentissage en ajustant dynamiquement les recommandations et les activités en fonction des retours et des performances de l'apprenant.

Créer un environnement d'apprentissage hautement personnalisés et efficaces, où chaque apprenant reçoit un parcours unique et optimisé pour ses besoins, maximisant ainsi l'engagement et la rétention des connaissances.

## APPRENTISSAGE ADAPTATIF EN E-LEARNING



**BKT**  
Estime la maîtrise



Recommandations  
via deep learning



**IRT**  
Ajuste difficulté  
des exercices



**RL**  
Optimise les  
parcours

# Item Response Theory (IRT)

La Théorie de la Réponse aux Items (IRT) a été développée principalement dans les années 1950 et 1960 par des chercheurs comme Frederic M. Lord, Georg Rasch et Paul Lazarsfeld

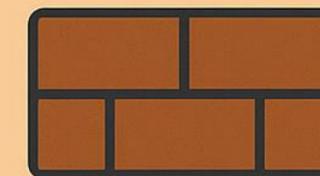
Élément	Métaphore
Élève	Joueur avec un niveau de saut 
Question	Obstacle plus ou moins haut 
Difficulté	Hauteur de l'obstacle 
Compétence ( $\theta$ )	Force du saut du joueur  
Réussite probabilité	Estimée selon $\theta$ et les paramètres 

L'IRT (Théorie de la Réponse aux Items) modélise la probabilité qu'une personne réponde correctement à une question en fonction de son niveau de compétence (appelé "trait latent") et des caractéristiques de la question. Chaque question a des paramètres la difficulté, le pouvoir de discrimination (à quel point elle distingue les niveaux de compétence) et parfois une chance de deviner correctement.

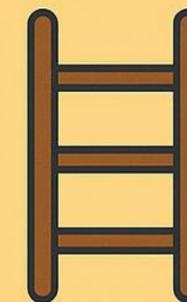
En gros, plus une personne a un niveau de compétence élevé, plus elle a de chances de répondre correctement aux questions. **L'IRT permet ainsi de mieux comprendre la performance individuelle et d'adapter les tests aux niveaux des participants pour obtenir des mesures plus précises et équitables.**



**ÉLÈVE**  
 Joueur avec  
 un niveau de  
 saut



**QUESTION**  
 Obstacle  
 dans un jeu



**DIFFICULTÉ**  
 Hauteur de  
 l'obstacle



**COMPÉTENCE**  
 Capacité de  
 saut du joueur



**RÉUSSITE**  
 Probabilité  
 de franchir l'obstacle



# Item Response Theory (IRT)

## 1. Chaque question a 3 paramètres (3PL Model) :

Paramètre	Signification
<b>Difficulté (b)</b>	Plus elle est élevée, plus la question est dure
<b>Discrimination (a)</b>	Capacité de la question à différencier les bons et les moins bons élèves
<b>Guessing (c)</b>	Chance de réussir en devinant

