



Rapport de Projet TP1 : Système Intelligent de Recommandation Académique

Othman SALAHI
Malak HOUALI
Mohamed MAKRANI

Groupe : 2

23 Janvier 2026

Table des matières

1	Introduction	2
2	Gestion du Projet et Méthodologie Agile	2
2.1	Vue Globale du Product Backlog	2
3	Sprint 1 : Analyse et Conception	2
3.1	Objectifs du Sprint	2
3.2	Travaux Réalisés	3
3.3	Diagramme de Cas d'Utilisation	3
3.4	Diagramme de Classes	4
4	Sprint 2 : Développement de l'Intelligence Artificielle	5
4.1	Objectifs du Sprint	5
4.2	Travaux Réalisés	5
5	Sprint 3 : Développement Applicatif	6
5.1	Objectifs du Sprint	6
5.2	Réalisations Techniques	6
6	Sprint 4 : Tests et Validation	7
6.1	Objectifs du Sprint	7
6.2	Validation et Livraison	7
7	Organisation et Structuration par Épics	8
7.1	Détail des Épics	8
8	Présentation de l'Application Finale	9
8.1	Authentification et Profil	9
8.2	Tableau de Bord et Recommandations	9
8.3	Espace Administration	11
9	Problèmes Rencontrés et Solutions	11
9.1	Problème : Incompatibilité des Types de Données	11
9.2	Solution Mise en œuvre	11
10	Installation et Déploiement	12
10.1	Prérequis	12
10.2	Procédure de Lancement	12
11	Conclusion	12

1 Introduction

Dans le cadre du module *Méthodes Agiles*, ce projet vise la conception et le développement d'un **système intelligent de recommandation académique** destiné aux étudiants universitaires marocains. L'objectif principal est d'exploiter des techniques d'Intelligence Artificielle afin d'analyser le profil académique d'un étudiant et de proposer une orientation pédagogique adaptée à ses performances et caractéristiques scolaires.

Le projet a été mené en appliquant rigoureusement la méthodologie **Agile Scrum**, avec une organisation du travail en sprints successifs, un suivi via l'outil **Jira** et une livraison incrémentale des fonctionnalités.

2 Gestion du Projet et Méthodologie Agile

Le projet a été structuré selon le framework **Scrum**, favorisant une approche itérative, collaborative et orientée valeur. Les rôles Scrum ont été répartis au sein de l'équipe, et les artefacts principaux (Product Backlog, Sprint Backlog, incréments) ont été maintenus tout au long du cycle de développement.

La planification et le suivi des tâches ont été réalisés à l'aide de **Jira**, permettant une visualisation claire de l'avancement, des priorités et des dépendances entre les différentes activités.

2.1 Vue Globale du Product Backlog

Le Product Backlog regroupe l'ensemble des fonctionnalités fonctionnelles et techniques nécessaires à la réalisation du système de recommandation académique.

