

Conception d'un STI adapté au MOOCs



Membres du groupe:

- DONANG LEMOTIEU AROLD
 - FOSSO DEFFO RUDY
 - KIIKI IROUME JOSEPH ARIEL
 - SONDI MANGA JEFRIED HUGO
 - TEDONZE NICK JOFREIN
-

Table de Matières

Résumé	3
Introduction	4
Méthodologie	4
modèle du STI	5
Phase d'expérimentation et résultats	14
Discussion	15
Conclusion	15
Bibliographie	15

Résumé

L'intelligence d'un système tient à sa capacité de s'adapter à l'utilisateur de façon dynamique. Les Systèmes Tutoriels Intelligents (STI) ou les systèmes d'aide à l'apprentissage visent à fournir à l'utilisateur des informations pertinentes lui permettant d'acquérir les connaissances adéquates et de résoudre le problème recherché par celui-ci. Des modèles sont nécessaires et notamment un modèle de l'étudiant. Dans le même esprit, la création de curriculum implique un support de connaissances diverses et la création d'un modèle du tuteur pour simuler le fonctionnement du tuteur dans ses fonctions académiques et pédagogiques. Les Systèmes d'Aide Intelligente à la Conception des Cours constituent une nouvelle catégorie de systèmes intelligents qui visent à aider un enseignant à concevoir un cours en fonction d'un auditoire donné. Le présent rapport se propose de décrire la conception et l'implémentation d'une plateforme de MOOCS selon le modèle de STI décrit plus haut; Après avoir effectué un rappel des différents types de modèles étudiants et de tuteurs existants et des techniques utilisées pour sa mise à jour dans les STI, nous allons concevoir un modèle de l'enseignant-auteur pour notre plateforme de Moocs basé sur la recommandation des cours adapté au profil de l'apprenant. Nous allons effectuer ensuite des tests pour confirmer les performances de notre modèle.

Introduction

Les systèmes tutoriels intelligents (STI) sont des environnements d'apprentissage informatisés issus de l'EIAO (Enseignement Intelligemment Assisté par Ordinateur) qui visent à personnaliser la formation. En effet ils ont été développés pour répondre aux limites de l'EAO (Enseignement Assisté par Ordinateurs) en ayant recours à l'intelligence artificielle pour mettre en place des systèmes plus souples et interactifs qui s'adaptent "aux besoins spécifiques de l'apprenant en évaluant et diagnostiquant ses problèmes afin de lui fournir l'aide nécessaire.

Les STI répondent ainsi au besoin de placer l'apprenant au centre du processus d'apprentissage, ceci en utilisant de plus en plus d'algorithmes et d'outils du NLP. Ceci étant, comment pouvons nous à partir des différents modèles de l'apprenant, de l'expert, et du tuteur, faire apprendre de manière efficace, les apprenants ?

Pour répondre à cette question nous allons mettre sur pied un MOOC qui s'adapte de façon intelligente aux besoins de l'apprenant en suivant les étapes suivantes : Mise sur pied d'un modèle d'évaluation et d'un modèle de recommandation, mise sur pied du MOOC à proprement parler(application Web).

Méthodologie

Pour réaliser le projet de mise en place d'un STI adapté aux MOOCs, il est nécessaire d'établir une méthodologie à suivre qui passe premièrement par la définition du modèle de STI utilisé c'est à dire la spécification du modèle de l'apprenant, du modèle du domaine et du modèle pédagogique utilisé. Ensuite il faut définir la structure de l'application autrement dit l'architecture logicielle, les technologies utilisées en frontend et en backend ainsi que le système de gestion de base de données utilisé pour assurer la persistance des données. Enfin il est primordial de définir la stratégie d'évaluation du modèle pour pouvoir conclure par la suite sur la pertinence de la méthode utilisée.

1. modèle du STI

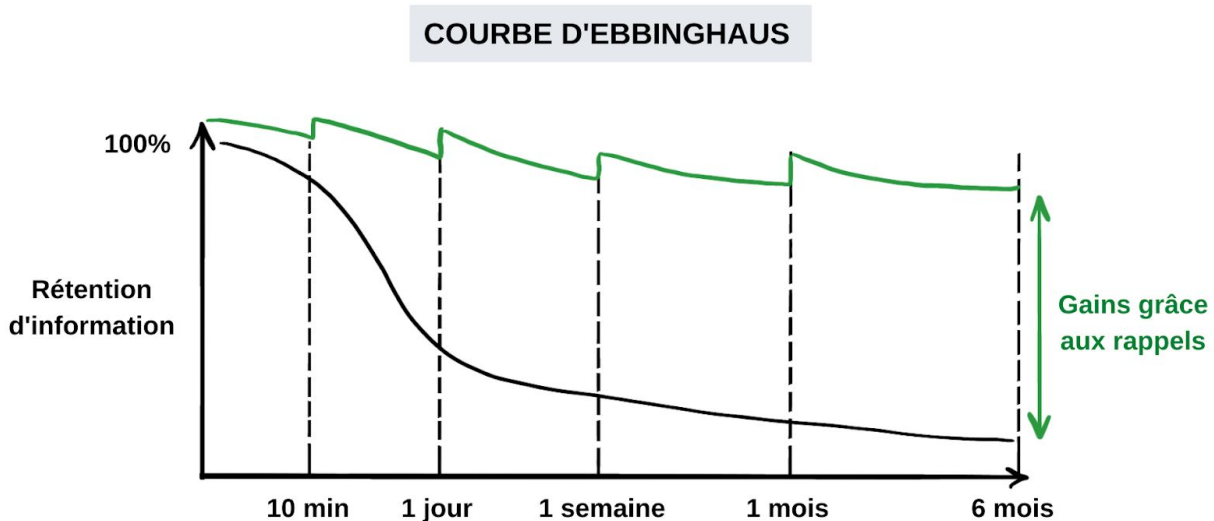
“ Les STI sont centrés sur l'apprenant, c'est-à-dire que le contrôle se base concrètement sur les actions effectuées ou sur les réponses que donne l'apprenant aux questions posées. Ainsi, c'est grâce au diagnostic cognitif posé à la suite de ce que fait l'apprenant que l'adaptation du système est possible. ” par Grandbastien et Labat