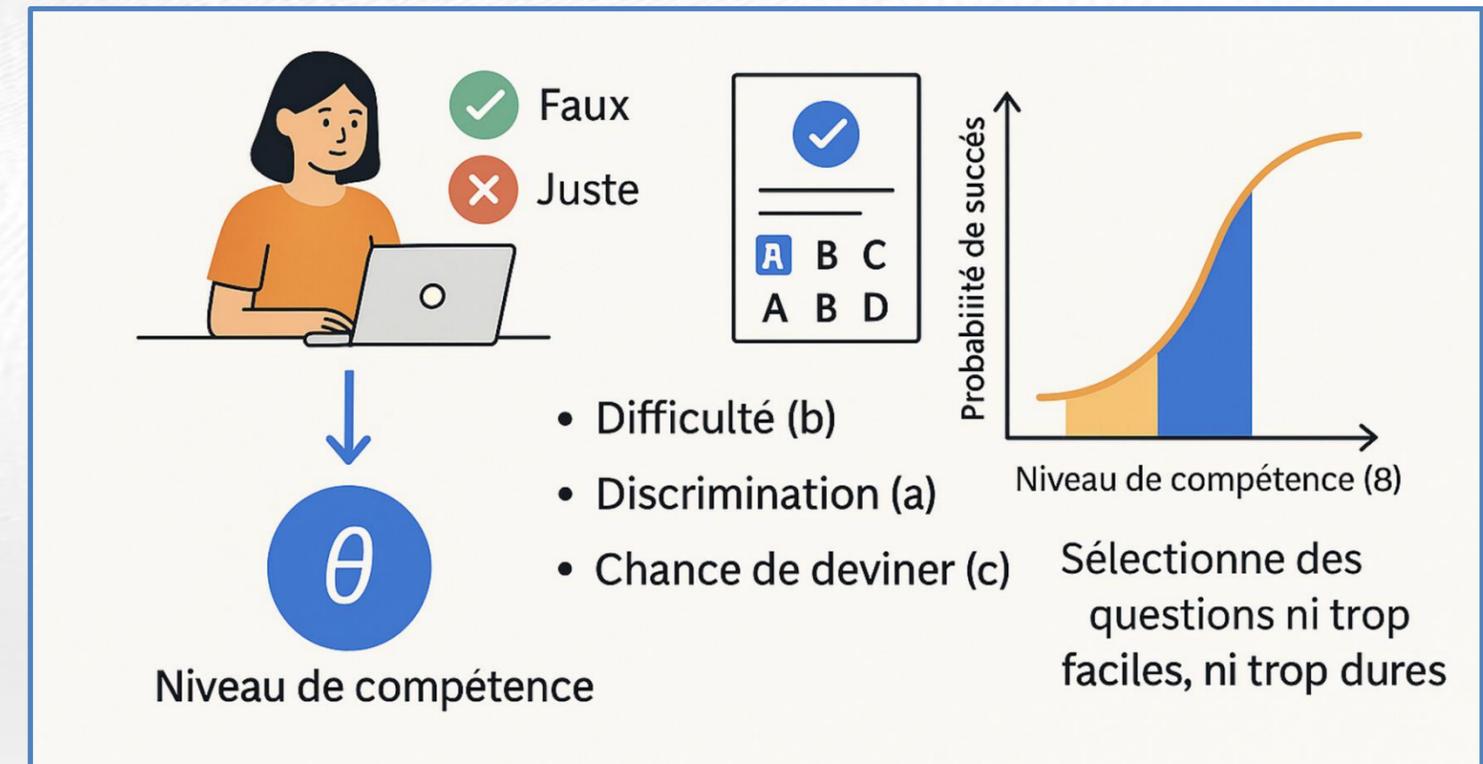
Exemple : Une QCM facile avec 4 choix → 25% de chance de deviner ( $c = 0.25$ )

### Exemple

Imaginons cette question :

- Difficulté = 0.5
- Discrimination = 1
- Guessing = 0.2
- Un élève avec  $\theta = -1$  a très peu de chances de répondre juste
- Un élève avec  $\theta = 1$  a beaucoup plus de chances !

La courbe IRT ressemble à un S : En bas = faibles chances de réussir  
 En haut = fortes chances



## 2. Chaque élève a un niveau de compétence ( $\theta$ )

- $\theta$  peut aller de -3 (très faible) à +3 (très fort)
- Plus  $\theta$  est élevé, plus l'élève a de chances de réussir des questions difficiles

# Item Bayesian Knowledge Tracing (BKT)

Le **Bayesian Knowledge Tracing (BKT) (1994)** est un modèle probabiliste qui suit l'apprentissage d'un apprenant au fil du temps. Il estime la probabilité qu'un apprenant maîtrise une compétence donnée après chaque réponse qu'il fournit à une question.

**BKT prend en compte plusieurs éléments** : la probabilité initiale que l'apprenant connaisse déjà la compétence, la probabilité d'apprendre cette compétence après chaque tentative, et la probabilité de faire des erreurs ou de deviner correctement.

Avec ces informations, le BKT met à jour en continu la probabilité que l'apprenant maîtrise la compétence, ce qui permet de personnaliser les parcours d'apprentissage en proposant des activités ou des exercices adaptés au niveau réel de l'apprenant.

Paramètre	Description simplifiée
$P(L_0)$	Probabilité de départ qu'il ait compris
$P(T)$	Chance qu'il apprenne après un essai
$P(G)$	Chance de réussir par hasard (guess)
$P(S)$	Chance d'échouer malgré la maîtrise (slip)