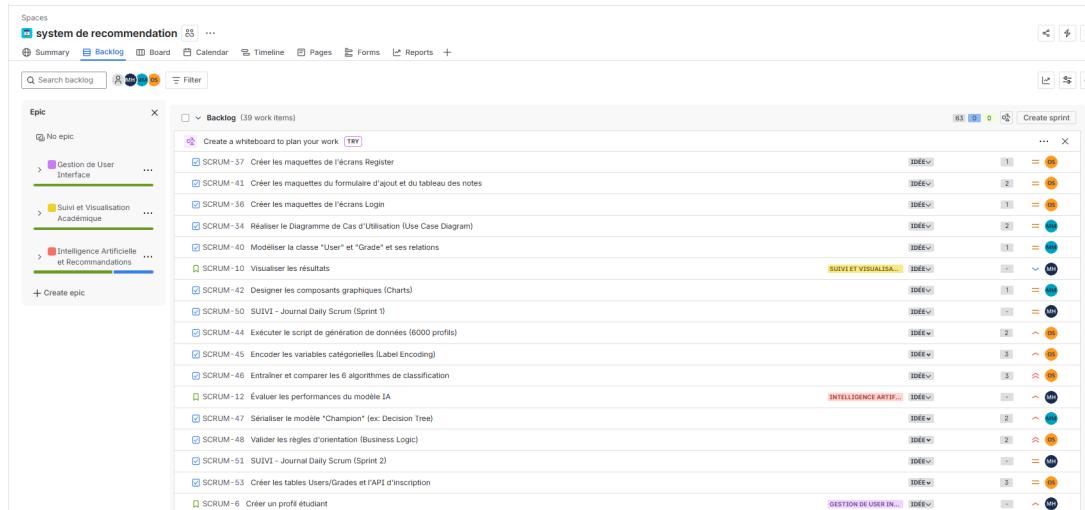


FIGURE 1 – Product Backlog du projet sur Jira

3 Sprint 1 : Analyse et Conception

3.1 Objectifs du Sprint

Le Sprint 1, intitulé *Analyse et Conception*, avait pour objectif de poser les fondations du projet. Il s'est concentré sur la compréhension du besoin métier, la modélisation du

système et la conception des interfaces utilisateur, sans implémentation logicielle directe.

3.2 Travaux Réalisés

Les principales tâches réalisées durant ce sprint sont les suivantes :

- **Analyse des besoins fonctionnels** du système de recommandation.
- **Conception des maquettes UI** des écrans principaux (Login, Register, gestion des notes).
- **Modélisation UML** à travers des diagrammes de cas d'utilisation et de classes.
- **Définition des entités clés** du système (User, Grade, Profil Étudiant).

Ces tâches correspondent aux tickets Jira suivants :

- SCRUM-34 : Diagramme de cas d'utilisation
- SCRUM-40 : Modélisation de la classe "User" et "Grade"
- SCRUM-36 : Maquettes écrans Login
- SCRUM-37 : Maquettes écrans Register

3.3 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation permet d'identifier les interactions principales entre l'étudiant et le système. Les fonctionnalités clés incluent la création de profil, la saisie des notes académiques, la génération de recommandations IA et la visualisation des résultats.