a. le modèle de l'apprenant

Le modèle de l'apprenant est la composante qui représente l'état des connaissances de l'étudiant. Il est utilisé pour guider le comportement du tuteur intelligent. Son rôle est de permettre une analyse approfondie des réponses de l'étudiant, la détection de ses erreurs et la production d'interventions adéquates par le tuteur. Sa modélisation peut aller du plus simple au complexe selon le niveau de précision et d'importance que l'on accorde à chaque aspect de l'apprenant réel. Comme la connaissance experte est résidente dans le module expert du domaine, le modèle étudiant est habituellement construit en relation avec celui-ci. Trois principales techniques, caractérisant ainsi le type de modèle concerné, sont utilisées pour mettre à jour le modèle étudiant :

- **Le modèle de recouvrement (Overlay Models)** : Ici on suppose que toutes les différences entre le comportement de l'étudiant et le comportement de l'expert peuvent être expliquées par le manque d'habiletés de l'étudiant et la connaissance de l'apprenant est vu comme un sous-ensemble de celle de l'expert qui reste à compléter.
- **Le modèle différentiel** : est une extension du modèle de recouvrement. Au lieu de comparer la connaissance de l'étudiant avec celle de l'expert, le modèle différentiel est produit en comparant la performance de l'étudiant avec la performance de l'expert dans la même situation.
- **Le modèle correctionnel** : il est utilisé pour représenter les croyances des étudiants à l'extérieur des connaissances du modèle expert. La technique revient à coder la connaissance experte puis à l'augmenter avec les incompréhensions ou les erreurs que fait l'apprenant.

Nous avons opté ici pour un modèle de recouvrement (Overlay Models). En plus de ceci nous avons également modélisé la mémoire d'un apprenant qui contient un ensemble de connaissances qui se dégradent au fur et à mesure que le temps passe suivant la courbe de d'Ebbinghaus encore appelée courbe de l'oubli ci-après :



Sur cette courbe, nous pouvons voir en noir le pourcentage de rétention en fonction du temps sans un système de rappel et en vert le pourcentage de rétention avec un système de rappels. Fort est de constater qu'avec un système de rappel adéquat on peut maintenir le niveau de rétention pour un cours à 100%. Nous avons donc mis en place un système de rappel pour la révision des connaissances déjà acquises afin de leur faire persister dans la mémoire de l'apprenant.

b. le modèle du domaine

Le module expert du domaine contient l'ensemble des connaissances du domaine à enseigner ainsi que les liens qui les réunissent. Il a pour tâche la résolution des problèmes du domaine. Aussi appelé modèle expert, il représente en quelque sorte " l'apprenant idéal ". Il peut être modélisé de trois façons :

-
- **approche boîte noire** : appliquer une quelconque méthode de raisonnement sur le domaine, non explicitée, sans aucune transparence à l'utilisateur.
 - **un système expert** : peut expliquer ses raisonnements
 - **Un modèle cognitif** qui simule la façon dont l'humain utilise ses connaissances.

Nous avons opté ici pour le modèle du système expert qui peut expliquer ses raisonnements. En effet l'évaluation ici étant faite via une série de QCMs posés à l'apprenant, il est bénéfique pour lui de connaître pourquoi une proposition est la réponse et pas les autres.