## Bayesian Knowledge Tracing:

Suivre l'apprentissage comme un détective du savoir



### 1. Observation de l'élève

### 2. Chaque réponse donne un indice



### 3. Quatre probabilités clés utilisées



$P(L_0)$



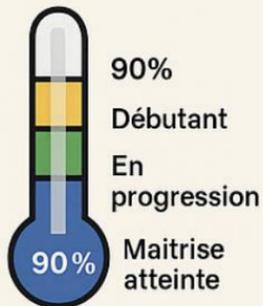
$P(T)$



$P(G)$

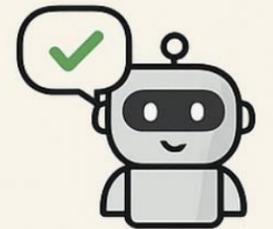


$P(S)$



### 4. Thermomètre du savoir

20% Débutant  
60% En progression  
90% Maîtrise atteinte



### Résumé visuel

- 👁️ Observer les réponses
- 🔄 Utiliser des probabilités
- 📊 Mettre à jour les croyances
- 🎯 Adapter l'apprentissage

# Item Bayesian Knowledge Tracing (BKT)

L'élève apprend un concept (ex. : la division). On ne sait pas s'il l'a compris. Le système observe ses réponses.

## 2. Chaque réponse donne un indice

Visuel : Deux chemins :

-  Bonne réponse → "Peut-être qu'il a compris"
-  Mauvaise réponse → "Il ne semble pas encore avoir compris"

Le modèle analyse chaque réponse et ajuste sa "croyance".

## 4 probabilités pour estimer la maîtrise

Paramètre	Description simplifiée
$P(L_0)$	Probabilité de départ qu'il ait compris
$P(T)$	Chance qu'il apprenne après un essai
$P(G)$	Chance de réussir par hasard (guess)
$P(S)$	Chance d'échouer malgré la maîtrise (slip)



- Prend en compte les réponses successives (juste ou faux)
- Met à jour une probabilité de maîtrise

**$P(L_0)$**  : probabilité initiale de savoir  
 **$P(T)$**  : probabilité d'apprendre  
 **$P(G)$**  : probabilité de deviner juste  
 **$P(S)$**  : probabilité de se tromper malgré la maîtrise

# Reinforcement Learning (RL)

L'apprentissage par renforcement est un type d'apprentissage automatique dans lequel un **agent** apprend à **prendre des décisions** en interagissant avec un **environnement**.

L'agent reçoit :

- **des récompenses positives** s'il fait une bonne action,
- **des pénalités** s'il fait une mauvaise action.

👉 Il apprend progressivement à **maximiser sa récompense** globale en ajustant ses actions selon les retours obtenus.

C'est le même principe que **dresser un animal** ou **jouer à un jeu vidéo**.

## Robot dans un labyrinthe

Le robot doit sortir d'un labyrinthe.

Il peut aller à **gauche**, à **droite**, **en avant**, **en arrière**.

À chaque mouvement :

-  S'il se rapproche de la sortie → il reçoit +1 point (récompense)
-  S'il touche un mur → -1 point (punition)
-  S'il sort → +10 points ! 🎉

➡ Le robot va **tester différents chemins**, mémoriser les meilleurs, et éviter les erreurs.