Academic Recommendation System Use Case Diagram

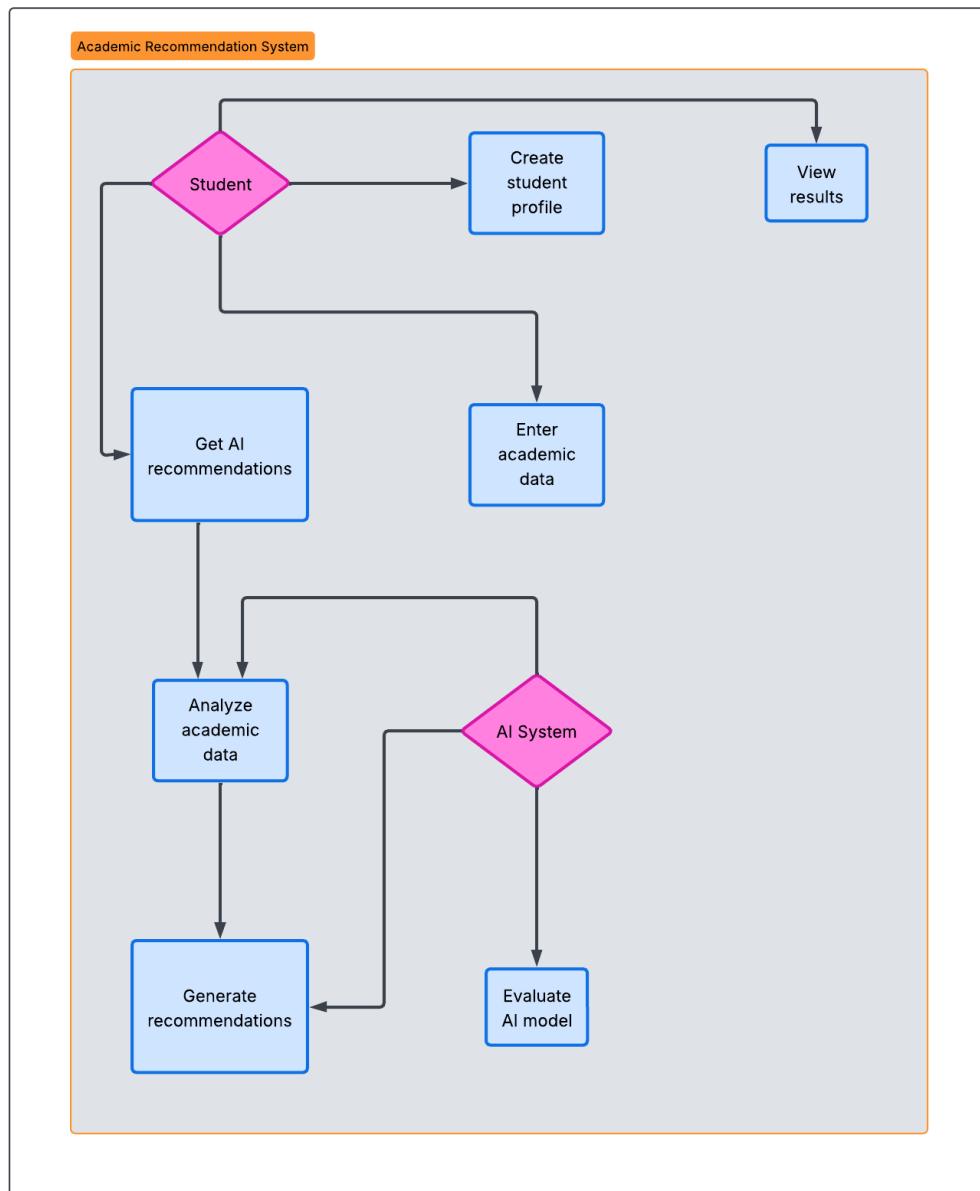


FIGURE 2 – Diagramme de Cas d’Utilisation

3.4 Diagramme de Classes

Le diagramme de classes met en évidence la structure statique du système, notamment les classes **User** et **Grade**, ainsi que leurs relations.

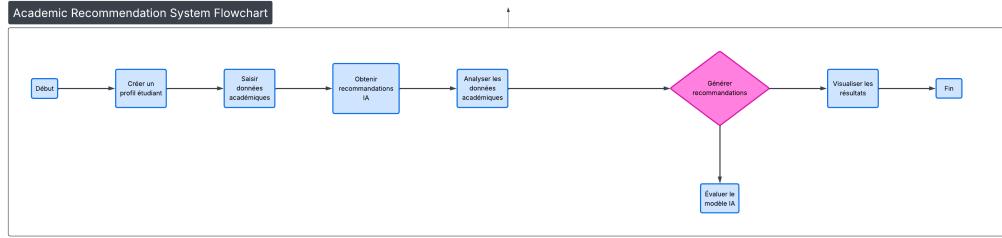


FIGURE 3 – flowchart d’application

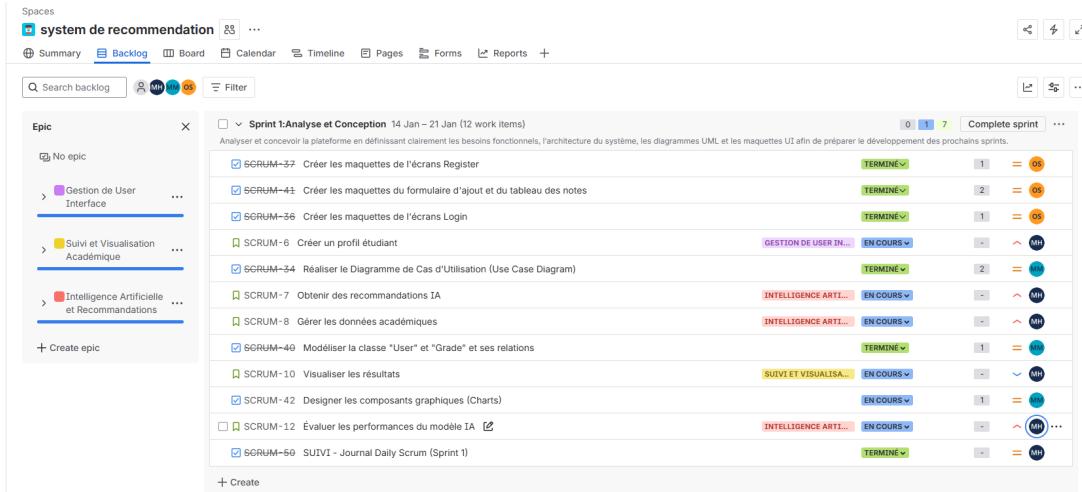


FIGURE 4 – Sprint 1 – Tableau Jira : Analyse et Conception

4 Sprint 2 : Développement de l’Intelligence Artificielle

4.1 Objectifs du Sprint

Le Sprint 2 avait pour objectif principal l’implémentation complète de la brique **Intelligence Artificielle**. Il s’est focalisé sur la préparation des données, l’entraînement de plusieurs modèles de classification et la sélection du modèle le plus performant.

