



**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**

MOUHANAD EL FILALI

**Année universitaire**

2025/2026

# Table des matières

Liste des figures . . . . .	3
Remerciements . . . . .	4
Résumé . . . . .	5
Abstract . . . . .	6
<b>Introduction Générale</b>	<b>7</b>
<b>Chapitre I — Présentation du projet</b>	<b>8</b>
1.1 Contexte et sujet du projet . . . . .	9
1.2 Problématique . . . . .	9
1.3 Objectifs du projet . . . . .	10
1.4 User Stories . . . . .	11
1.5 Périmètre fonctionnel du projet . . . . .	12
<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 — Collecte des données et prédiction de base . . . . .	16
2.5.2 Sprint 2 — Analyse avancée et génération de rapports . . . . .	17
2.5.3 Sprint 3 — Gestion du modèle IA et amélioration continue . . . . .	18
<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 . . . . .	22
3.3 Diagramme d'activités . . . . .	24
<b>Chapitre IV — Architecture du Système et Pipeline d'Apprentissage Automatique</b>	<b>26</b>
4.1 Architecture générale du système . . . . .	27
4.1.1 Couche Présentation – Interface Web Streamlit . . . . .	28
4.1.2 Couche Logique Métier – Backend FastAPI . . . . .	28
4.1.3 Couche Données – MySQL et Stockage de fichiers . . . . .	29

4.2	Justification du choix de l'architecture . . . . .	29
4.3	Pipeline d'apprentissage automatique (IA) . . . . .	31
4.3.1	Modèle Random Forest . . . . .	32
<b>Chapitre V — Réalisation et Implémentation</b>		<b>34</b>
5.1	Environnement et outils de développement . . . . .	35
5.2	Organisation générale du projet . . . . .	36
5.2.1	Vue globale de l'arborescence . . . . .	36
5.2.2	Organisation du backend (FastAPI) . . . . .	37
5.2.3	Organisation du frontend (Streamlit) . . . . .	38
5.2.4	Organisation du module IA et des modèles . . . . .	38
5.2.5	Avantages de cette organisation . . . . .	39
5.2.6	Interfaces et écrans de l'application . . . . .	39
	Perspectives et travaux futurs . . . . .	44
<b>Bibliographie</b>		<b>45</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 prédiction d'orientation . . . . .	22
3.3	Diagramme d'activités du processus de prédiction d'orientation . . . . .	24
4.1	Architecture générale du système d'orientation académique . . . . .	27
4.2	Logos : Streamlit — Python . . . . .	28
4.3	Logos : FastAPI — Python . . . . .	28
4.4	Logo :MySQL . . . . .	29
4.5	Pipeline global du processus d'apprentissage automatique . . . . .	31
4.6	Logos : Random Forest . . . . .	33
5.1	Logo Visual Studio Code . . . . .	35
5.2	Logo Git . . . . .	35
5.3	Logo Jira . . . . .	36
5.4	Vue globale de l'arborescence du projet Orientation-IA . . . . .	36
5.5	Organisation interne du backend FastAPI . . . . .	37
5.6	Organisation du frontend Streamlit . . . . .	38
5.7	Organisation du module ia et des modèles . . . . .	38
5.8	Interface principale de saisie des informations étudiantes . . . . .	39
5.9	Saisie des informations personnelles de l'étudiant . . . . .	40
5.10	Saisie des notes scolaires et des scores comportementaux . . . . .	40
5.11	Affichage des résultats et recommandations d'orientation . . . . .	41
5.12	Dashboard d'analyse . . . . .	41
5.13	Historique des recommandations . . . . .	42

## 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 **prédire** 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

---

# 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.

## 1.1 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.

## 1.2 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.

## 1.3 Objectifs du projet

### Objectif général

L'objectif principal de ce projet est de concevoir et de développer un système intelligent d'aide à l'orientation académique permettant de recommander automatiquement une filière d'études adaptée au profil d'un utilisateur, en s'appuyant sur l'analyse de données personnelles, scolaires et comportementales à l'aide de techniques d'Intelligence Artificielle.

### Objectifs spécifiques

Afin d'atteindre cet objectif général, les objectifs spécifiques suivants ont été définis :

- Concevoir et structurer un dataset contenant les informations nécessaires à la prédiction d'orientation (notes scolaires, type de baccalauréat, compétences comportementales, centres d'intérêt, etc.).
- Mettre en place un pipeline d'apprentissage automatique permettant l'entraînement, le réentraînement et l'évaluation d'un modèle d'Intelligence Artificielle dédié à la recommandation d'orientation académique.
- Développer une interface web simple, intuitive et interactive à l'aide de **Streamlit**, afin de permettre à l'utilisateur de saisir ses informations et de consulter facilement les résultats.
- Implémenter un mécanisme de prédiction permettant de générer des recommandations personnalisées accompagnées d'indicateurs de compatibilité.
- Offrir à l'utilisateur la possibilité de consulter les résultats et recommandations générés par le système, ainsi que l'historique des orientations produites.
- Générer automatiquement un rapport d'orientation au format PDF intégrant les résultats, les recommandations et les informations principales de la prédiction.
- Mettre en place des fonctionnalités de gestion du dataset (import et mise à jour) afin d'améliorer continuellement la qualité des données utilisées par le système.
- Permettre le suivi des performances du modèle d'apprentissage automatique à travers

des métriques d'évaluation, afin d'analyser la fiabilité des prédictions.

