



ECOLE MAROCAINE DES  
SCIENCES DE L'INGENIEUR  
Membre de  
HONORIS UNITED UNIVERSITIES

## RAPPORT DE PROJET DE FIN D'ANNÉE

*4<sup>ème</sup> Année en Ingénierie Informatique et Réseaux  
Option MIAGE*

### Système Intelligent de Prédiction et Recommandation d'Orientation Académique

---

**Réalisé par**

CHARTY MALAK

BEN ABDELLAH RANIA

**Encadrant**

FILALI MOHANAD

**Année universitaire**

2025/2026

# Table des matières

|  |           |
|--|-----------|
| Liste des figures . . . . .  | 3         |
| Remerciements . . . . .  | 4         |
| Résumé . . . . .   | 5         |
| Abstract . . . . .   | 6         |
| <br>   |           |
| <b>Introduction Générale</b>   | <b>7</b>  |
| <br>   |           |
| <b>Chapitre I — Présentation du projet</b>   | <b>8</b>  |
| 1.1 Introduction . . . . .   | 9         |
| 1.2 Contexte et sujet du projet . . . . .  | 9         |
| 1.3 Problématique . . . . .  | 9         |
| 1.4 Objectifs du projet . . . . .  | 10        |
| 1.5 User Stories . . . . .   | 11        |
| 1.6 Périmètre fonctionnel du projet . . . . .  | 12        |
| Conclusion . . . . .   | 12        |
| <br>   |           |
| <b>Chapitre II — Méthodologie de travail</b>   | <b>13</b> |
| 2.1 Méthodologie Agile . . . . .   | 14        |
| 2.2 Cadre Scrum . . . . .  | 15        |
| 2.3 Les artefacts Scrum . . . . .  | 15        |
| 2.4 Les cérémonies Scrum . . . . .   | 15        |
| 2.5 Organisation des sprints et planification Scrum . . . . .                        | 16        |
| 2.5.1 Sprint 1 — Saisie des données et prédition initiale . . . . .                  | 16        |
| 2.5.2 Sprint 2 — Fonctionnalités avancées et analyse des résultats . . . . .         | 17        |
| 2.5.3 Sprint 3 — Gestion et évolution du modèle d'apprentissage automatique          | 18        |
| <br>   |           |
| <b>Chapitre III — Conception du système d'orientation académique</b>                 | <b>19</b> |
| 3.1 Diagramme de cas d'utilisation . . . . .   | 20        |
| 3.2 Diagramme de séquence . . . . .  | 21        |
| 3.3 Diagramme d'activités . . . . .  | 23        |
| <br>   |           |
| <b>Chapitre IV — Architecture du Système et Pipeline d'Apprentissage Automatique</b> | <b>25</b> |
| 4.1 Architecture générale du système . . . . .                                       | 26        |
| 4.1.1 Couche Présentation – Interface Web Streamlit . . . . .                        | 27        |

|   |  |           |
|---|--|-----------|
| 4.1.2   | Couche Logique Métier – Backend FastAPI . . . . .        | 27        |
| 4.1.3   | Couche Données – MySQL et Stockage de fichiers . . . . . | 28        |
| 4.2   | Justification du choix de l'architecture . . . . .       | 28        |
| 4.3   | Pipeline d'apprentissage automatique (IA) . . . . .      | 30        |
| <b>Chapitre V — Réalisation et Implémentation</b> |  | <b>32</b> |
| Introduction . . . . .                            |  | 33        |
| 5.1   | Organisation générale du projet . . . . .                | 33        |
| 5.1.1   | Vue globale de l'arborescence . . . . .                  | 33        |
| 5.1.2   | Organisation du backend (FastAPI) . . . . .              | 34        |
| 5.1.3   | Organisation du frontend (Streamlit) . . . . .           | 35        |
| 5.1.4   | Organisation du module IA et des modèles . . . . .       | 35        |
| 5.1.5   | Avantages de cette organisation . . . . .                | 36        |
| 5.1.6   | Interfaces et écrans de l'application . . . . .          | 36        |
| Conclusion du chapitre . . . . .                  |  | 40        |
| <b>Chapitre VI — Conclusion Générale</b>          |  | <b>41</b> |

# Table des figures

|      |  |    |
|------|--|----|
| 2.1  | Planning et suivi du Sprint 1 dans Jira . . . . .                            | 17 |
| 2.2  | Planning et suivi du Sprint 2 dans Jira . . . . .                            | 17 |
| 2.3  | Planning et suivi du Sprint 3 dans Jira . . . . .                            | 18 |
| 3.1  | Diagramme de cas d'utilisation du système d'orientation académique . . . . . | 20 |
| 3.2  | Diagramme de séquence du processus de prédition d'orientation . . . . .      | 21 |
| 3.3  | Diagramme d'activités du processus de prédition d'orientation . . . . .      | 23 |
| 4.1  | Architecture générale du système d'orientation académique . . . . .          | 26 |
| 4.2  | Logos : Streamlit — Python . . . . .   | 27 |
| 4.3  | Logos : FastAPI — Python . . . . .   | 27 |
| 4.4  | Logo :MySQL . . . . .  | 28 |
| 4.5  | Pipeline global du processus d'apprentissage automatique . . . . .           | 30 |
| 5.1  | Vue globale de l'arborescence du projet Orientation-IA . . . . .             | 33 |
| 5.2  | Organisation interne du backend FastAPI . . . . .                            | 34 |
| 5.3  | Organisation du frontend Streamlit . . . . .                                 | 35 |
| 5.4  | Organisation du module ia et des modèles . . . . .                           | 35 |
| 5.5  | Interface principale de saisie des informations étudiantes . . . . .         | 36 |
| 5.6  | Saisie des informations personnelles de l'étudiant . . . . .                 | 37 |
| 5.7  | Saisie des notes scolaires et des scores comportementaux . . . . .           | 37 |
| 5.8  | Affichage des résultats et recommandations d'orientation . . . . .           | 38 |
| 5.9  | Dashboard du conseiller d'orientation . . . . .                              | 39 |
| 5.10 | Historique des recommandations . . . . .                                     | 39 |

## Remerciements

Nous tenons à exprimer nos sincères remerciements à notre encadrant pédagogique, **M. Filali Mohanad**, pour son accompagnement, ses conseils et son suivi tout au long de ce projet.

Nous remercions également l'**EMSI** et l'ensemble du corps professoral pour la qualité de la formation dispensée, ainsi que pour les moyens mis à notre disposition durant notre parcours académique.

Enfin, nous adressons nos remerciements à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet.

## Résumé

L'orientation académique représente une étape déterminante pour les étudiants post-bac, mais elle reste souvent influencée par des choix subjectifs et un manque d'informations exploitables. Ce projet propose une solution d'aide à la décision basée sur l'Intelligence Artificielle, capable de **prédir** et **recommander** une filière adaptée au profil d'un étudiant à partir de données scolaires et personnelles (notes, type de baccalauréat, centres d'intérêt, indicateurs psychométriques, etc.).

La solution, nommée *Orientation-IA*, s'appuie sur une architecture en trois couches : une interface web interactive développée avec **Streamlit**, une API métier en **FastAPI** assurant le prétraitement, l'inférence et la gestion des données, ainsi qu'une couche de persistance basée sur **MySQL** et un stockage de fichiers pour les modèles et rapports. Un pipeline d'apprentissage automatique complet a été mis en place (préparation des données, prétraitement, entraînement, comparaison des modèles, prédiction et explication). L'application offre également des visualisations, un historique des recommandations et une génération automatique de rapport PDF afin d'améliorer la compréhension et la traçabilité des résultats.

## Abstract

Academic orientation is a key step for post-baccalaureate students, yet it is often driven by subjective choices and limited access to actionable information. This project proposes an Artificial Intelligence-based decision support system that **predicts** and **recommends** the most suitable academic track for a student by analyzing academic and personal data (grades, baccalaureate type, interests, and psychometric indicators).

The proposed solution, called *Orientation-IA*, is built on a three-tier architecture : an interactive web interface developed with **Streamlit**, a **FastAPI** backend handling data validation, preprocessing, inference, and services, and a data layer combining **MySQL** with file storage for trained models and generated reports. A complete machine learning pipeline was implemented (data preparation, preprocessing, training, model comparison, prediction, and explanation). The application also provides visual analytics, recommendation history, and automated PDF report generation to improve interpretability and traceability of results.

# Introduction Générale

L'orientation académique constitue une étape déterminante dans le parcours des étudiants, car elle conditionne leur réussite académique et leur insertion professionnelle. Cependant, ce choix demeure souvent complexe en raison de la diversité des filières, du manque d'informations personnalisées et de l'accompagnement limité dont disposent les étudiants. Face à ces difficultés, il devient nécessaire de proposer des solutions innovantes capables d'aider les étudiants à prendre des décisions éclairées et adaptées à leur profil.