c. le modèle pédagogique

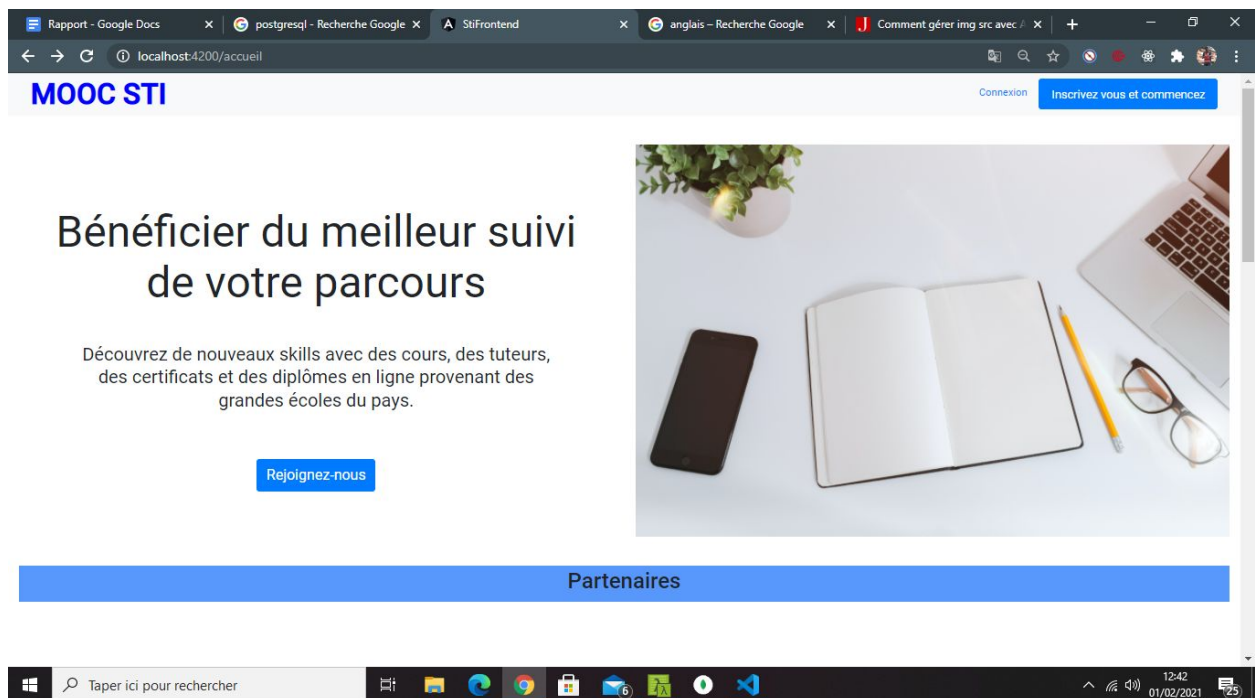
C'est ce modèle qui définit comment enseigner. Il a deux principales fonctions que sont: susciter l'apprentissage et l'évaluer. Ces deux fonctions sont traitées soit séparément soit conjointement. La question principale ici est de savoir quand et comment intervenir auprès de l'apprenant pour le soutenir dans son apprentissage. La fonction de soutien à l'apprentissage peut elle-même se subdiviser en deux sous-fonctions: le diagnostic cognitif, qui consiste à détecter les sources des erreurs, et la sélection de stratégies de tutorat ou de remédiation qui soient les mieux adaptées. Pour ce faire, on divise chaque cours en sections et les QCMs qui y correspondent à la fin d'une évaluation on détermine les pourcentages d'erreurs de l'apprenant face aux QCMs de chaque section. La prochaine évaluation prendra en compte ces pourcentages d'erreur par section; Ainsi les QCMs des sections les moins assimilées seront majoritaires proportionnellement aux taux d'erreurs. De plus une telle division des cours en sections permet de compartimenter le contenu du cours ce qui permet à l'apprenant de mieux appréhender la structure du cours et ainsi d'accélérer son processus d'apprentissage. Un système de rappel est aussi implémenté pour garantir la persistance des connaissances apprises au fur et à mesure de son évolution sur la plateforme de MOOCs.

2. Structure de l'application

Pour la mise sur pied de notre application, nous avons opté pour une architecture 3 tiers, avec un client, un serveur web et un serveur de base de données. Le client web fournit une interface agréable et adaptée à l'évolution des cours. Dans le serveur web, s'exécutent les fonctions métiers et les algorithmes d'IA et le serveur de base de données assure la persistance des données.

a. Frontend

Le frontend pour notre application fournit une interface dynamique et ergonomique, basée sur les technologies web telles que HTML, CSS et un framework javascript. Le frontend fournit un espace apprenant bien élaboré où il peut s'inscrire/se connecter, visualiser ses cours, suivre ses cours, passer ses évaluations et suivre son évolution. Il peut également consulter la liste des cours qui lui sont recommandés ou rechercher un cours dans une barre de recherche. Nous avons utilisé les couleurs bleu et blanc pour l'élaborer.