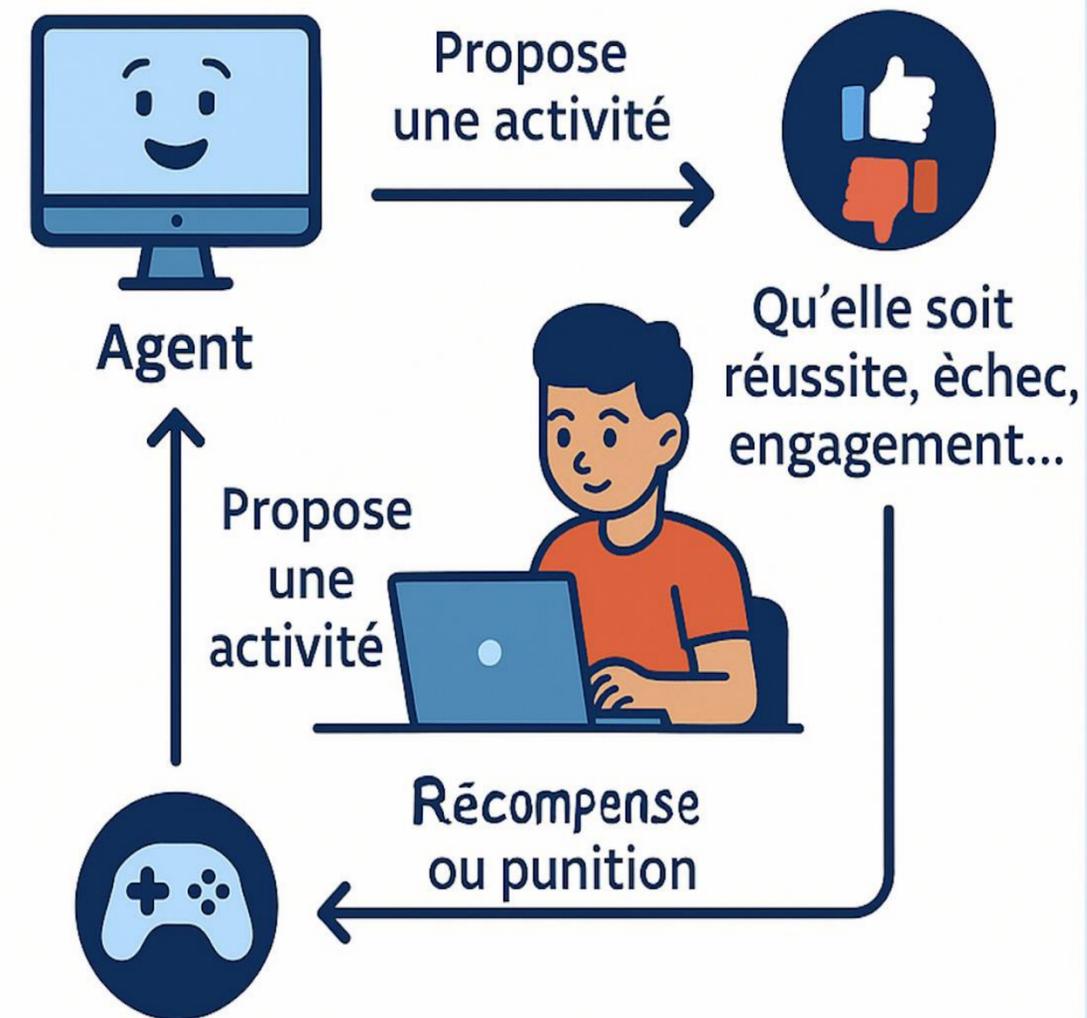
📊 3 éléments clés du Reinforcement Learning :

Élément	Rôle
<b>Agent</b>	Celui qui prend les décisions (robot, IA, joueur)
<b>Environnement</b>	Le monde dans lequel il agit (jeu, simulation...)
<b>Politique</b>	Stratégie qu'il apprend pour choisir les bonnes actions