Dans ce contexte, l'Intelligence Artificielle et les techniques d'apprentissage automatique offrent de nouvelles perspectives pour analyser les données scolaires et comportementales des étudiants afin de fournir des recommandations personnalisées. Ce projet s'inscrit dans cette démarche en proposant la conception et la réalisation d'un système intelligent d'aide à l'orientation académique, intitulé *Orientation-IA*.

L'objectif principal de ce travail est de développer une application capable de recommander des filières académiques pertinentes en se basant sur les performances scolaires, les compétences comportementales et les informations personnelles des étudiants. Pour atteindre cet objectif, une architecture modulaire combinant une interface web interactive, un backend performant et un modèle d'apprentissage automatique a été mise en place.

Ce rapport présente l'ensemble des étapes de réalisation du projet, depuis l'analyse du contexte et la définition des besoins, jusqu'à la conception, l'implémentation et l'évaluation du système proposé.

---

## **CHAPITRE I :**

Présentation du projet

---

## 1.1 Introduction

L'orientation académique constitue une étape décisive dans le parcours d'un étudiant. Pourtant, ce choix est souvent effectué de manière subjective, influencé par des facteurs non mesurables ou par un manque d'informations fiables sur les filières et les compétences requises. Dans ce contexte, l'Intelligence Artificielle offre un moyen innovant de guider les étudiants vers des choix plus éclairés et adaptés à leur profil.

Ce premier chapitre présente le contexte général du projet, sa problématique, les objectifs poursuivis ainsi que les user stories qui définissent les attentes fonctionnelles des différents acteurs impliqués.

## 1.2 Contexte et sujet du projet

Dans un contexte où les étudiants post-bac rencontrent des difficultés à choisir une filière adaptée à leurs compétences, leurs résultats scolaires et leurs préférences personnelles, les établissements cherchent à proposer des solutions d'orientation plus personnalisées.

Ce projet vise à développer un système d'Intelligence Artificielle capable de prédire la filière académique la plus adaptée à un étudiant, en analysant ses données personnelles (notes, type de baccalauréat, centres d'intérêt, niveau de stress, etc.).

L'application repose sur un modèle de Machine Learning entraîné sur un dataset représentatif, et met à disposition une interface web conçue avec **Streamlit**, permettant à l'étudiant ainsi qu'au conseiller d'orientation d'interagir facilement avec le système.

## 1.3 Problématique

La problématique principale du projet s'articule autour de la question suivante :

*Comment proposer aux étudiants une recommandation d'orientation fiable, personnalisée et justifiable, basée sur une analyse intelligente de leurs données personnelles et scolaires ?*

Cette problématique met en évidence plusieurs limites observées dans les systèmes d'orientation traditionnels :

- Absence d'un outil automatisé et objectif d'aide à la décision.
- Manque de prise en compte des compétences non scolaires (stress, communication, créativité, motivation, etc.).

- Difficulté pour les étudiants à interpréter leurs performances ou à identifier clairement leurs points forts.
- Charge importante pour les conseillers d'orientation qui doivent analyser de grands volumes d'informations hétérogènes.

L'Intelligence Artificielle permet de répondre à ces défis en offrant un système prédictif plus précis, explicable et accessible via une interface web dédiée.

## 1.4 Objectifs du projet

### Objectif général

L'objectif principal de ce projet est de développer un système d'aide à l'orientation permettant de recommander une filière d'études adaptée à un étudiant, en se basant sur un ensemble de données personnelles, scolaires et comportementales.

### Objectifs spécifiques

Pour atteindre cet objectif général, plusieurs objectifs spécifiques ont été définis :

- Concevoir un dataset contenant les informations pertinentes (notes, type de baccalauréat, heures d'étude, niveau de stress, centres d'intérêt, etc.).
- Entraîner un modèle d'apprentissage automatique performant et fiable.
- Fournir à l'utilisateur une interface web simple et intuitive, développée avec **Streamlit**, pour interagir avec le système.
- Générer un rapport PDF intégrant les résultats, les explications de la prédiction et les recommandations proposées.
- Permettre au conseiller d'orientation de consulter les tendances globales ainsi que les résultats individuels des étudiants.
- Offrir à l'administrateur IA un accès spécifique pour gérer le dataset, contrôler la qualité des données et entraîner à nouveau le modèle.

## 1.5 User Stories

Les user stories décrivent les besoins fonctionnels du système selon une approche centrée utilisateur. Elles sont formulées suivant la syntaxe standard : « *En tant qu'utilisateur, je veux ... afin de ...* ».

- **US1** : En tant qu'utilisateur, je veux saisir mes informations personnelles afin d'obtenir une orientation académique personnalisée.
- **US2** : En tant qu'utilisateur, je veux consulter plusieurs filières proposées avec un pourcentage de compatibilité afin de comparer les options d'orientation.
- **US3** : En tant qu'utilisateur, je veux visualiser mes points forts (logique, communication, créativité, etc.) afin de mieux comprendre mon profil.
- **US4** : En tant qu'utilisateur, je veux comprendre les critères ayant influencé la prédition afin d'interpréter correctement le résultat obtenu.
- **US5** : En tant qu'utilisateur, je veux consulter les orientations générées par le système afin d'analyser les recommandations proposées.
- **US6** : En tant qu'utilisateur, je veux filtrer les profils selon les orientations proposées afin de faciliter la recherche et l'analyse des résultats.
- **US7** : En tant qu'utilisateur, je veux consulter les tendances globales des orientations afin d'identifier les filières les plus recommandées.
- **US8** : En tant qu'utilisateur, je veux importer ou mettre à jour le dataset afin d'améliorer la qualité des données utilisées par le système.
- **US9** : En tant qu'utilisateur, je veux entraîner plusieurs modèles et comparer leurs performances afin de sélectionner le modèle le plus pertinent.
- **US10** : En tant qu'utilisateur, je veux sauvegarder le meilleur modèle ainsi que ses métriques afin de l'utiliser pour les prédictions futures.
- **US11** : En tant qu'utilisateur, je veux réentraîner le modèle en cas de mise à jour du dataset afin de maintenir la performance du système.
- **US12** : En tant qu'utilisateur, je veux accéder à une interface simple et intuitive via Streamlit afin d'interagir facilement avec le système.
- **US13** : En tant qu'utilisateur, je veux soumettre mes données et obtenir une prédition immédiate afin de recevoir rapidement une orientation.
- **US14** : En tant qu'utilisateur, je veux visualiser les résultats sous forme de graphiques afin de mieux interpréter les recommandations.
- **US15** : En tant qu'utilisateur, je veux télécharger un rapport PDF contenant l'analyse complète afin de conserver une trace des résultats.

## 1.6 Périmètre fonctionnel du projet

### Fonctionnalités incluses

- Formulaire complet de collecte des données étudiantes.
- Prédiction IA de la filière académique.
- Indicateurs psychométriques : logique, créativité, communication, motivation, etc.
- Génération automatique d'un rapport PDF détaillé.
- Consultation des résultats avec filtres.
- Gestion du dataset et entraînement du modèle.

## Conclusion

Ce premier chapitre a présenté le cadre général du projet, la problématique à résoudre ainsi que les objectifs fonctionnels et techniques. Il a également introduit les user stories décrivant les fonctionnalités attendues du système du point de vue de l'utilisateur.

Ces éléments permettent de comprendre l'intérêt du système proposé et constituent la base nécessaire pour aborder le chapitre suivant consacré à l'analyse et à la conception du système.

Le Chapitre 2 détaillera les besoins fonctionnels et non fonctionnels, et introduira les différents diagrammes UML permettant de modéliser le fonctionnement de l'application.

---

## **CHAPITRE II :**

Méthodologie de travail

---

# Introduction

Ce chapitre présente la méthodologie de travail adoptée pour la réalisation du projet de prédiction d'orientation étudiante. Dans un contexte où les besoins peuvent évoluer rapidement et où les modèles d'Intelligence Artificielle nécessitent des ajustements réguliers, il est indispensable de s'appuyer sur une approche flexible et itérative.

Nous avons ainsi choisi la méthodologie Agile, appuyée par le cadre Scrum, afin d'assurer une organisation structurée du travail tout en favorisant l'adaptabilité, la collaboration et l'amélioration continue. Ce chapitre détaille les principes Agile appliqués, les rôles et artefacts Scrum mobilisés, ainsi que la planification des différents sprints qui ont guidé l'avancement du projet.

L'objectif est d'exposer clairement la démarche méthodologique qui a permis de transformer progressivement les besoins identifiés en livrables fonctionnels et cohérents.

## 2.1 Méthodologie Agile

Dans le cadre de ce projet, nous avons adopté la méthodologie Agile, un cadre de travail moderne privilégiant la flexibilité, l'adaptabilité et l'amélioration continue. Contrairement aux approches classiques comme le cycle en V, où les étapes sont rigides et définies à l'avance, l'Agile repose sur des cycles itératifs courts appelés *sprints*.

Cette méthodologie permet notamment :

- une meilleure réactivité face aux changements ;
- une communication continue au sein de l'équipe ;
- une amélioration progressive basée sur les retours ;
- une livraison fréquente de fonctionnalités utilisables.