b. Backend

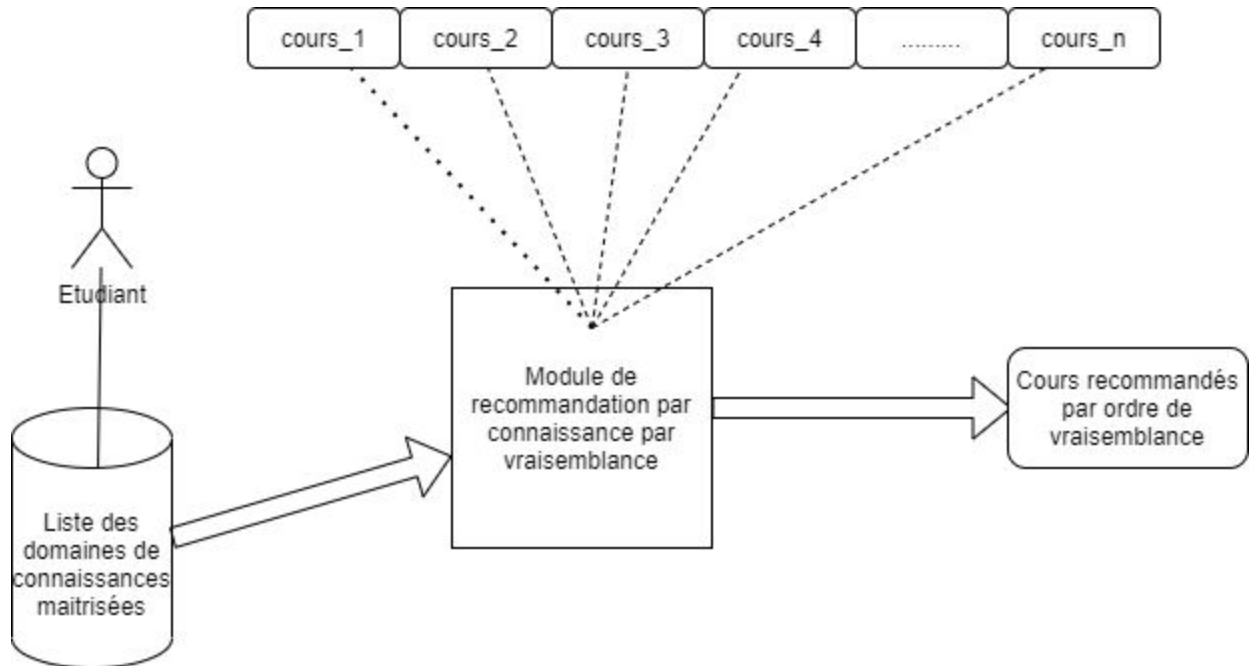
1) Technologie d'implémentation

Un back-end (parfois aussi appelé un arrière-plan) est un terme désignant un étage de sortie d'un logiciel devant produire un résultat; c'est celui ci qui se charge d'effectuer tous les traitements nécessaires au fonctionnement de notre application. la technologie que nous avons utilisé dans notre cas est le Framework Django basé sur le langage python. Nous avons choisi Django pour plusieurs raisons diverses; Premièrement notre modèle d'intelligence artificielle étant implémenté en Python il semblait évident de choisir Django comme outil d'implémentation de notre Backend, Deuxièmement Django est avant tout un framework qui permet de développer un site web orienté contenu très rapidement. Il simplifie la gestion des cas classiques comme la création de pages statiques, la pagination, les listes d'objets, les systèmes d'authentification, etc. Django est rapide, et pas seulement en termes de développement. De nombreux sites à fort trafic choisissent Django pour ses performances Web : temps de chargement des pages, affichage des contenus, etc. Enfin, en termes de sécurité, Django vous aide à vous protéger contre les attaques les plus fréquentes : injection SQL, cross-site scripting, clickjacking, etc.

2) Module de recommandation de cours

2.1) Recommandation par domaine de connaissance

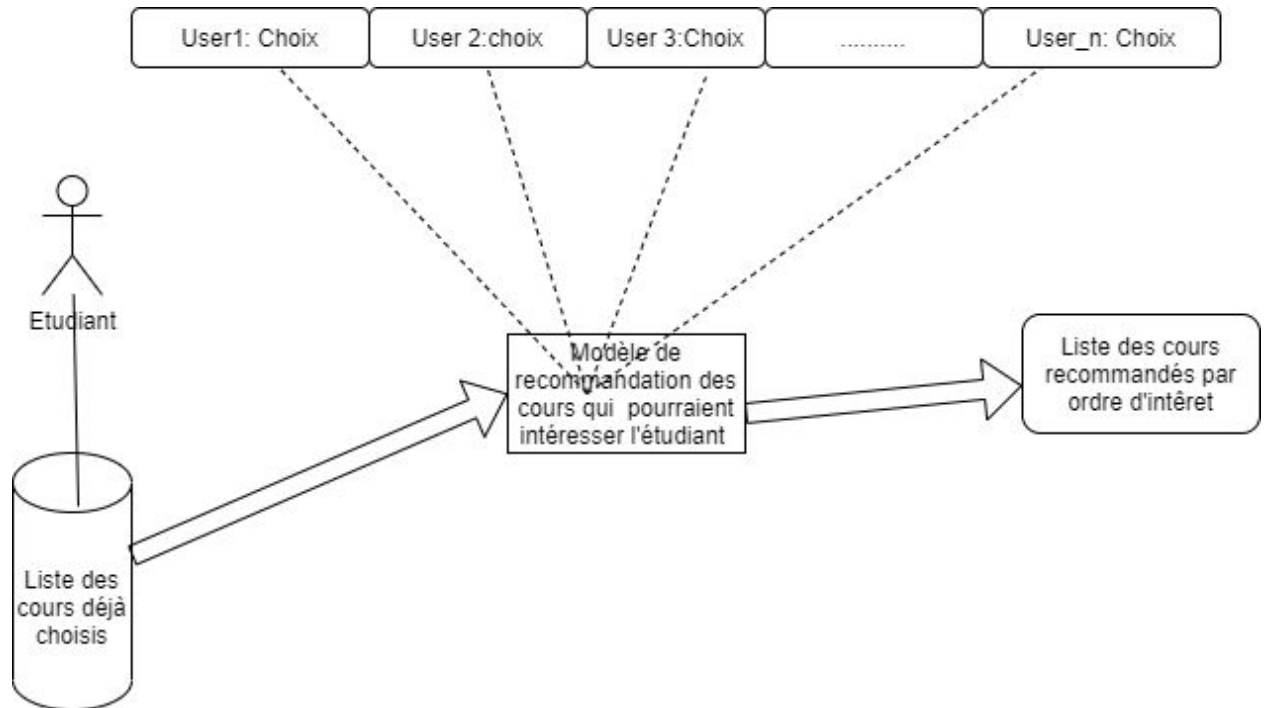
Pour modéliser notre tuteur nous avons mis sur pied un système expert qui à partir des connaissances de l'apprenant recommande un cours à l'apprenant. Pour ce faire nous avons associé à chaque cours un ensemble de domaine de connaissance. Ainsi pour la prédiction pour un étudiant donné nous calculons la vraisemblance de son aptitude à suivre un cours donné en fonction des prérequis dont il dispose. Au final nous préconisons les cours par ordre de vraisemblance.