# RL : Optimise les parcours

**Objectif:** Optimiser le parcours d'apprentissage de chaque élève

## Fonctionnement simplifié

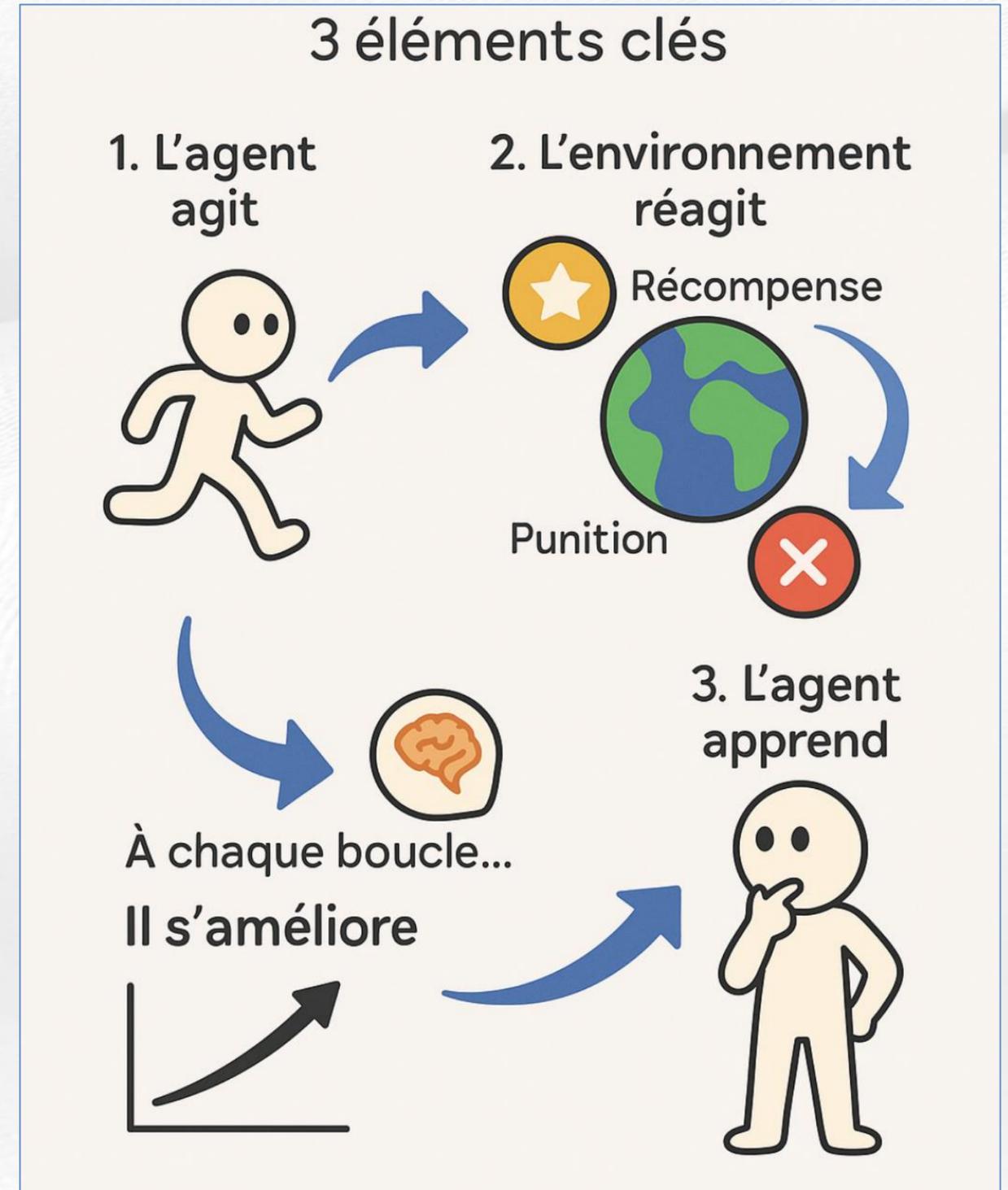


Comme un jeu vidéo où l'IA apprend quel est le meilleur chemin pour progresser

# Reinforcement Learning (RL)

## Origines historiques :

- **1950s** : Fondation du concept – **Richard Bellman** introduit la notion de **programmation dynamique** et de **valeur d'utilité**.
- **1960s–70s** : Premiers travaux mathématiques en théorie du contrôle.
- **1989** : **Chris Watkins** introduit **Q-learning**, un algorithme fondamental de RL.
- **1990s–2000s** : Développement du RL statistique, lien avec le machine learning.
- **2015** : Explosion médiatique avec **DeepMind (Google)** qui utilise le **Deep Reinforcement Learning** pour battre des humains à des jeux comme **Atari** ou **Go**.



Jeux vidéo (ex : AlphaGo), Robots autonomes

, Systèmes de recommandation, Finance et trading algorithmique, Optimisation industrielle, Éducation adaptative.

# LLM (Large Language Model)

Un **LLM** (Large Language Model), ou **grand modèle de langage**, est un type de modèle d'intelligence artificielle conçu pour comprendre et générer du texte en langage naturel.



Élément	Image mentale
<b>LLM</b>	 <i>Un perroquet super intelligent qui a lu toute la bibliothèque</i>
<b>Apprentissage</b>	 <i>Il lit beaucoup et devine les mots manquants dans les phrases</i>
<b>Réponse</b>	 <i>Il complète ce que tu écris, mot après mot, comme un jeu</i>
<b>Mots → Maths</b>	 <i>Il transforme chaque mot en nombre pour faire des calculs</i>
<b>Attention</b>	 <i>Il surligne les mots importants pour mieux répondre</i>

## Images mentales d'un LLM



**LLM**

Un perroquet super intelligent qui a lu toute la bibliothèque



**Apprentissage**

Il lit beaucoup et devine les mots manquants dans les phrases



**Réponse**

Il complète ce que tu écris, mot après mot, comme un jeu



**Mots → Maths**

Il transforme chaque mot en nombre pour faire des calculs

**3**

### Il transforme les mots en nombres

Les mots sont convertis en des chiffres appelés tokens.

Exemples:

chat

2116

mange

3872



Vecteur

[0,23, -1,55, 0,89...]



Le modèle analyse ces vecteurs pour «comprendre» le texte.

# Les limites des LLMs (Large Language Model)

## 1. Pas de compréhension réelle

→ Ils manipulent les mots sans en comprendre le sens comme un humain.

## 2. Dépendent des données d'entraînement

→ Ils peuvent reproduire des biais, erreurs ou stéréotypes présents dans les textes appris.