## Principes fondamentaux

Les principes appliqués dans ce projet s'inspirent du **Manifeste Agile**, et reposent sur :

- la collaboration plutôt que des processus rigides ;
- l'adaptation au changement plutôt qu'un plan figé ;
- la livraison fréquente plutôt que des cycles longs ;
- la satisfaction du client par une amélioration continue.

Cette approche est particulièrement adaptée aux projets d'Intelligence Artificielle où les données évoluent et où les modèles doivent être ajustés régulièrement.

## 2.2 Cadre Scrum

Pour structurer le travail Agile, nous avons utilisé le framework **Scrum**, très répandu dans les projets informatiques. Scrum organise le projet en sprints permettant de livrer un incrément fonctionnel à chaque cycle.

### Les rôles Scrum

- 1. Product Owner (PO)** Définit les besoins fonctionnels, priorise le backlog et garantit la valeur ajoutée du produit. Dans ce projet, ce rôle est assuré par l'encadrant pédagogique.
- 2. Scrum Master** Assure le respect du cadre Scrum, élimine les obstacles et facilite les cérémonies. Ce rôle a été partagé de manière collaborative au sein de l'équipe.
- 3. Équipe de développement** Responsable du développement du modèle IA, de l'interface Streamlit et du traitement des données.

## 2.3 Les artefacts Scrum

- 1. Product Backlog** Il s'agit de la liste complète des besoins du projet :
  - user stories (prédiction, visualisations, rapport PDF);
  - exigences fonctionnelles (collecte des données, prédiction, interface Streamlit);
  - exigences non fonctionnelles (performance du modèle, rapidité).
- 2. Sprint Backlog** Chaque sprint sélectionne une partie du Product Backlog :
  - développement du modèle IA ;
  - interface utilisateur ;
  - visualisations et rapport PDF.
- 3. Incrément** Livrable fonctionnel produit à la fin d'un sprint : dataset nettoyé, première version du modèle, interface initiale, etc.

## 2.4 Les cérémonies Scrum

- 1. Sprint Planning** Réunion planifiant les objectifs du sprint et les user stories à réaliser.

**2. Daily Scrum** Réunion quotidienne de 10–15 minutes permettant de synchroniser l'équipe autour de :

- ce qui a été fait;
- ce qui sera fait;
- les obstacles rencontrés.

**3. Sprint Review** Présentation de l'incrément à la fin du sprint (modèle entraîné, interface fonctionnelle, analyses exploratoires...).

**4. Sprint Retrospective** Réunion permettant d'identifier :

- ce qui a bien fonctionné;
- ce qui doit être amélioré;
- les actions pour le sprint suivant.

## 2.5 Organisation des sprints et planification Scrum

Dans le cadre de la méthodologie Scrum adoptée pour ce projet, le travail a été structuré en trois sprints principaux. Chaque sprint regroupe un ensemble cohérent de *user stories*, organisées selon leur priorité, leur dépendance et leur importance pour la construction progressive du système de prédiction et de recommandation académique.

Les sprints ont été planifiés dans **Jira**, ce qui a permis :

- d'avoir une vision claire de l'avancement global;
- de répartir la charge de travail de manière équilibrée;
- d'assurer la traçabilité des décisions et des ajustements réalisés à chaque cycle.

### 2.5.1 Sprint 1 — Saisie des données et prédiction initiale

Le premier sprint a été consacré à la construction du socle fonctionnel du projet. L'objectif principal était de permettre la saisie des informations nécessaires et l'obtention d'une première prédiction générée par un modèle d'apprentissage automatique.

Durant ce sprint, nous avons mis en place :

- l'interface initiale permettant la saisie des données utilisateur;
- la génération d'une prédiction immédiate à partir du modèle IA;
- l'affichage des filières avec un pourcentage de compatibilité;
- l'entraînement et la comparaison d'un premier ensemble de modèles de Machine Learning.

| Tableau Sprint 1         |           | Add dates  | (6 work items) | 0     | 0 | 0 | Start sprint | ... |
|--------------------------|-----------|--|----------------|-------|---|---|--------------|-----|
| <input type="checkbox"/> | POEI20-14 | US1 – Saisie des informations étudiant           |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-15 | US2 – Proposition de filières avec pourcentage   |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-17 | US4 – Explication des critères de recommandation |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-25 | US12 – Accéder à une interface simple            |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-26 | US13 – Obtenir une prédiction immédiate          |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-22 | US9 – Entraîner et comparer les modèles          |                | TO DO | ▼ | - |              |     |

+ Create

**FIGURE 2.1** – Planning et suivi du Sprint 1 dans Jira

Ce sprint a permis d’obtenir un *MVP* (Minimum Viable Product) fonctionnel, servant de base aux sprints suivants.

## 2.5.2 Sprint 2 — Fonctionnalités avancées et analyse des résultats

Le second sprint a visé l’enrichissement fonctionnel du système en ajoutant des fonctionnalités avancées ainsi que des outils d’analyse visuelle.

| Tableau Sprint 2         |           | Add dates                                   | (5 work items) | 0      | 0     | 0 | Start sprint | ... |
|--------------------------|-----------|---|----------------|--------|-------|---|--------------|-----|
| <input type="checkbox"/> | POEI20-18 | US5 – Consulter les résultats des étudiants |                | TO DO  | ▼     | - |              |     |
| <input type="checkbox"/> | POEI20-16 | US3 – Visualisation des points forts        |                | TO DO  | ▼     | - |              |     |
| <input type="checkbox"/> | POEI20-19 | US6 – Filtrer les étudiants                 | ▶              | + Epic | TO DO | ▼ | -            | ... |
| <input type="checkbox"/> | POEI20-28 | US15 – Télécharger un rapport PDF           |                | TO DO  | ▼     | - |              |     |
| <input type="checkbox"/> | POEI20-27 | US14 – Visualiser les graphiques            |                | TO DO  | ▼     | - |              |     |

+ Create

**FIGURE 2.2** – Planning et suivi du Sprint 2 dans Jira

Les principaux travaux réalisés durant ce sprint sont :

- la consultation des prédictions générées ;
- la visualisation des points forts à travers des graphiques explicatifs ;
- la mise en place de filtres pour organiser et rechercher les résultats ;
- l’intégration de la génération d’un rapport PDF ;
- l’ajout de visualisations globales sur les données et les orientations proposées.

Ce sprint a apporté une dimension analytique au système et a amélioré la lisibilité et l’exploitation des résultats.

### 2.5.3 Sprint 3 — Gestion et évolution du modèle d'apprentissage automatique

Le troisième sprint s'est concentré sur les fonctionnalités liées à la gestion et à l'évolution continue du modèle d'Intelligence Artificielle.

| Tableau Sprint 3         |           | Add dates                                  | (4 work items) | 0     | 0 | 0 | Start sprint | ... |
|--------------------------|-----------|--|----------------|-------|---|---|--------------|-----|
| <input type="checkbox"/> | POEI20-21 | US8 – Importer ou mettre à jour le dataset |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-23 | US10 – Suivre et sauvegarder les métriques |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-24 | US11 – Réentraînement automatique          |                | TO DO | ▼ | - |              |     |
| <input type="checkbox"/> | POEI20-20 | US7 – Ajouter un retour qualitatif         |                | TO DO | ▼ | - |              |     |

FIGURE 2.3 – Planning et suivi du Sprint 3 dans Jira

Ce sprint a introduit :

- l'importation et la mise à jour du dataset ;
- le suivi et la sauvegarde des métriques de performance des modèles ;
- le réentraînement du modèle en cas de nouvelles données ;
- l'ajout d'un retour qualitatif pour améliorer la qualité globale du système.

Ce sprint finalise l'écosystème du projet en rendant la solution évolutive, supervisée et adaptable aux changements des données et des besoins.

## Conclusion du chapitre

En résumé, la méthodologie Agile associée au cadre Scrum a permis de structurer le projet de manière flexible, itérative et efficace. L'organisation en sprints, soutenue par les différentes cérémonies Scrum, a facilité la planification, le suivi et l'amélioration continue du système.

Grâce à cette approche, chaque incrément livré a apporté une évolution tangible : première prédiction, outils d'analyse des résultats, puis supervision et évolution continue du modèle d'Intelligence Artificielle. La méthode adoptée s'est ainsi révélée parfaitement adaptée aux exigences d'un projet d'IA nécessitant des ajustements fréquents et une adaptation constante aux données.

Ce cadre méthodologique constitue donc une base solide ayant guidé l'avancement du projet jusqu'aux étapes de conception et de développement présentées dans le chapitre suivant.

---

### **CHAPITRE III :**

Conception du système d'orientation académique

---

# Introduction du chapitre

Ce chapitre est consacré à la conception du système d'orientation académique. Il vise à présenter les modèles UML permettant de décrire le fonctionnement interne de l'application ainsi que les interactions entre les différents acteurs et composants. Les diagrammes réalisés permettent de passer d'une vision fonctionnelle (user stories, besoins utilisateurs) à une vision structurelle et dynamique, servant de base à la mise en œuvre technique.