4.2 Travaux Réalisés

Un dataset synthétique de **6000 profils étudiants** a été généré (SCRUM-44) pour entraîner les modèles. Six algorithmes de classification (Decision Tree, Random Forest, SVM, etc.) ont été comparés (SCRUM-46). Le modèle **Decision Tree** a été retenu et sérialisé (SCRUM-47) pour son excellent rapport performance/interprétabilité.

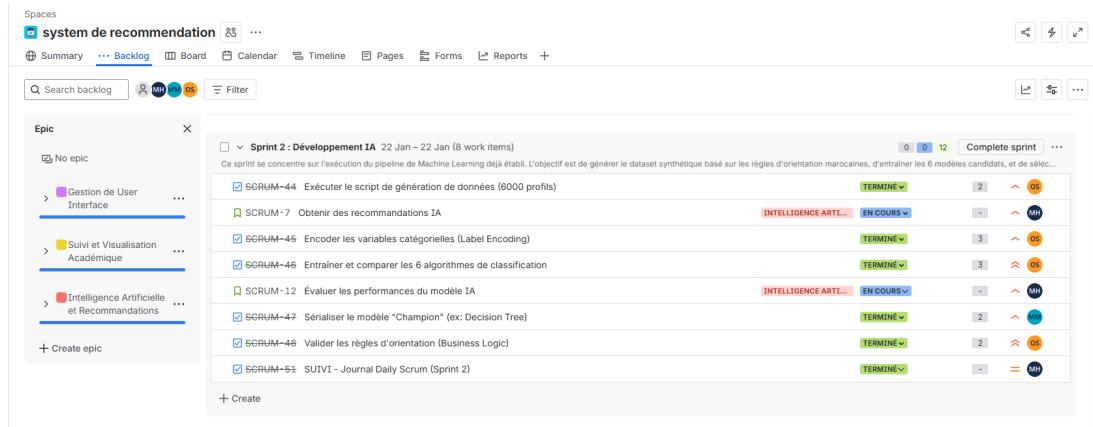


FIGURE 5 – Sprint 2 – Tableau Jira : Développement IA

5 Sprint 3 : Développement Applicatif

5.1 Objectifs du Sprint

Le Sprint 3 a marqué la phase de construction de l’application web. L’objectif était de créer l’architecture Backend, de développer le Frontend et d’intégrer le modèle IA entraîné au Sprint précédent.

5.2 Réalisations Techniques

L’équipe a travaillé sur la mise en place complète de la stack technique :

- **Base de données (SCRUM-53/15)** : Création des tables `Users` et `Grades`.
- **Backend API (SCRUM-54/63)** : Développement des endpoints CRUD pour les notes et de la route `/predict` pour l’orientation.
- **Frontend Authentication (SCRUM-61/56)** : Implémentation sécurisée du Login/Register et gestion de sessions.
- **Visualisation (SCRUM-58)** : Intégration de Chart.js pour afficher les résultats graphiques.
- **Administration (SCRUM-59)** : Crédit d’une page Admin pour le monitoring des performances du modèle.