## 1.4 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 et académiques afin de permettre au système de générer une recommandation d'orientation adaptée à mon profil.
- **US2** : En tant qu'utilisateur, je veux lancer une prédiction d'orientation afin d'obtenir une filière recommandée automatiquement par le modèle IA.
- **US3** : En tant qu'utilisateur, je veux consulter les résultats et recommandations affichés après la prédiction afin de comprendre l'orientation proposée et les options disponibles.
- **US4** : En tant qu'utilisateur, je veux visualiser un classement des filières avec un score ou pourcentage de compatibilité afin de comparer plusieurs choix d'orientation.
- **US5** : En tant qu'utilisateur, je veux consulter l'historique des orientations générées afin de suivre mes recommandations et retrouver des résultats précédents.
  
- **US6** : En tant qu'utilisateur, je veux consulter une orientation enregistrée (détail d'un résultat) afin d'analyser la recommandation et ses informations associées.
- **US7** : En tant qu'utilisateur, je veux accéder à une interface simple et intuitive via Streamlit afin d'interagir facilement avec le système et ses fonctionnalités.
- **US8** : En tant qu'utilisateur, je veux télécharger un rapport d'orientation au format PDF afin de conserver une trace des résultats et des recommandations.
- **US9** : En tant qu'utilisateur, je veux importer un dataset d'entraînement afin d'alimenter le système avec des données exploitables pour l'apprentissage.
- **US10** : En tant qu'utilisateur, je veux mettre à jour le dataset (ajout ou remplacement de données) afin d'améliorer la qualité et l'actualité des données utilisées.
  
- **US11** : En tant qu'utilisateur, je veux entraîner le modèle IA à partir du dataset afin de construire une version utilisable pour la prédiction d'orientation.
- **US12** : En tant qu'utilisateur, je veux réentraîner le modèle après mise à jour du dataset afin de maintenir ou améliorer la performance du système.
- **US13** : En tant qu'utilisateur, je veux suivre les performances du modèle (métriques) afin d'évaluer la fiabilité des prédictions générées.
- **US14** : En tant qu'utilisateur, je veux consulter les métriques et résultats d'évalua-

tion du modèle afin de comparer l'évolution des performances après entraînement ou réentraînement.

- **US15** : En tant qu'utilisateur, je veux disposer d'un accès aux fonctionnalités principales (prédiction, consultation, gestion des données et suivi du modèle) afin d'utiliser le système de bout en bout, de la saisie jusqu'au rapport.

## 1.5 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 du chapitre

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, leurs dépendances et leur contribution à la construction progressive du système de prédiction et de recommandation académique.

Les sprints ont été planifiés et suivis à l’aide de l’outil **Jira**, ce qui a permis :

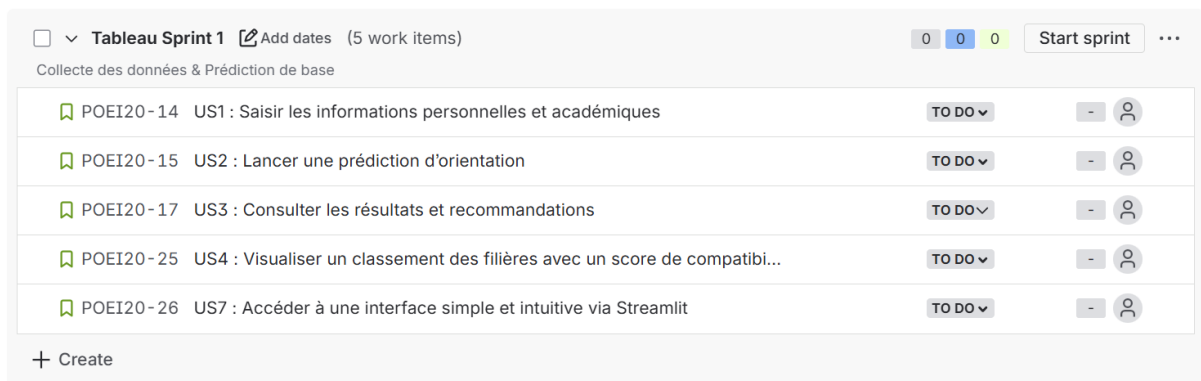
- d’avoir une vision claire de l’avancement global du projet ;
- de répartir les fonctionnalités sur plusieurs itérations ;
- d’assurer la traçabilité des user stories et des évolutions du système.