À travers les diagrammes de cas d'utilisation, d'activités et de séquence, nous mettons en évidence la logique du processus de prédiction, les flux de données, ainsi que la coordination entre les modules constituant le système.

## 3.1 Diagramme de cas d'utilisation

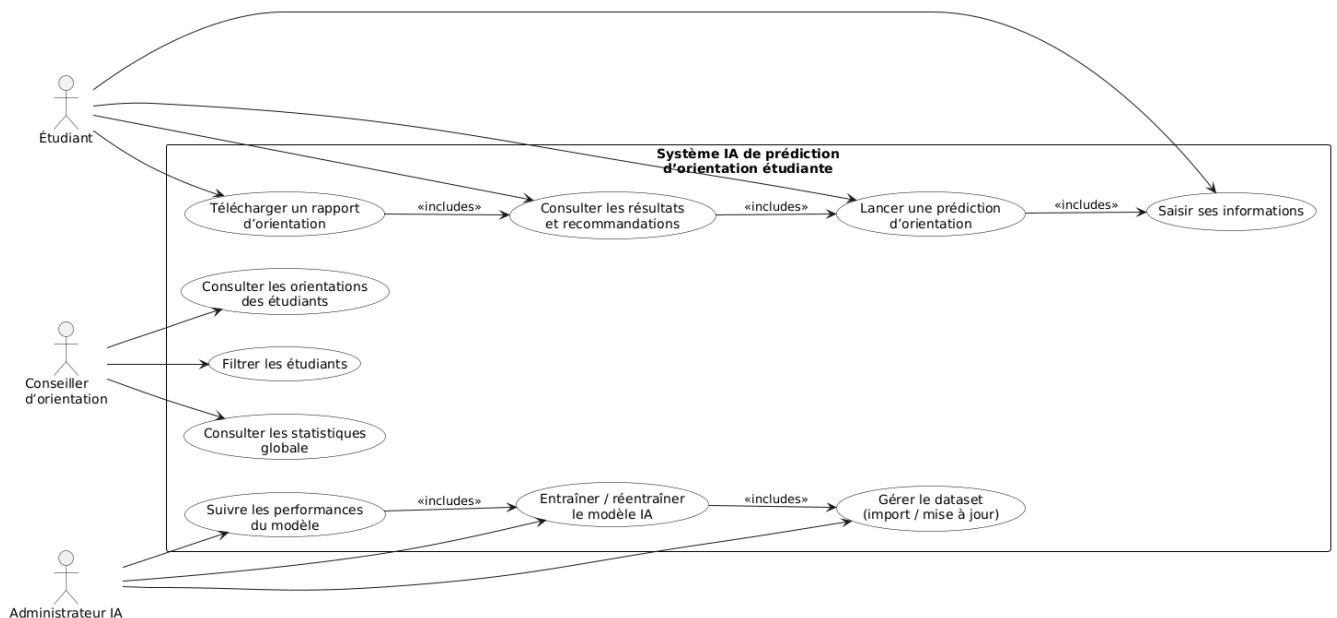


FIGURE 3.1 – Diagramme de cas d'utilisation du système d'orientation académique

## =>Description du diagramme de cas d'utilisation

Le diagramme de cas d'utilisation ci-dessus présente les principales interactions entre les acteurs du système d'orientation académique et les fonctionnalités offertes par la plateforme.

Trois acteurs interviennent :

- **Étudiant** : il saisit ses informations pour obtenir une prédiction d'orientation. Il peut consulter les résultats et recommandations générées par l'IA, lancer une prédiction, et télécharger un rapport d'orientation.

- **Conseiller d'orientation** : il accède aux orientations des étudiants, peut filtrer les profils, et consulter les statistiques globales permettant d'analyser les tendances d'orientation.
- **Administrateur IA** : il supervise la qualité du modèle. Il gère le dataset (import et mise à jour), entraîne ou réentraîne le modèle IA, et suit ses performances.

Les relations *includes* indiquent les dépendances entre les cas d'utilisation :

- pour obtenir une prédiction, l'étudiant doit d'abord saisir ses informations ;
- la consultation des résultats inclut la possibilité de télécharger un rapport ;
- l'entraînement du modèle dépend de la gestion du dataset.

Ce diagramme permet ainsi d'obtenir une vue globale sur les fonctionnalités principales du système, ainsi que sur le rôle de chaque acteur dans le processus complet d'orientation académique.

## 3.2 Diagramme de séquence

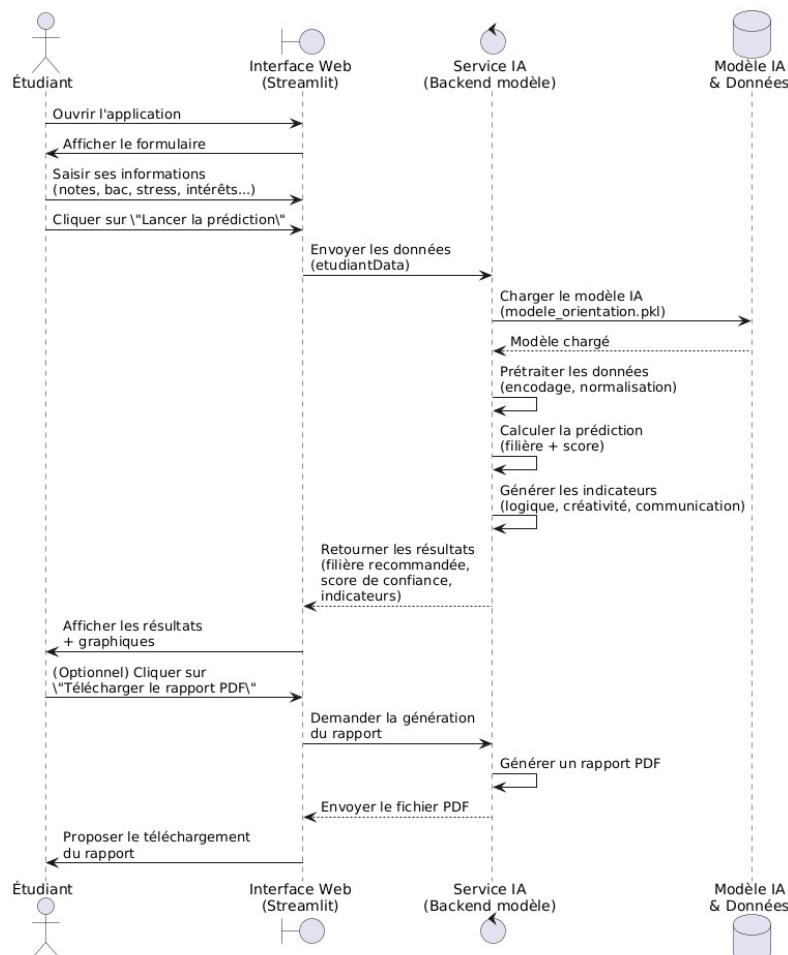


FIGURE 3.2 – Diagramme de séquence du processus de prédiction d'orientation

## =>Description du diagramme de séquence

Le diagramme de séquence ci-dessus décrit l'enchaînement des interactions entre les principaux composants du système lors du processus de prédiction d'orientation académique.

Le scénario débute lorsque l'étudiant accède à l'application et remplit le formulaire depuis l'interface web développée en Streamlit. Après la saisie, les informations sont transmises au service IA chargé d'exécuter le modèle de prédiction.