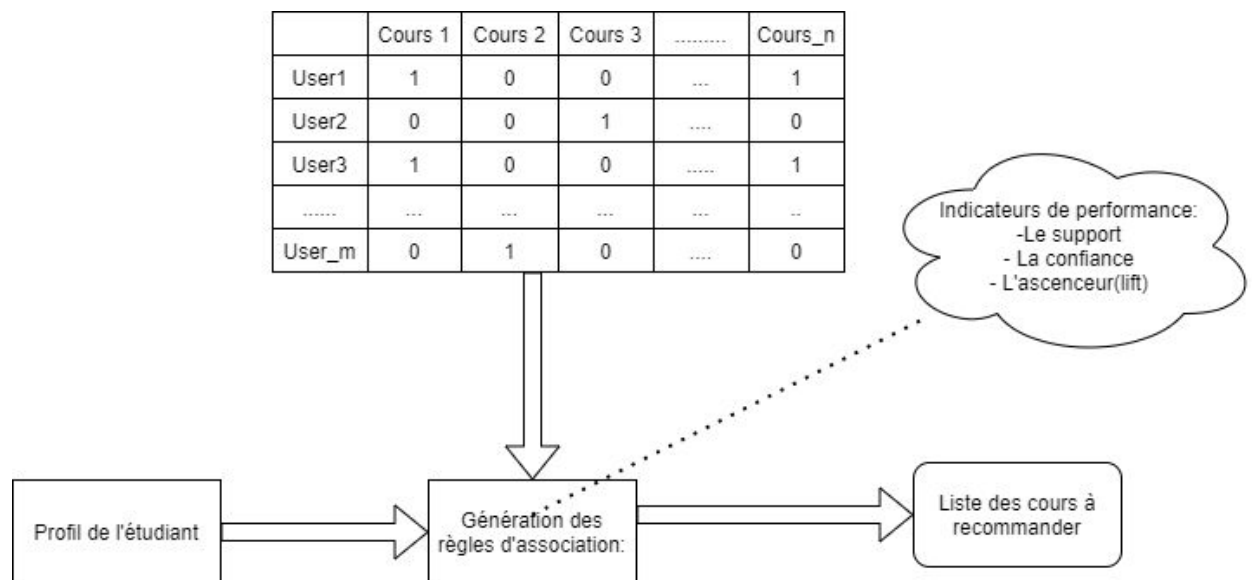
Ainsi chaque étudiant aura une vue différente de l'ensemble des cours recommandés en fonction des domaines de connaissance dont il dispose d'avance.

2.2) Recommandation par profil des étudiants (Association rule Mining)

Ici ce module de recommandation se base sur le choix des différents étudiants pour orienter les nouveaux étudiants sur la plateforme. Ainsi en fonction des choix des autres on guide un nouvel étudiant à travers le mécanisme de recommandation.



Ainsi si l'ensemble des étudiants ayant suivi le cours X sont intéressés par le cours Y alors on préconisera au prochain venu sur la plateforme ayant ce profil de suivre le cours Y. Pour ce faire nous avons utilisé plusieurs indicateurs statistiques pour la recommandation à savoir la confiance, le support et l'ascenseur.