## 3. Inventent parfois des réponses (hallucinations)

→ Ils peuvent donner des infos fausses avec beaucoup de confiance.

## 4. Pas de mémoire à long terme (par défaut)

→ Ils ne se rappellent pas des échanges précédents, sauf si on les relie à un historique.

## 5. Pas de jugement moral ou éthique propre

→ Ils ne savent pas ce qui est bien ou mal sans qu'on leur impose des règles externes.

## 6. Sensibles aux formulations

→ Une question mal posée peut donner une mauvaise réponse, même si le modèle « sait » la bonne.

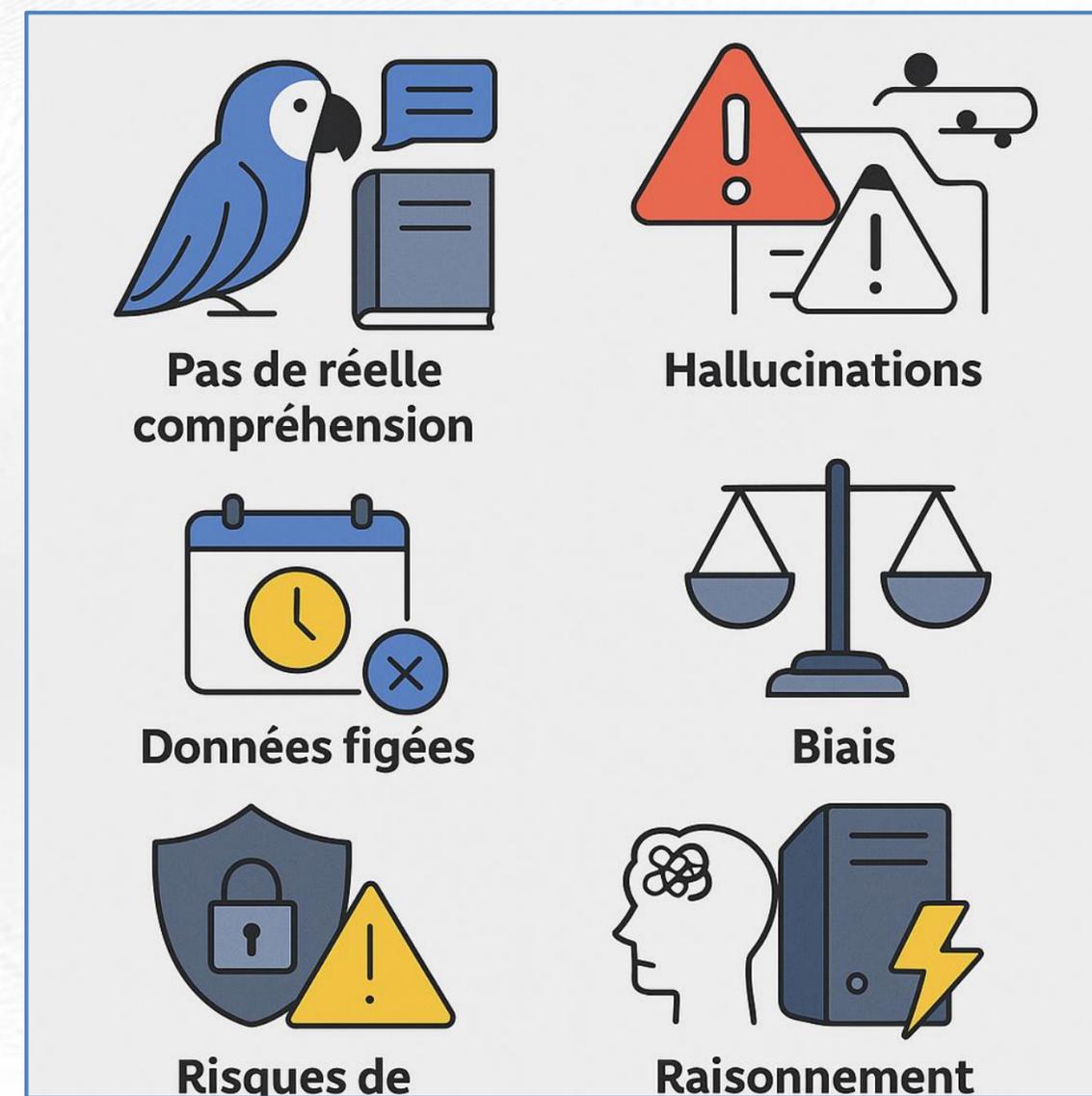
## 7. Limités dans le temps

→ Ils n'ont pas accès aux événements récents (sauf si connectés au web).