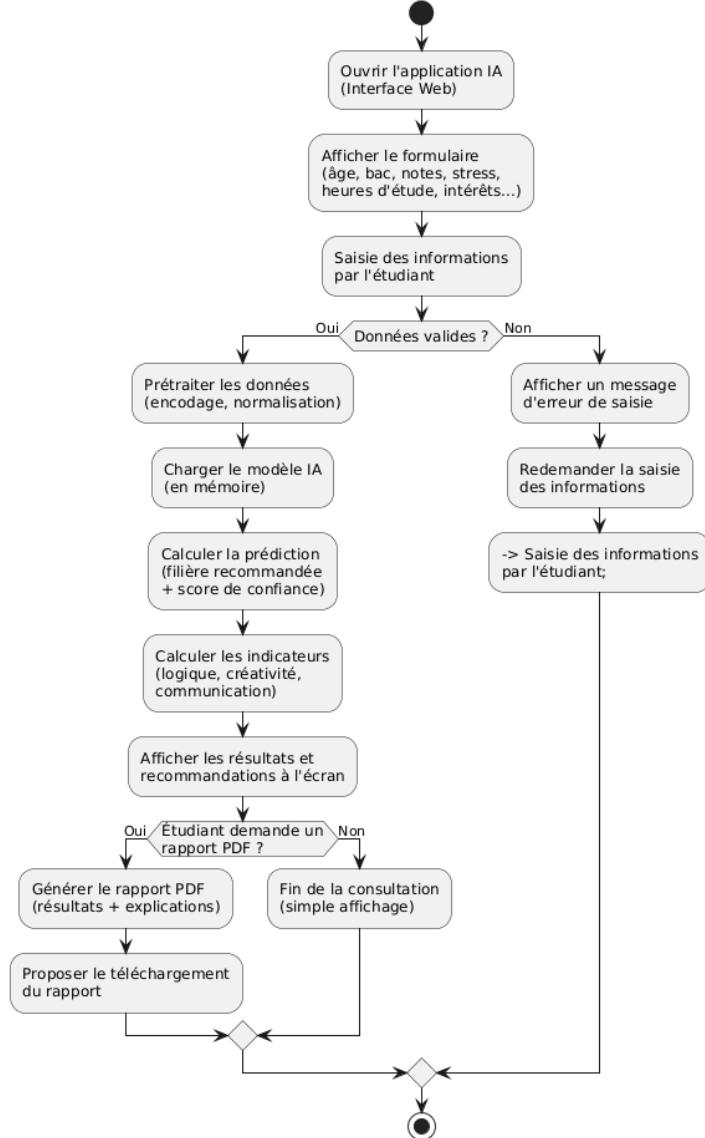
Le service IA commence par charger le modèle d'apprentissage automatique, puis applique les opérations de prétraitement, notamment l'encodage et la normalisation des données. Il calcule ensuite la prédiction, constituée d'une filière recommandée accompagnée d'un score de confiance, ainsi que les indicateurs psychométriques permettant d'expliquer le résultat (créativité, logique, communication...).

Ces informations sont renvoyées à l'interface web, qui se charge de les afficher sous forme de textes, graphiques et recommandations personnalisées.

L'étudiant peut également demander la génération d'un rapport PDF. Dans ce cas, l'interface envoie une requête spécifique au service IA, qui génère automatiquement le fichier PDF avant de le renvoyer à l'utilisateur. L'application propose alors un lien permettant le téléchargement du document.

Ce diagramme met ainsi en évidence la coordination entre l'étudiant, l'interface Streamlit, le backend IA et le modèle de Machine Learning, en illustrant clairement la répartition des responsabilités et la dynamique d'échange entre les différentes composantes du système.

### 3.3 Diagramme d'activités



**FIGURE 3.3 – Diagramme d’activités du processus de prédition d’orientation**

### =>Description du diagramme d'activités

Ce diagramme d'activités illustre le déroulement complet du processus de prédition d'orientation académique depuis l'interface web jusqu'à l'affichage final des résultats.

Le processus débute lorsque l'étudiant ouvre l'application et accède au formulaire lui permettant de saisir ses informations personnelles, scolaires et psychométriques (âge, type de baccalauréat, notes, niveau de stress, centres d'intérêts, etc.). Une première vérification est effectuée pour garantir la validité des données saisies. En cas d'erreur, un message d'alerte est affiché et l'étudiant est invité à corriger ses informations.

Lorsque les données sont valides, elles passent par une phase de prétraitement comprenant l'encodage des variables et la normalisation des valeurs. Le modèle d'IA est ensuite chargé en mémoire afin de générer une prédiction accompagnée d'un score de confiance, ainsi que plusieurs indicateurs psychométriques utiles pour interpréter les résultats (logique, créativité, communication...).

Les résultats et recommandations personnalisées sont ensuite présentés à l'écran. L'étudiant peut choisir de générer un rapport PDF contenant une synthèse de la prédiction et des explications détaillées. Si cette option est sélectionnée, le document est généré automatiquement et mis à disposition en téléchargement. Dans le cas contraire, la consultation s'achève simplement sur l'affichage des résultats.

Ce diagramme met en évidence la logique complète du flux de traitement : de la saisie des informations à la génération éventuelle d'un document final, en passant par l'analyse effectuée par le modèle d'Intelligence Artificielle.

## Conclusion du chapitre

Ce chapitre a présenté la conception globale du système d'orientation académique, en s'appuyant sur les différents modèles UML nécessaires pour comprendre son fonctionnement interne. Les diagrammes de cas d'utilisation, d'activités et de séquence ont permis de décrire de manière claire :

- les interactions entre les acteurs et le système ;
- les étapes du processus de prédiction ;
- les échanges entre l'interface Streamlit, le service IA et le modèle d'apprentissage.

Cette phase de conception constitue une étape essentielle, car elle sert de base solide à la mise en œuvre technique qui sera détaillée dans le chapitre suivant. Grâce à ces modèles, la structure du système est clarifiée, les responsabilités sont bien identifiées et le développement peut être réalisé de manière cohérente et maîtrisée.

---

## **CHAPITRE IV :**

Architecture du Système et Pipeline d'Apprentissage Automatique

---

# Introduction

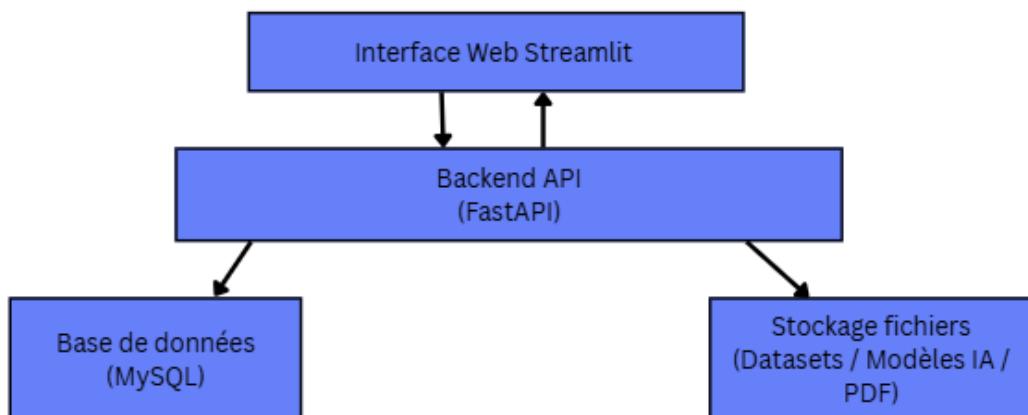
Ce chapitre présente l'architecture technique globale du système ainsi que le pipeline d'apprentissage automatique qui constitue le cœur du moteur de recommandation. L'objectif est d'expliquer la structure logicielle du projet, les technologies utilisées et la manière dont les différentes couches interagissent pour assurer le fonctionnement du système. Dans un second temps, le pipeline IA est détaillé afin de décrire les étapes successives permettant de transformer des données brutes en prédictions fiables et interprétables. Cette conception assure la cohérence entre l'architecture logicielle et les algorithmes d'intelligence artificielle utilisés pour générer les recommandations d'orientation académique.

## 4.1 Architecture générale du système

L'architecture adoptée repose sur une approche *trois tiers*, largement utilisée dans les systèmes web modernes. Elle se compose de trois couches principales :

- **Couche Présentation (Frontend)** : interface web Streamlit;
- **Couche Logique Métier (Backend)** : API développée avec FastAPI;
- **Couche Données** : base de données MySQL et espace de stockage des fichiers.

Chaque couche s'appuie sur des technologies adaptées aux exigences d'un système d'Intelligence Artificielle moderne.



**FIGURE 4.1 – Architecture générale du système d'orientation académique**

#### 4.1.1 Couche Présentation – Interface Web Streamlit

La couche présentation constitue le point d'entrée du système. Elle est développée avec **Streamlit**, un framework Python moderne spécialement conçu pour créer rapidement des interfaces web interactives destinées aux applications de Data Science et d'Intelligence Artificielle.

=> Streamlit se distingue par :

- son intégration native avec Python ;
- sa rapidité de développement ;
- sa capacité à afficher des visualisations interactives ;
- sa simplicité d'utilisation et sa prise en main intuitive.



**FIGURE 4.2** – Logos : Streamlit — Python

=> Cette couche permet aux utilisateurs d'interagir avec les fonctionnalités principales du système :

- saisie des informations étudiantes ;
- visualisation des résultats et graphiques explicatifs ;
- téléchargement d'un rapport PDF ;
- accès aux outils d'administration (dataset, ré entraînement du modèle).

#### 4.1.2 Couche Logique Métier – Backend FastAPI

La couche logique métier centralise l'ensemble des traitements applicatifs. Elle est implémentée en **FastAPI**, un framework Python performant et largement utilisé pour exposer des modèles d'Intelligence Artificielle.

=> FastAPI présente plusieurs avantages :

- très haute performance et traitement asynchrone ;
- documentation automatique (Swagger UI) ;
- compatibilité totale avec les librairies IA ;
- facilité de maintenance et d'évolution.



**FIGURE 4.3** – Logos : FastAPI — Python

=> Elle assure les fonctions essentielles :

- prétraitement et validation des données ;

- chargement du modèle IA et exécution des prédictions ;
- réentraînement du modèle ;
- mise à jour des métriques ;
- communication avec MySQL ;
- gestion des fichiers (datasets, modèles, PDF).