### 2.5.1 Sprint 1 — Collecte des données et prédiction de base

Le premier sprint a été consacré à la mise en place du socle fonctionnel du système. L’objectif principal était de permettre à l’utilisateur de saisir ses informations personnelles et académiques, puis d’obtenir une première prédiction d’orientation générée automatiquement par le modèle d’apprentissage automatique.

Durant ce sprint, les fonctionnalités suivantes ont été mises en place :

- saisie des informations personnelles et académiques de l’utilisateur ;
- lancement d’une prédiction d’orientation académique ;
- affichage des résultats et recommandations générées par le modèle IA ;
- visualisation d’un classement des filières avec un score de compatibilité ;
- mise en place d’une interface simple et intuitive via Streamlit.

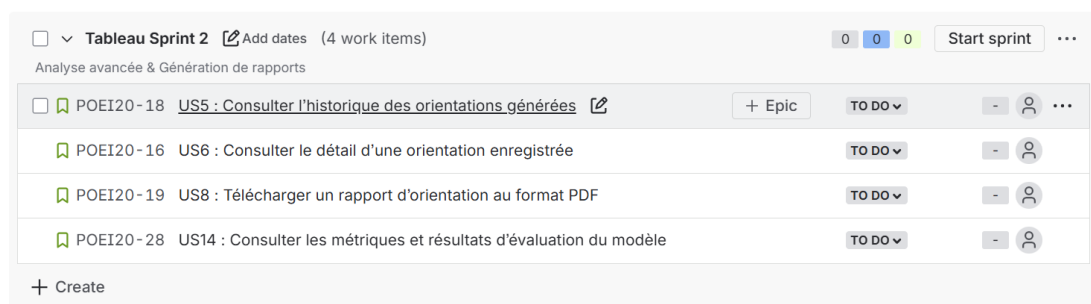


**FIGURE 2.1** – Planning et suivi du Sprint 1 dans Jira

Ce sprint a permis d’obtenir un *MVP* (Minimum Viable Product) fonctionnel, constituant la première version utilisable du système.

## 2.5.2 Sprint 2 — Analyse avancée et génération de rapports

Le second sprint a visé l’enrichissement des fonctionnalités de consultation et d’analyse des résultats produits par le système. Il s’est concentré sur l’exploitation des prédictions générées, leur historique ainsi que leur restitution sous forme de rapports.



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

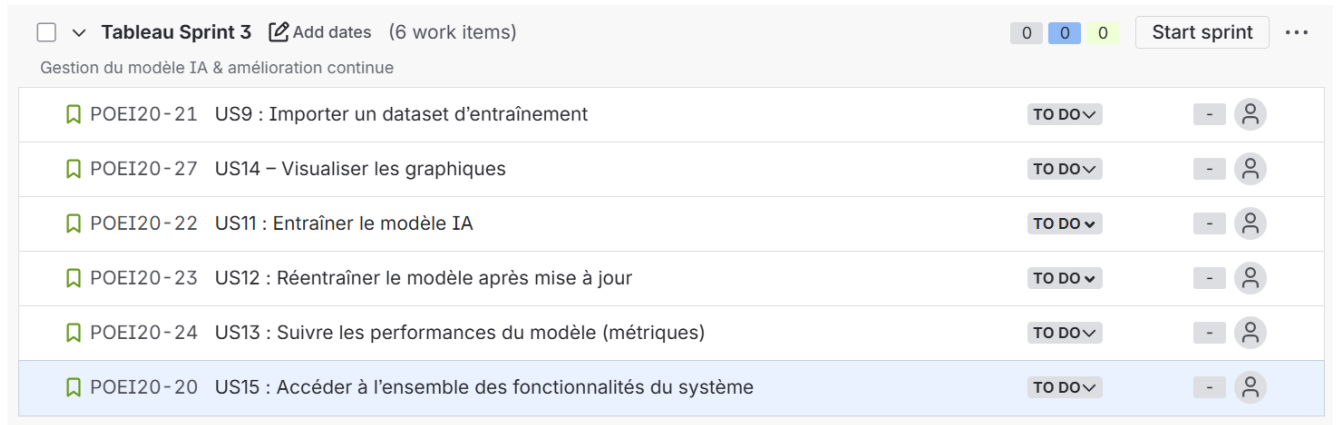
Les principaux travaux réalisés durant ce sprint sont :

- consultation de l’historique des orientations générées ;
- accès au détail d’une orientation enregistrée ;
- génération et téléchargement d’un rapport d’orientation au format PDF ;
- consultation des métriques et résultats d’évaluation du modèle IA.