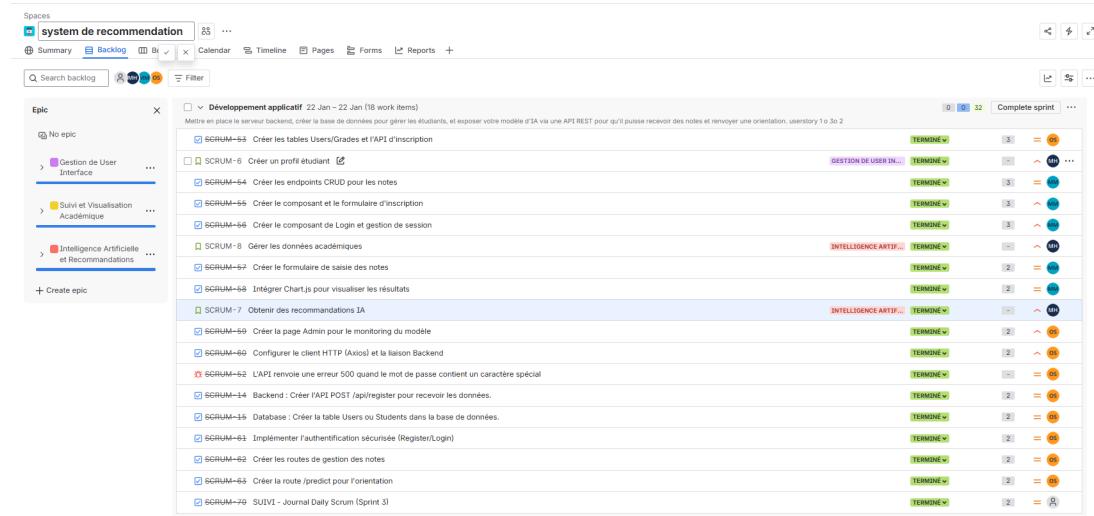


FIGURE 6 – Sprint 3 – Tableau Jira : Développement Applicatif

6 Sprint 4 : Tests et Validation

6.1 Objectifs du Sprint

Le dernier sprint a été consacré à la stabilisation de l'application, à la validation fonctionnelle et à la documentation finale du projet.

6.2 Validation et Livraison

Les efforts se sont portés sur :

- **Tests Fonctionnels (SCRUM-64)** : Exécution du parcours utilisateur complet (de l'inscription à la recommandation) pour s'assurer de l'absence de bugs critiques.
- **Performance (SCRUM-65)** : Validation des temps de réponse de l'IA.
- **Documentation (SCRUM-66/67)** : Rédaction du rapport technique et du guide d'installation (README).

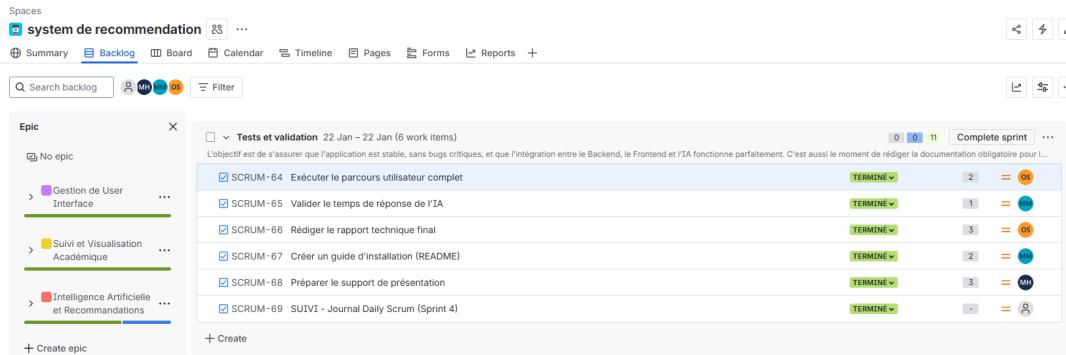


FIGURE 7 – Sprint 4 – Tableau Jira : Tests et Validation

7 Organisation et Structuration par Épics

Pour assurer une meilleure lisibilité du Product Backlog et garantir que chaque développement apporte de la valeur à une fonctionnalité précise, nous avons regroupé l'ensemble des User Stories en trois grandes **Épics**.

Cette structuration nous a permis de maintenir une séparation claire entre le développement de l'interface, la logique métier de visualisation et le cœur intelligent du système.