Soit à évaluer la règle suivante $X \Rightarrow Y$ pour un ensemble de transactions T

-
- **Le support** de X relatif à un ensemble de transactions de T représente l'ensemble des transactions contenant X

$$\text{supp}(X) = \frac{|\{t \in T; X \subseteq t\}|}{|T|}$$

- **La confiance**(confidence) de $X \Rightarrow Y$ représente l'ensemble des transactions t contenant Y et sachant X

$$\text{conf}(X \Rightarrow Y) = \text{supp}(X \cup Y) / \text{supp}(X)$$

- **L'Ascenseur(lift)**: c'est un ratio qui caractérise la pertinence d'une association

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \times \text{supp}(Y)}$$

3) Modèle de l'évaluation de l'étudiant

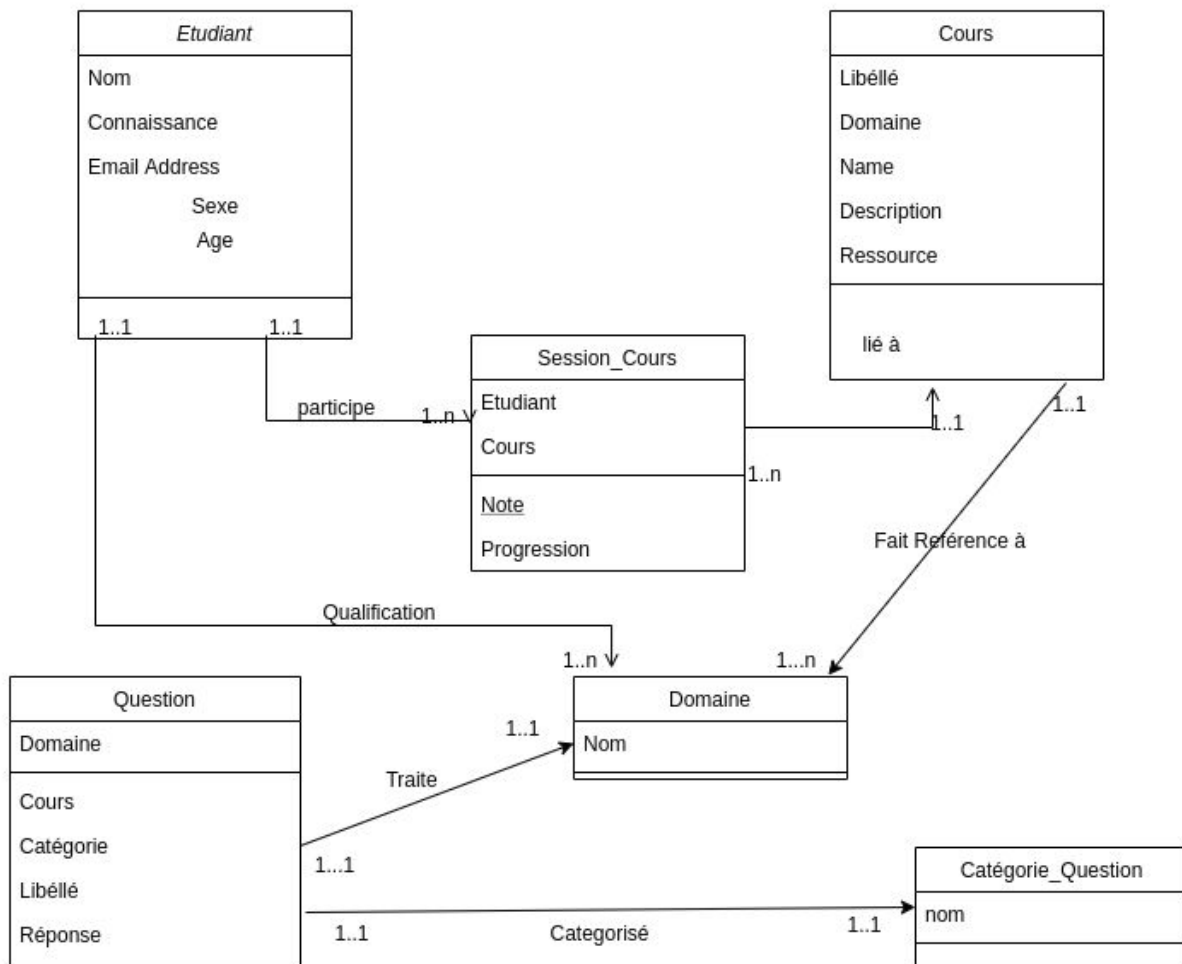
Pour l'évaluation de l'étudiant, il nous fallu tout d'abord faire un modèle de choix des questions pour la construction des évaluations ; pour le faire, nous avons lié chaque question de notre base de données à une section du cours car nos cours étaient divisés en plusieurs chapitres. Notre modèle d'évaluation est basé sur une série de 20 QCMS ; Initialement on suppose que le premier examen de l'étudiant sera basé de façon équitable sur les différentes sections du cours et à partir des examens suivants on va générer les questions de l'examen en respectant les proportions des sujets sur lesquels l'étudiant a eu le plus de difficultés. L'évaluation sera donc progressive et on va arrêter d'évaluer l'étudiant sur un sujet uniquement quand on aura dépassé le quota d'admission à l'examen. Une fois l'étudiant admis à l'examen, on pourra ajouter les domaines d'expertise touchés par le cours à l'ensemble de ses connaissances.

c. Base de données

Pour que notre application fonctionne correctement, nous avons besoin de stocker les données des utilisateurs (noms , mots de passe, évolution, cours en évolution, cours en cours, cours terminés...). Pour la persistance des données, nous utilisons le SGBD relationnel **Postgresql**. Il est mis en connexion avec le backend pour l'accès aux données. Dans notre modèle les entités que nous avons eu besoin de stocker :