#### **4.1.3 Couche Données – MySQL et Stockage de fichiers**

La couche données regroupe deux espaces complémentaires :

- une base de données relationnelle MySQL,
- un espace de stockage dédié aux fichiers IA.

##### **-Base de données MySQL**

MySQL est utilisé pour stocker les données structurées :

- informations étudiants ;
- résultats de prédiction ;
- retours qualitatifs ;
- logs et historique d'exécution.



**FIGURE 4.4 – Logo :MySQL**

##### **-Stockage de fichiers**

Un espace séparé conserve les données non structurées :

- datasets (entraînement / mise à jour) ;
- modèles IA sauvegardés ;
- rapports PDF générés pour les utilisateurs.

Ce découpage garantit une gestion optimale entre données structurées et fichiers volumineux.

## **4.2 Justification du choix de l'architecture**

L'architecture adoptée a été choisie pour répondre aux exigences techniques, fonctionnelles et évolutives du projet. Elle permet de garantir un équilibre entre performance, modularité

et simplicité de déploiement.

- **Adaptation naturelle au cycle de vie d'un système IA**

Les projets d'Intelligence Artificielle nécessitent des mécanismes spécifiques tels que :

- le réentraînement régulier du modèle ;
- la gestion de différentes versions du modèle et des datasets ;
- le stockage de fichiers volumineux ;
- l'exécution de traitements parfois coûteux en ressources.

L'utilisation d'un backend **FastAPI** combiné à un stockage fichiers permet de gérer efficacement ces contraintes.

- **Souplesse d'intégration et évolutivité**

L'architecture en trois couches permet de modifier ou remplacer un composant sans affecter les autres :

- remplacement ou amélioration du modèle d'IA ;
- ajout d'une nouvelle interface (mobile, dashboard administrateur) ;
- migration vers une autre base de données.

- **Interopérabilité et facilité de déploiement**

Grâce à **FastAPI**, les fonctionnalités IA sont exposées en API REST, permettant :

- la connexion de n'importe quel frontend ;
- un déploiement distribué (Streamlit séparé du backend).

- **Expérience utilisateur optimale**

Le choix de **Streamlit** garantit une interface :

- fluide et réactive ;
- adaptée aux étudiants, conseillers et administrateurs IA ;
- intégrant facilement des visualisations.

- **Séparation nette des flux IA et métier**

Cette architecture garantit :

- une isolation de la logique IA ;
- des traitements optimisés côté serveur ;
- un frontend rapide même pour des prédictions lourdes.

## 4.3 Pipeline d'apprentissage automatique (IA)

# IA Pipeline

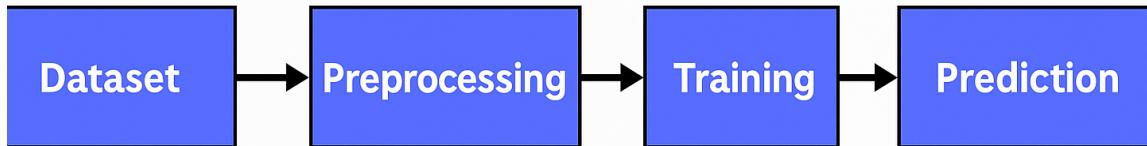


FIGURE 4.5 – Pipeline global du processus d'apprentissage automatique

Le pipeline d'apprentissage automatique constitue le cœur du système de prédiction et regroupe l'ensemble des étapes nécessaires pour transformer les données brutes en recommandations exploitables. Il se compose de quatre phases principales, décrites ci-dessous.

### • **Dataset — Collecte et préparation des données**

Cette première étape consiste à rassembler les informations nécessaires à l'entraînement du modèle, telles que :

- les notes des étudiants ;
- les compétences comportementales ;
- les centres d'intérêt ;
- les informations socio-éducatives.

Le dataset est ensuite nettoyé pour éliminer les incohérences, doublons ou valeurs manquantes afin d'assurer une base fiable pour l'apprentissage.

### • **Preprocessing — Prétraitement des données**

Le prétraitement transforme les données brutes en un format compatible avec les algorithmes d'Intelligence Artificielle. Cette étape inclut :

- l'encodage des variables catégorielles ;
- la normalisation ou standardisation des données numériques ;
- la séparation du dataset en ensembles d'entraînement et de test ;
- l'application de filtres ou corrections si nécessaire.

Cette phase est cruciale pour assurer la qualité des données et la performance du modèle

final.

### • Training — Entraînement du modèle IA

Lors de cette étape, plusieurs modèles sont testés afin d'évaluer leurs performances. Le système :

- entraîne différents algorithmes (Random Forest, SVM, MLP, etc.);
- compare leurs métriques (accuracy, F1-score, etc.);
- sélectionne le modèle le plus performant;
- sauvegarde la version optimale pour les prédictions futures.

Cette phase permet de garantir que le modèle retenu est celui offrant la meilleure fiabilité.

### • Prediction — Génération des recommandations

Une fois le modèle entraîné, il peut prédire la filière la plus adaptée pour un nouvel étudiant. Le système :

- applique le même prétraitement aux nouvelles données saisies;
- exécute le modèle sélectionné;
- calcule un pourcentage de compatibilité pour chaque filière;
- génère une explication des critères ayant influencé la prédiction.

Cette étape constitue l'aboutissement du pipeline, produisant une recommandation personnalisée, fiable et compréhensible pour l'étudiant ou le conseiller.

## Conclusion du chapitre

Ce chapitre a permis de présenter l'architecture du système ainsi que le fonctionnement interne du pipeline d'apprentissage automatique. L'approche en trois couches — présentation, logique métier et données — garantit une modularité et une évolutivité adaptées aux exigences d'un système basé sur l'Intelligence Artificielle.

Le pipeline IA, structuré en quatre étapes principales (dataset, prétraitement, entraînement et prédiction), constitue le cœur du modèle de recommandation. Il assure une transformation progressive et maîtrisée des données afin de produire des prédictions fiables et explicables.

Cette architecture et ce pipeline fournissent ainsi une base solide pour la mise en œuvre technique détaillée qui sera développée dans le chapitre suivant, consacré à la réalisation du système.

---

**CHAPITRE V :**  
Réalisation et Implémentation

---

# Introduction

Ce chapitre présente la réalisation et l'implémentation du système d'orientation académique *Orientation-IA*. Il décrit la mise en œuvre concrète des différents composants de l'application, depuis l'organisation générale du projet jusqu'à l'implémentation des interfaces et des fonctionnalités principales. Les écrans de l'application sont également présentés afin d'illustrer le parcours utilisateur et les interactions entre les différentes couches du système.

## 5.1 Organisation générale du projet

Le projet *Orientation-IA* est organisé selon une structure modulaire visant à séparer clairement les différentes responsabilités du système. Cette organisation facilite la compréhension du code, la maintenance, l'évolutivité et le travail collaboratif.

L'arborescence globale du projet est présentée à la Figure 5.1 et repose sur une séparation logique entre l'environnement de développement, le backend, le frontend, le module d'apprentissage automatique et les ressources associées.

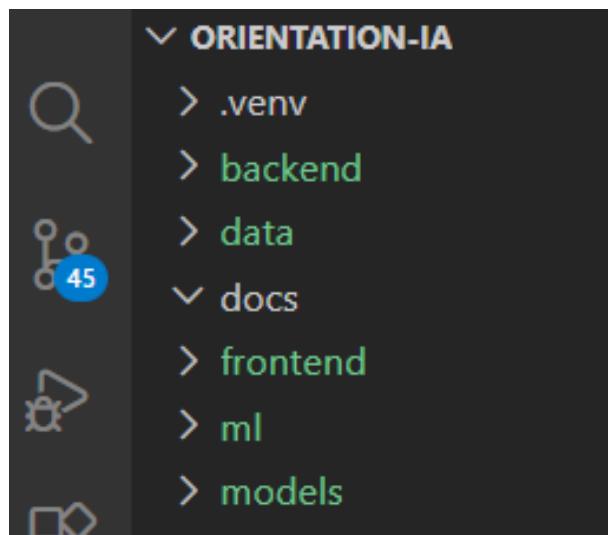


FIGURE 5.1 – Vue globale de l'arborescence du projet Orientation-IA

### 5.1.1 Vue globale de l'arborescence

Le projet est structuré autour des répertoires principaux suivants :

- **.venv** : environnement virtuel Python contenant les dépendances du projet ;
- **backend** : implémentation de l'API FastAPI et de la logique métier ;
- **frontend** : interface utilisateur développée avec Streamlit ;
- **ml** : scripts liés à l'expérimentation et à l'entraînement du modèle ;

- **models** : stockage des modèles d'apprentissage automatique et des métadonnées ;
- **data** : données utilisées pour l'entraînement et les tests ;
- **docs** : documentation et ressources associées au projet.

Cette organisation reflète une séparation claire entre les composants applicatifs, les données et les modèles IA.