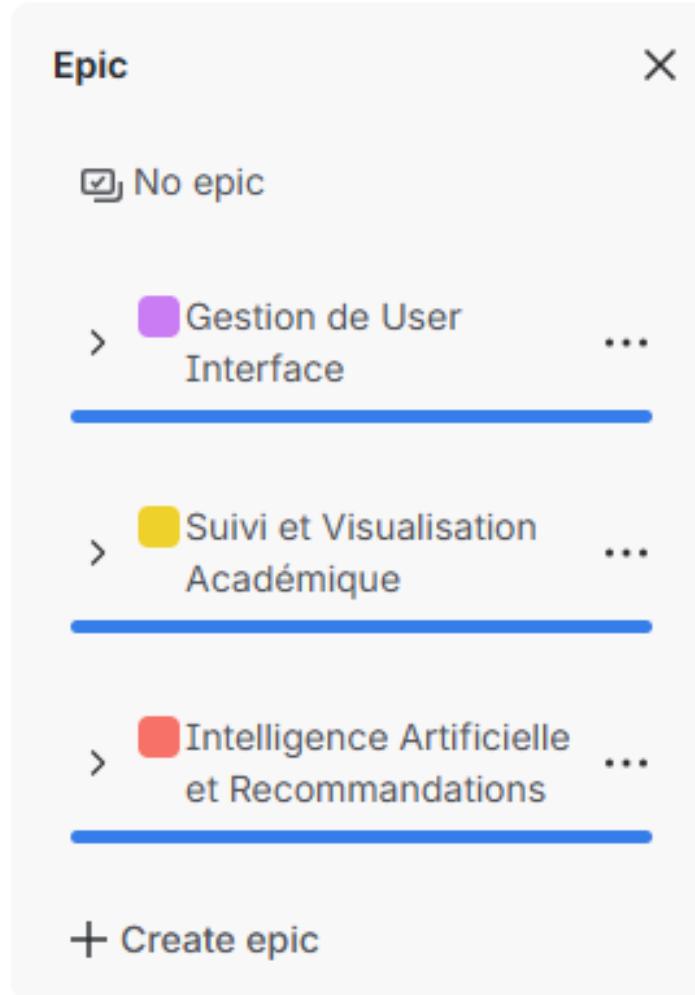


FIGURE 8 – Les trois Épics structurantes du projet sur Jira

7.1 Détail des Épics

— **Gestion de User Interface :**

Cette Épic regroupe toutes les tâches liées à l'expérience utilisateur et à la navigation. Elle inclut la conception des maquettes, l'intégration du Frontend (HTML/CSS/JS), ainsi que les mécanismes d'authentification et de sécurité (Login, Register, gestion de session).

— **Suivi et Visualisation Académique :**

Elle concerne la partie "Dashboard" de l'application. L'objectif était de transformer les données brutes (notes, moyennes) en indicateurs visuels pertinents. C'est ici que

s'intègrent les graphiques interactifs (Chart.js) et les tableaux de bord permettant à l'étudiant de suivre son évolution.

— **Intelligence Artificielle et Recommandations :**

C'est le cœur du projet. Cette Épic rassemble les tâches de Data Science : la génération du dataset synthétique, le nettoyage des données, l'entraînement des modèles (Decision Tree, etc.), la sérialisation du modèle champion et son exposition via une API pour générer les prédictions d'orientation.

8 Présentation de l'Application Finale

L'application résultante offre une interface moderne et intuitive pour les étudiants et les administrateurs.

8.1 Authentification et Profil

L'étudiant commence par créer un compte et se connecter pour accéder à son espace personnel.

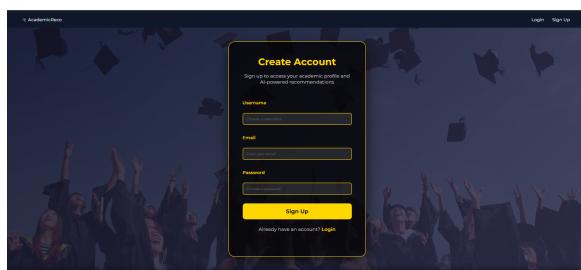


FIGURE 9 – Écran d'inscription

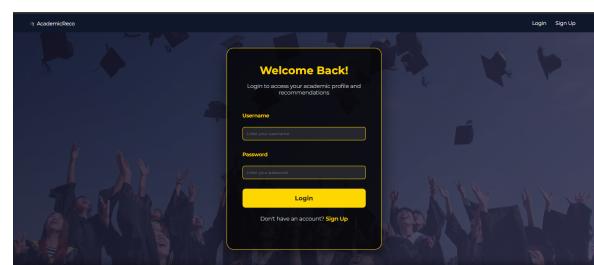


FIGURE 10 – Écran de connexion

L'étudiant peut ensuite éditer son profil et saisir ses informations académiques.