## 8. Peuvent être manipulés

→ Avec certaines formulations, on peut les pousser à donner des réponses biaisées ou inappropriées.

:



# RAG (Génération Augmentée de Récupération)

Fonctionnement du RAG (Retrieval-Augmented Generation) : un **moteur de recherche d'information** avec un **modèle de génération de texte**

## 1. Entrée utilisateur (Prompt + Query)

L'utilisateur saisit une requête (question ou instruction).

*Exemple : "Explique-moi les dernières avancées sur le traitement de l'asthme."*

## 2. Recherche d'informations pertinentes

Le système envoie la requête vers une base de **sources de connaissances** (documents, bases de données, articles...).

## 3. Récupération du contexte pertinent

Les documents ou extraits **les plus pertinents** sont récupérés et extraits. Ils forment le "**contexte enrichi**".

## 4. Construction d'une nouvelle requête enrichie

Le système combine :

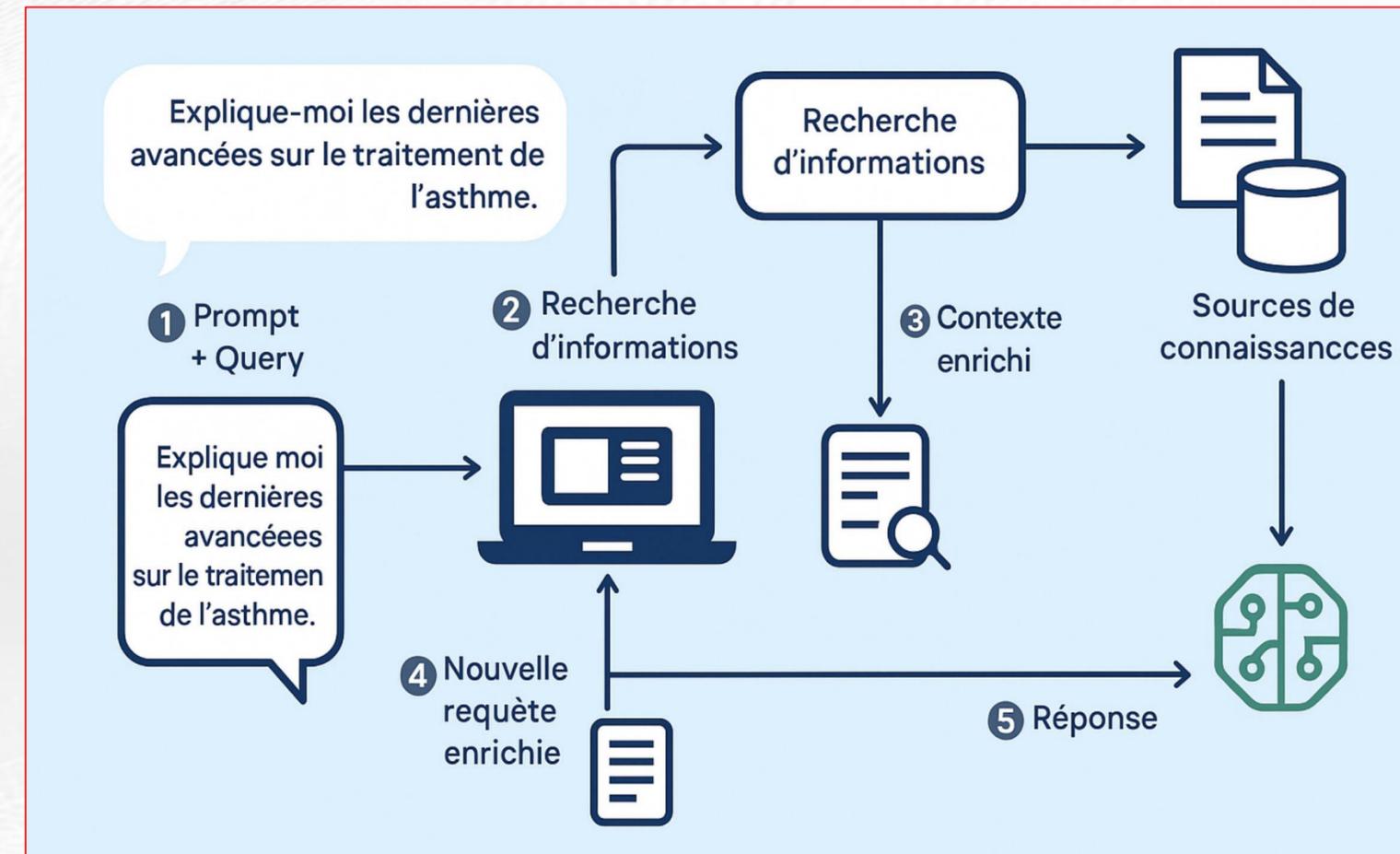
- le prompt original,
- la requête utilisateur,
- et le contexte enrichi