### 5.1.2 Organisation du backend (FastAPI)

```

> .venv
└── backend
    ├── app
    │   ├── __pycache__
    │   ├── core
    │   ├── models
    │   ├── routers
    │   ├── schemas
    │   └── services
    │       ├── __pycache__
    │       ├── ml_model.py
    │       ├── __init__.py
    │       ├── main.py
    │       ├── test_db.py
    │       └── .env
    └── requirements.txt

```

**FIGURE 5.2 – Organisation interne du backend FastAPI**

Le dossier **backend** contient l'ensemble de la logique serveur de l'application. Il est structuré autour du répertoire **app**, qui regroupe les composants essentiels de l'API.

- **main.py** : point d'entrée de l'application FastAPI, responsable de l'initialisation du serveur et du routage principal ;
- **routers/** : définition des endpoints de l'API (prédiction, entraînement, gestion des données, etc.) ;
- **services/** : implémentation de la logique métier, incluant l'appel au modèle IA et le traitement des requêtes ;
- **models/** : définition des modèles internes utilisés par l'application ;
- **schemas/** : schémas de validation des données échangées entre le frontend et le backend ;
- **core/** : configuration générale et éléments transverses ;

- **ml\_model.py** : gestion du chargement et de l'exécution du modèle d'apprentissage automatique ;
- **test\_db.py** : scripts de test pour la connexion et la manipulation de la base de données ;
- **requirements.txt** : liste des dépendances Python nécessaires au backend ;
- **.env** : fichier de configuration des variables d'environnement.

Cette structuration permet de respecter les bonnes pratiques de développement des API REST et d'assurer une bonne lisibilité du code.

### 5.1.3 Organisation du frontend (Streamlit)

```

    < frontend
      > _pycache_
      ✎ app.py
      ≡ requirements.txt
  
```

**FIGURE 5.3 – Organisation du frontend Streamlit**

Le dossier **frontend** contient l'interface utilisateur développée avec Streamlit.

- **app.py** : fichier principal de l'interface, responsable de l'affichage des pages, de la collecte des données utilisateur et de l'appel aux services backend ;
- **requirements.txt** : dépendances spécifiques à l'interface Streamlit.

Le frontend communique avec le backend via des requêtes HTTP, ce qui garantit une séparation nette entre l'interface utilisateur et la logique métier.

### 5.1.4 Organisation du module IA et des modèles

```

    > ml
      < models
        { } model_meta.json
        ≡ pipeline.joblib
  
```

**FIGURE 5.4 – Organisation du module ia et des modèles**

Le dossier **ml** est dédié aux expérimentations et à l'entraînement des modèles d'apprentissage automatique. Le dossier **models** regroupe les artefacts produits par le pipeline IA :

- **pipeline.joblib** : modèle entraîné et sérialisé utilisé pour les prédictions ;
- **model\_meta.json** : métadonnées associées au modèle (paramètres, métriques, version).

Cette séparation permet de gérer efficacement les différentes versions du modèle et de faciliter leur réutilisation dans l'application.

### 5.1.5 Avantages de cette organisation

Cette organisation du projet présente plusieurs avantages :

- séparation claire entre interface, backend et Intelligence Artificielle ;
- facilité de maintenance et d'évolution du système ;
- possibilité de réentraîner ou remplacer le modèle IA sans impacter l'interface ;
- meilleure lisibilité du code pour les développeurs et les évaluateurs.

Elle constitue une base solide pour l'implémentation détaillée des fonctionnalités présentées dans les sections suivantes.

### 5.1.6 Interfaces et écrans de l'application

The screenshot shows a web browser window with the URL `localhost:8501`. The page title is **Orientation des étudiants**, accompanied by a graduation cap icon. A sub-instruction reads: "Remplis le formulaire pour obtenir une recommandation personnalisée." On the left, a dark sidebar titled "Orientation IA" contains a "Navigation" section with three items: "Orientation" (selected), "Dashboard", and "Historique". The main content area has a header "Informations étudiant" with fields for "Nom" (Charty), "Prénom" (Malak), and "Email" (charty@gmail.com). Below this, "Ville" (Casablanca) and "Type de Bac" (Sciences Math A) are listed. Under "Notes (sur 20)", there are four rows of subjects with numerical values and adjustment buttons (- +):

- Mathématiques: 20,00
- Physique / Chimie (PC): 19,00
- SVT: 15,00
- Anglais: 13,00
- Économie: 15,00
- Français: 16,00
- Philosophie: Arabe
- Informatique: Informatique

A "Activer Windows" button with a tooltip "Accédez aux paramètres pour activer Windows." is also visible.

FIGURE 5.5 – Interface principale de saisie des informations étudiantes

La Figure 5.5 présente l'interface principale de l'application *Orientation IA*, développée avec **Streamlit**. Elle permet à l'étudiant de saisir ses informations personnelles et ses notes scolaires, qui constituent les données d'entrée du processus de prédition. L'interface, claire et intuitive, intègre une navigation latérale donnant accès aux différentes sections de l'application. Cette page constitue le point d'entrée du système et alimente le pipeline d'apprentissage automatique afin de générer une recommandation d'orientation académique personnalisée.

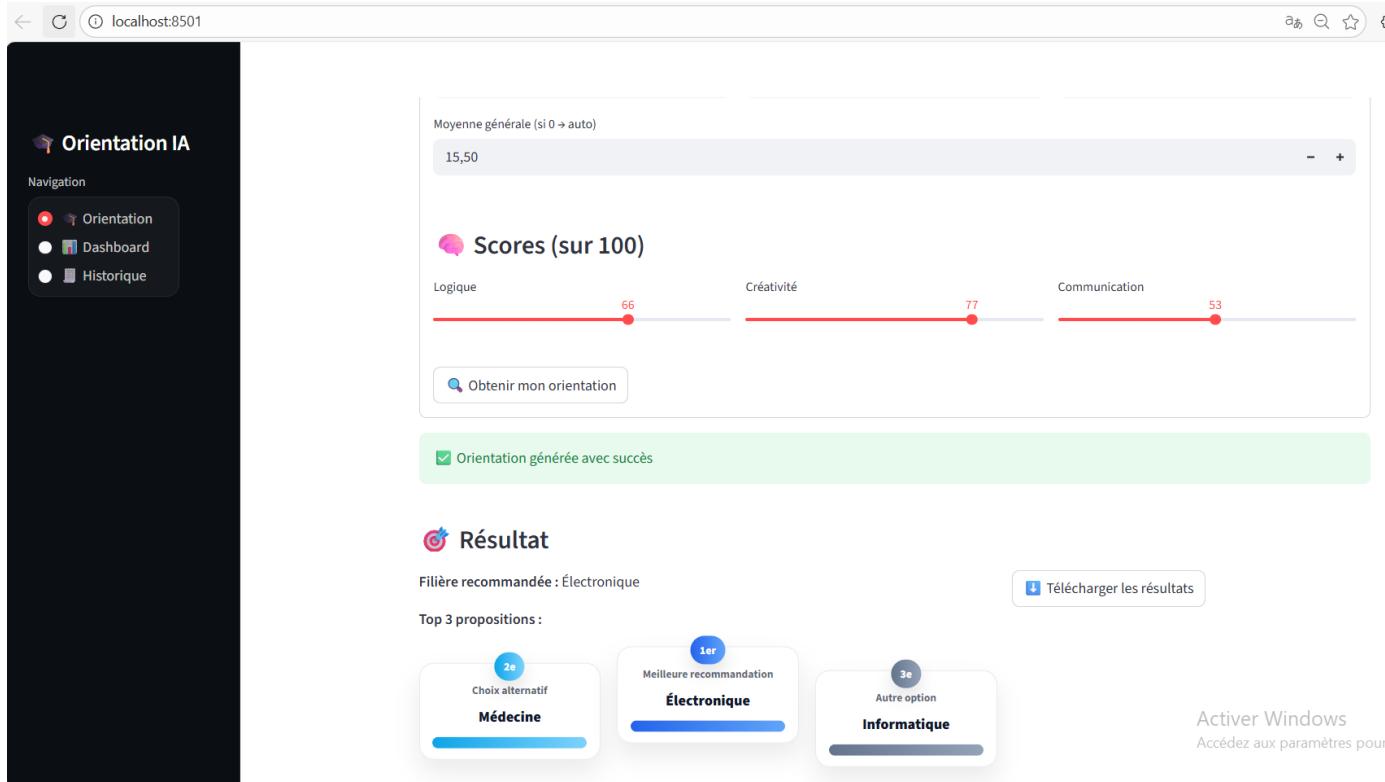
**FIGURE 5.6 – Saisie des informations personnelles de l'étudiant**

La Figure 5.6 illustre l'interface de saisie des informations personnelles de l'étudiant, incluant les données d'identification et le type de baccalauréat. Ces informations constituent le contexte général utilisé, avec les données académiques, pour affiner la recommandation d'orientation. L'interface est conçue de manière simple et intuitive afin de faciliter la saisie des données.