- **Les domaines d'expertise:** Les domaines d'expertise représentent l'ensemble de domaines de connaissances que va traiter notre plateforme ,l'ensemble de domaines dont vont parler les cours de notre plateforme.
- **Les Cours et leurs Supports :** les cours proprement dit et les divers sujets d'interactions entre le tuteur et l'étudiant.
- **L'Étudiant :** celui-ci va représenter chaque étudiant ainsi que les données constituant les profils , les préférences et les comportements des étudiants .
- **Les Sessions de Cours :** une session représente une interaction d'un étudiant avec la base de connaissance.une session sera créé chaque fois que l'étudiant va s'inscrire à un cours et aura pour rôle de stocker les données de l'étudiant relativement au cours qu'il est entrain de suivre ;on pourra par exemple stocker la note de l'étudiant ainsi que sa progression sur le dit cours
- **Les Questions :** Les Questions constituent la base du système d'évaluation de l'étudiant ; nous avons donc stocker les questions relatives aux différentes notions pour permettre une évaluation efficace de l'étudiant dans les domaines où il montre le plus de difficultés .

Nous pouvons éventuellement stocker les différentes catégories de questions pour généraliser le modèle à des cas où les évaluations pourront avoir différentes formes.en tenant compte de tous les éléments et entités précédentes , nous pouvons avoir le Modèle de conception (MCD) de la figure la suivante:

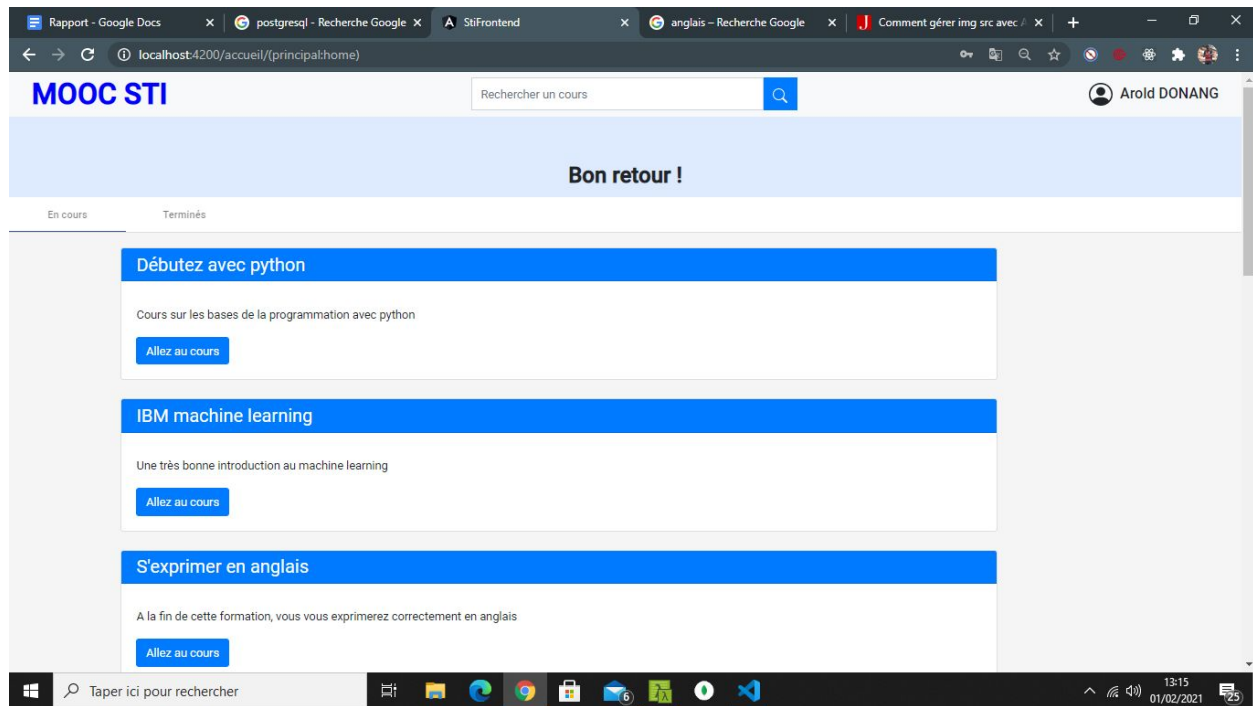


3. Aperçu de la solution

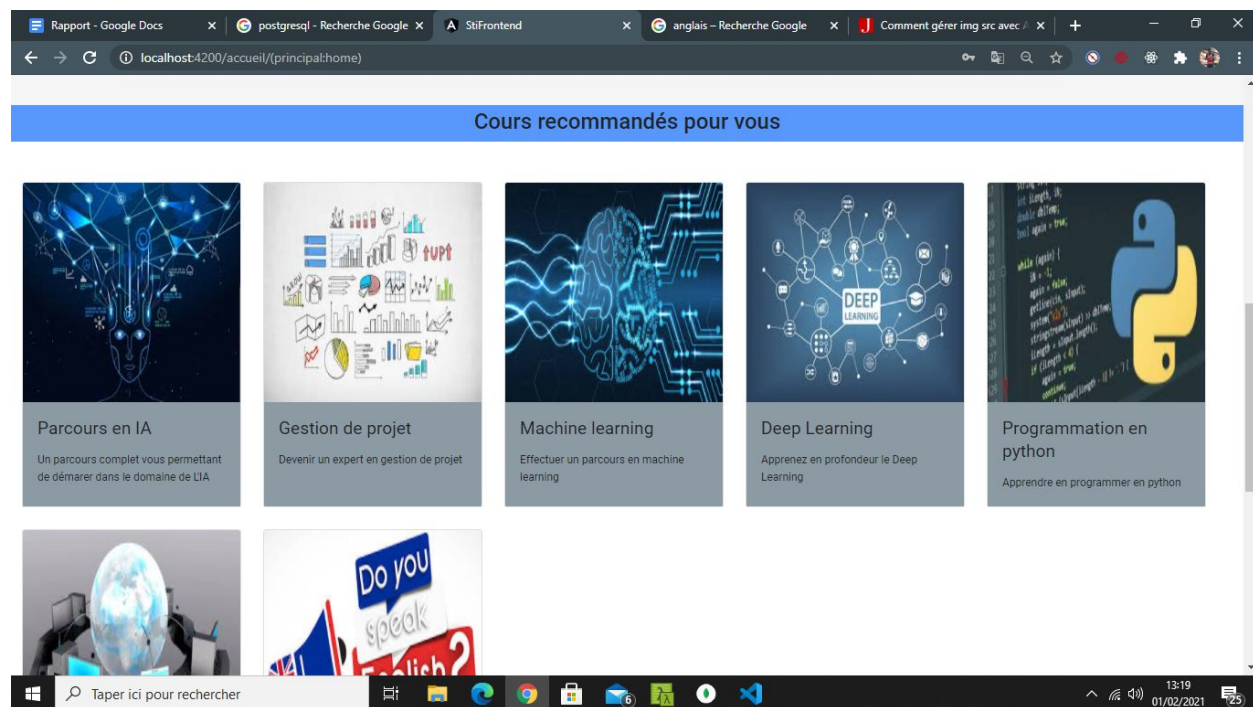
Aperçu de la solution

L'IHM est souple, fluide et facile de prise en main par l'apprenant. Sur sa page, il aperçoit les cours en cours de déroulement, ses cours terminés et il peut consulter une section de recommandation pour voir les cours recommandés pour lui.

Dans la figure suivante, nous avons l'interface apprenant qui nous montre la liste de ses cours en cours de déroulement.



L'interface de l'apprenant lui présente aussi le cours qui lui sont recommandés par le moteur de recommandation, en fonction des cours auxquels il est inscrit. L'image suivante présente la section de recommandation des cours .



L'apprenant a aussi la possibilité de faire une recherche sur les thèmes qui l'intéressent dans la barre de recherche

Discussion

Au regard de certaines plateformes existantes de tutoring à l'instar de Moodle (Modular Object Oriented Dynamic Learning Environment) notre travail ajoute un système de recommandation relevant du data mining et de l'intelligence artificielle enfin d'assurer un confort de l'apprenant et son évaluation. Nous avons aussi ajouté une représentation des cours en domaine de connaissance en vue de leur classification.

Comme perspective nous comptons améliorer notre module d'évaluation par génération automatique des questions pour un cours donné grâce à un module d'intelligence artificielle. Cela permettra de faciliter davantage le travail de l'expert.

Conclusion

En définitive, tout au long de ce projet il était question pour nous de mettre sur pied un système tuteur intelligent, capable de faire apprendre les apprenants de manière efficace en fonction de leur profils. Pour y parvenir, nous avons mis sur pied un MOOC, qui est une plateforme de cours en ligne permettant aux apprenants qui s'y connectent d'apprendre des cours en fonction de leurs besoins; A cela nous avons ajouté un aspect d'intelligence artificielle permettant de profiler chaque apprenant en vue de leur offrir des cours adaptés à leurs niveaux ainsi que des évaluations adaptées à leur évolution notamment grâce à la mise sur pied d'un modèle d'évaluation et d'un modèle de recommandation en fonction du profil. Les résultats obtenus au terme de notre travail sont concluants, car nous avons pu recommander des cours adaptés lors des tests à un apprenant ainsi qu'une évaluation. Pour améliorer davantage l'efficacité de notre STI nous pouvons envisager la possibilité d'ajouter dans les jours à venir un système de traitement du langage naturel afin de prendre en compte les réactions des apprenants pendant le suivi des cours.

Bibliographie

- 1) <https://www.appvizer.fr/services-informatiques/framework/django>
- 2) <http://www.iro.umontreal.ca/~frasson/FrassonPub/ICO-93-Modele%20auteur.pdf>
- 3) https://en.wikipedia.org/wiki/Association_rule_learning
- 4) <http://gdac.uqam.ca/inf7470/notesDeCours/exemplesSTI.pdf>
- 5) <https://hal.archives-ouvertes.fr/hal-00696375/document>
- 6) <https://core.ac.uk/download/pdf/9540196.pdf>