→ pour créer une nouvelle entrée plus complète.

## 5. Génération de la réponse finale

Le tout est envoyé au **LLM (Large Language Model)** qui génère une réponse en s'appuyant :

- sur la requête, mais aussi sur les **informations récupérées**.







# Échelle TRL de l'invention

L'échelle TRL (Technology Readiness Level, ou Niveau de Maturité Technologique en français) est un outil utilisé pour évaluer le degré de développement d'une technologie – de l'idée initiale jusqu'au produit prêt à être utilisé en conditions réelles.

:

TRL	Description	État de l'invention
✓ TRL 1–3	Recherche et preuve de concept	✓ Fait (modules RAG, LLM, BKT, IRT)
✓ TRL 4–5	Intégration en environnement simulé (Docker, tests)	✓ POC fonctionnel avec Docker + Moodle
● TRL 6	Démo prototype dans environnement test	✓ Validé en environnement maîtrisé
● TRL 7	Prototype testé en situation réelle (classe pilote)	⚠ À planifier avec utilisateurs réels
🕒 TRL 8–9	Déploiement opérationnel et usage à grande échelle	🚧 Non atteint



## Brevet d'invention

### Récépissé de remise des pièces

N° de dépôt de la demande: 70882

Date de dépôt : 11/03/2025

**Déposant(s):**

-Université Abdelmalek Essaâdi // Avenue Palestine, M'hanech II B.P. 2117 - 93030, Tétouan

**Mandataire:**

-Mohamed EL Harzli // Présidence de l'Université Abdelmalek Essaâdi- M'Hanech II - Avenue 9 avril - BP: 2117, Tétouan

**Intitulé de l'invention :**

-Plateforme intelligente d'e-learning à génération dynamique personnalisée basée sur RAG et apprentissage adaptatif

**Inventeur(s) :**

Kamal Rekloui ; Anass Belcaid

**Pièces remises:**

- Formulaire de dépôt de brevet
- Description de l'invention
- Revendication(s)
- Abrégé du contenu technique de l'invention
- Planches de dessins
- Etat d'engagement

Casablanca, le 12/03/2025



### Comparaison Des Algorithmes D'Apprentissage Adaptatif

	Algorithme	Type	Avantages	Limites	Données nécessaires
1	IRT (Item Response Theory)	Statistique/Probabiliste	Précision dans l'estimation des compétences, robuste	Suppose des hypothèses rigides, pas de dynamique	Réponses à des items avec paramètres
2	BKT (Bayesian Knowledge Tracing)	Probabiliste (Bayésien)	Simple à implémenter, interprétable	Limité à des hypothèses simples, pas d'évolution complexe	Séquence de réponses binaires
3	DKT (Deep Knowledge Tracing)	Réseau neuronal	Capture les séquences complexes, haute précision	Moins interprétable, besoin de données massives	Séquence de réponses avec timestamp
4	LFA (Learning Factor Analysis)	Statistique	Modélise les facteurs d'apprentissage	Besoin de modélisation explicite des facteurs	Réponses + métadonnées pédagogiques
5	PFA (Performance Factor Analysis)	Statistique	Tient compte des performances passées	Complexité croissante avec le nombre d'essais	Historique des réussites/échecs
6	CDM (Cognitive Diagnosis Models)	Probabiliste/Diagnostique	Analyse fine des forces/faiblesses	Nécessite des matrices de compétences précises	Réponses détaillées par compétence
7	Bandits Algorithms	Optimisation / Reinforcement learning	Bonne adaptabilité, exploration/exploitation	Peut être lent à converger	Feedbacks utilisateurs, performances
8	Systèmes de recommandation	Machine Learning / Filtrage	Personnalisation basée sur l'utilisateur	Nécessite beaucoup de données utilisateurs	Historique d'interactions, préférences



Xaliss / شكرا / Merci