**FIGURE 5.7 – Saisie des notes scolaires et des scores comportementaux**

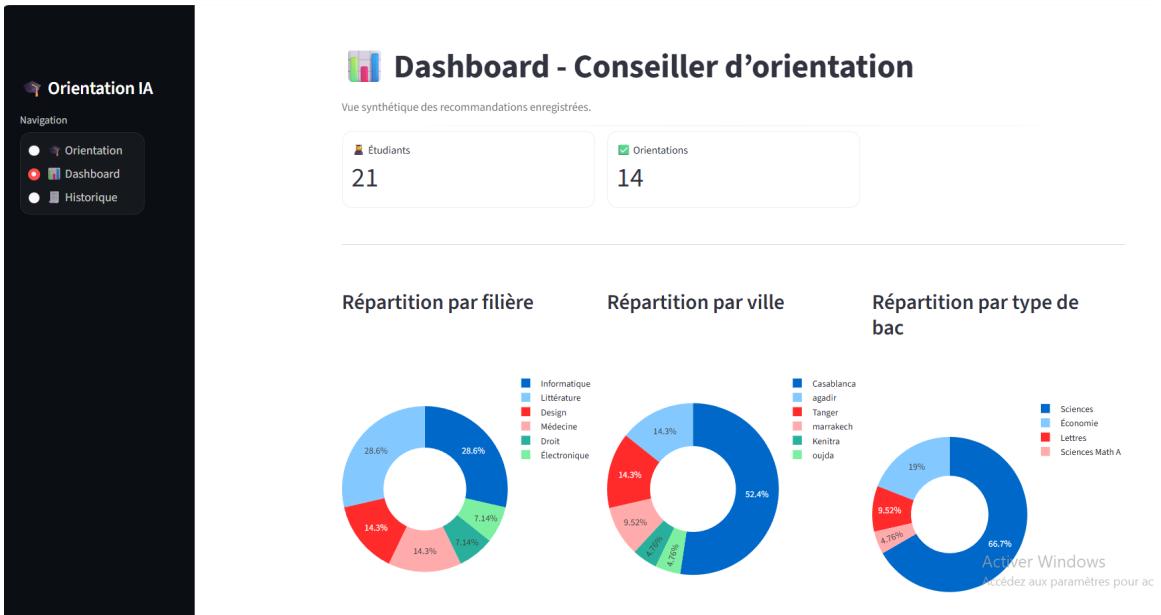
La Figure 5.7 présente l'interface de saisie des performances académiques et des scores

comportementaux de l'étudiant. Elle permet de renseigner les notes par matière, avec un calcul automatique de la moyenne générale, ainsi que des scores cognitifs via des curseurs interactifs. Ces données constituent les principales variables d'entrée du pipeline d'apprentissage automatique pour l'évaluation du profil académique et cognitif.



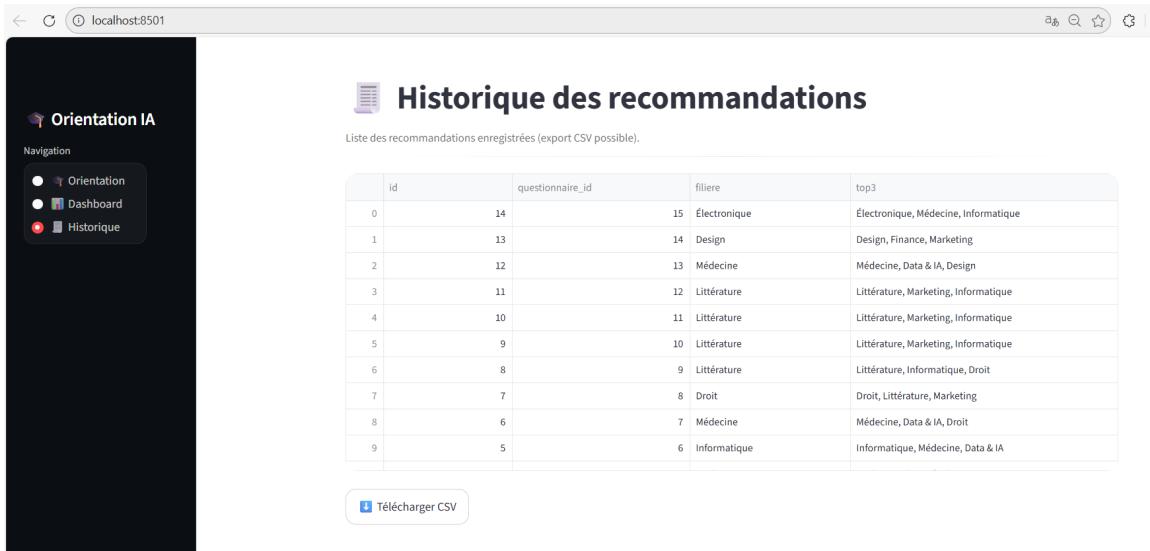
**FIGURE 5.8 – Affichage des résultats et recommandations d’orientation**

La Figure 5.8 illustre l'affichage des résultats générés par le système d'orientation après l'exécution du modèle IA. Elle présente la filière recommandée, un classement des meilleures propositions ainsi qu'une option de téléchargement du rapport PDF. Cette interface constitue l'aboutissement du processus de prédiction en fournissant une recommandation claire et personnalisée.



**FIGURE 5.9 – Dashboard du conseiller d'orientation**

La Figure 5.9 illustre le tableau de bord du conseiller d’orientation. Il présente une vue synthétique des recommandations générées, accompagnée d’indicateurs globaux et de visualisations graphiques. Ces éléments permettent d’analyser les tendances d’orientation selon différents critères et de mieux comprendre les profils des étudiants.



**FIGURE 5.10 – Historique des recommandations**

La Figure 5.10 présente la page d’historique des recommandations générées par le système. Elle affiche, sous forme de tableau, les orientations enregistrées et permet leur consultation, leur suivi ainsi que l’export des données. Cette interface facilite l’analyse et l’exploitation des recommandations passées.

## Conclusion du chapitre

Ce chapitre a présenté la réalisation et l'implémentation du système d'orientation académique *Orientation-IA*. Il a permis de décrire l'organisation générale du projet, la structuration des modules ainsi que la mise en œuvre concrète des composants frontend, backend et du module d'Intelligence Artificielle. Les principales interfaces de l'application ont été introduites afin d'illustrer le parcours utilisateur, depuis la saisie des informations étudiantes jusqu'à l'affichage des résultats et des recommandations d'orientation. Le tableau de bord du conseiller et la page d'historique viennent compléter le système en offrant des fonctionnalités de suivi et d'analyse des données.

Cette phase de réalisation confirme la cohérence des choix de conception et d'architecture adoptés. Le chapitre suivant sera consacré à l'évaluation du modèle et à l'analyse des résultats, afin d'apprécier la performance et la pertinence des recommandations générées.

---

**CHAPITRE VI :**

Conclusion Générale

---

Ce projet a porté sur la conception et la réalisation d'un système intelligent d'aide à l'orientation académique basé sur les techniques d'Intelligence Artificielle et d'apprentissage automatique. Face aux difficultés rencontrées par les étudiants dans le choix de leur filière, l'objectif principal était de proposer une solution capable de fournir des recommandations personnalisées, fiables et exploitables, en tenant compte des performances scolaires, des compétences comportementales et des informations personnelles des étudiants.

Tout au long de ce travail, une démarche méthodologique structurée a été adoptée. Après l'analyse du contexte et la définition des besoins fonctionnels, une architecture modulaire a été conçue, reposant sur une interface utilisateur développée avec Streamlit, un backend sous forme d'API REST implémentée avec FastAPI, et un module d'apprentissage automatique dédié à la génération des prédictions. Le pipeline IA mis en place, depuis la préparation des données jusqu'à la prédiction finale, constitue le cœur du système et garantit la cohérence des résultats produits.

La phase de réalisation a permis de concrétiser les choix de conception en une application fonctionnelle. Les différentes interfaces développées offrent une expérience utilisateur intuitive aussi bien pour l'étudiant que pour le conseiller d'orientation, tandis que le backend assure la gestion des données, l'exécution du modèle IA et le suivi des recommandations. Les résultats obtenus démontrent la faisabilité et l'intérêt d'un tel système dans un contexte d'orientation académique assistée par l'IA.

En perspective, plusieurs axes d'amélioration peuvent être envisagés, notamment l'enrichissement du dataset, l'intégration de nouveaux critères d'évaluation, l'amélioration des performances du modèle par l'utilisation d'algorithmes plus avancés, ainsi que le déploiement de l'application sur une plateforme cloud ou mobile. Ces évolutions permettraient de renforcer la précision des recommandations et d'élargir l'usage du système à un plus grand nombre d'utilisateurs.

En conclusion, ce projet a permis de mettre en œuvre une solution innovante et cohérente combinant Intelligence Artificielle et technologies web modernes, contribuant ainsi à une meilleure orientation académique et à un accompagnement plus personnalisé des étudiants.