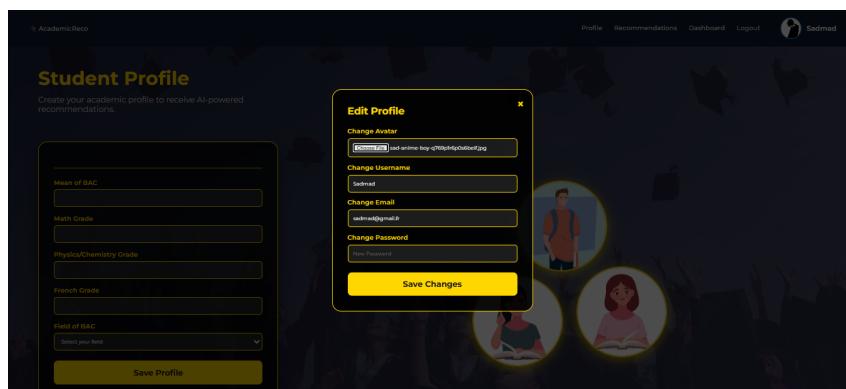


FIGURE 11 – Édition du profil étudiant

8.2 Tableau de Bord et Recommandations

Une fois les données saisies, le dashboard affiche une synthèse des performances et les recommandations générées par l'IA.

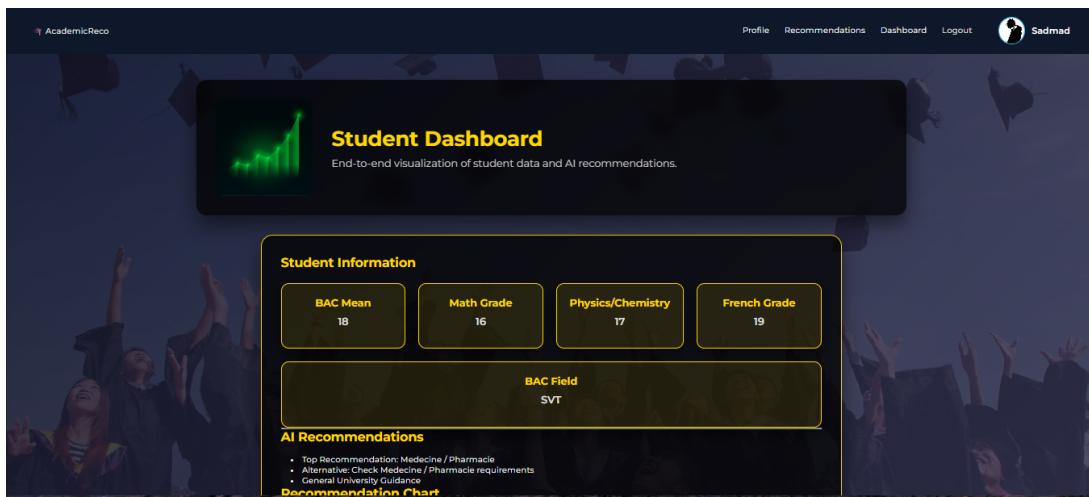


FIGURE 12 – Dashboard Étudiant : Vue d'ensemble

L'IA propose une orientation principale ainsi qu'une alternative, visualisées sous forme de graphiques interactifs.

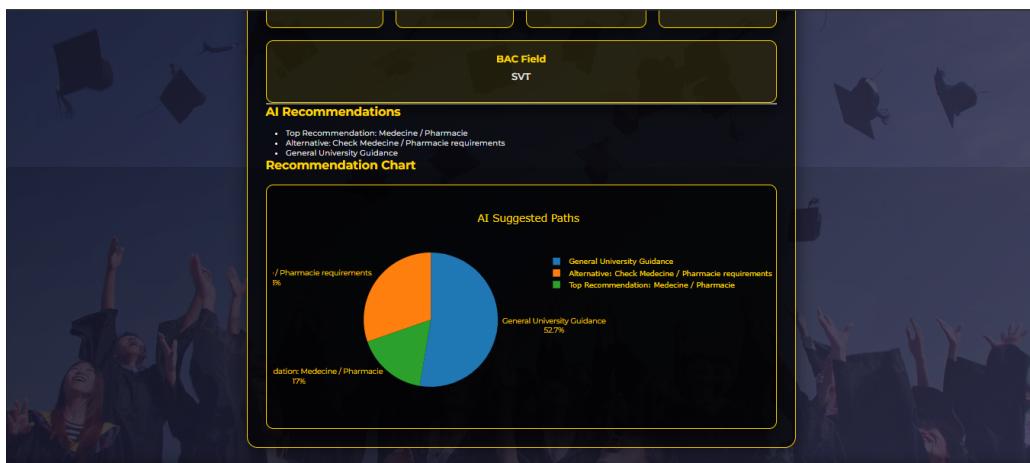


FIGURE 13 – Visualisation graphique des recommandations

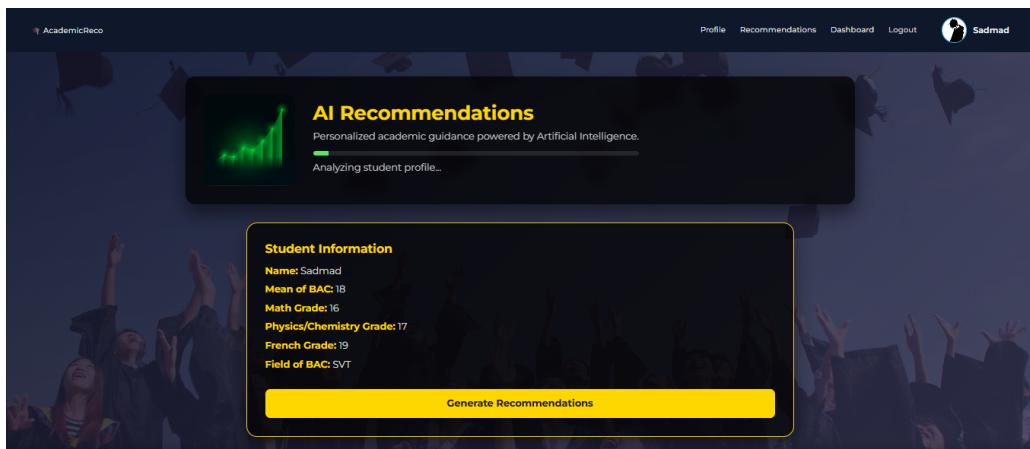


FIGURE 14 – Détail des recommandations IA

8.3 Espace Administration

L'administrateur dispose d'une vue dédiée pour comparer la précision des différents modèles de Machine Learning testés.



FIGURE 15 – Dashboard Admin : Comparaison des performances des modèles

9 Problèmes Rencontrés et Solutions

Durant la phase de développement et d'intégration, nous avons rencontré un défi technique lié à l'intégration du modèle.

9.1 Problème : Incompatibilité des Types de Données

Lors des premiers tests, le modèle générait une erreur car il attendait des valeurs numériques alors que l'interface envoyait des chaînes de caractères ("Oui"/"Non").

¹ `ValueError: could not convert string to float: 'Oui'`

Listing 1 – Erreur rencontrée lors de la prédiction

9.2 Solution Mise en œuvre

Nous avons implémenté une couche de transformation (mapping) dans le fichier `app.py` pour convertir dynamiquement les entrées avant de les soumettre au modèle.

¹ `# Conversion des entrees 'Oui'/'Non' en 1/0 pour le modele`
² `schoolsup = st.selectbox("Support ecole", ["Oui", "Non"])`
³ `schoolsup_val = 1 if schoolsup == "Oui" else 0`

```
4 # ... Logique appliquée aux autres variables
```

Listing 2 – Correction dans app.py (Mapping des valeurs)

10 Installation et Déploiement

Le déploiement de l'application se fait via une installation standard de l'environnement Python.

10.1 Prérequis

- Python 3.8 ou version supérieure
- Gestionnaire de paquets pip
- Navigateur web récent

10.2 Procédure de Lancement

1. Clonage du dépôt :

```
1 git clone https://github.com/OthmanSALAHIBI/  
    filiereRecommendationMorocco.git  
2 cd filiereRecommendationMorocco  
3
```

2. Installation des dépendances : Il est recommandé d'utiliser un environnement virtuel pour isoler les bibliothèques du projet.

```
1 pip install -r requirements.txt  
2
```

3. Lancement de l'application : Démarrage du serveur backend et de l'interface utilisateur.

```
1 # Remplacez 'app.py' par votre fichier principal (ex: main.py,  
    manage.py)  
2 python app.py  
3
```

4. Accès : Une fois le serveur lancé, l'application est accessible via l'URL : <http://localhost:5000> (ou le port indiqué dans votre terminal).

11 Conclusion

Ce projet a permis de mettre en œuvre une solution complète d'aide à la décision académique. En combinant une gestion de projet Agile rigoureuse sous Jira et des technologies modernes (Python, Scikit-learn, Docker), l'équipe a livré un produit fonctionnel, testé et documenté, répondant aux besoins des étudiants marocains.