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

### 2.5.3 Sprint 3 — Gestion du modèle IA et amélioration continue

Le troisième sprint s’est concentré sur les fonctionnalités liées à la gestion du modèle d’Intelligence Artificielle et à l’amélioration continue du système. L’objectif était de rendre la solution évolutive, supervisée et adaptable aux mises à jour des données.



☐ ▾ Tableau Sprint 3 <a href="#">Add dates</a> (6 work items)		0 0 0	Start sprint	...
Gestion du modèle IA & amélioration continue				
🔖 POEI20-21	US9 : Importer un dataset d'entraînement	TO DO ▾	-	👤
🔖 POEI20-27	US14 – Visualiser les graphiques	TO DO ▾	-	👤
🔖 POEI20-22	US11 : Entraîner le modèle IA	TO DO ▾	-	👤
🔖 POEI20-23	US12 : Réentraîner le modèle après mise à jour	TO DO ▾	-	👤
🔖 POEI20-24	US13 : Suivre les performances du modèle (métriques)	TO DO ▾	-	👤
🔖 POEI20-20	US15 : Accéder à l'ensemble des fonctionnalités du système	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 d’entraînement ;
- l’entraînement initial du modèle IA ;
- le réentraînement du modèle après modification des données ;
- le suivi des performances du modèle à l’aide de métriques d’évaluation ;
- l’accès à l’ensemble des fonctionnalités du système de bout en bout.

Ce sprint finalise l’écosystème du projet en intégrant la gestion complète du cycle de vie du modèle IA, depuis les données jusqu’à l’évaluation des performances.

## 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

---

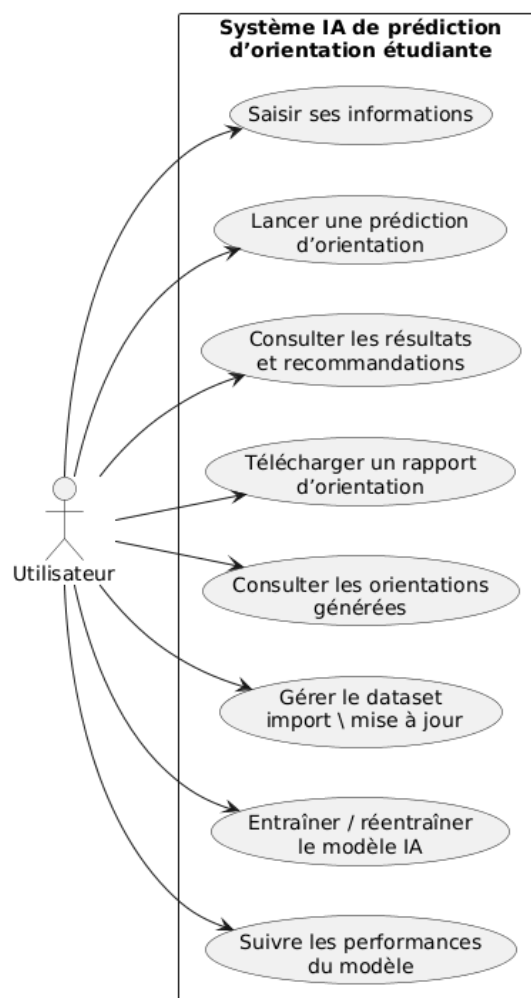
## Introduction du chapitre

Ce chapitre présente la conception du système d'orientation académique à travers différents modèles UML. Il a pour objectif de décrire le fonctionnement global de l'application, les principales interactions avec le système, ainsi que l'enchaînement des traitements internes.

Les diagrammes de cas d'utilisation, d'activités et de séquence permettent de passer d'une vision fonctionnelle des besoins à une représentation plus détaillée des flux de données et des mécanismes de prédiction, servant de base à la réalisation technique du système.

Dans ce projet, le terme « utilisateur » désigne toute personne interagissant avec la plateforme, que ce soit pour obtenir une orientation académique personnalisée ou pour consulter les résultats, analyses et tableaux de bord proposés par 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 illustre les principales fonctionnalités offertes par le système intelligent de prédiction d'orientation académique ainsi que les interactions possibles avec celles-ci.

Le système permet la saisie des informations nécessaires à l'analyse du profil académique et personnel. À partir de ces données, une prédiction d'orientation peut être lancée afin de générer des recommandations adaptées. Les résultats obtenus peuvent ensuite être consultés, analysés et exportés sous forme d'un rapport d'orientation au format PDF.

Le diagramme met également en évidence des fonctionnalités avancées liées à l'exploitation et à l'évolution du système. Il est possible de consulter les orientations générées, de gérer le dataset par l'importation ou la mise à jour des données, d'entraîner ou de réentraîner le modèle d'Intelligence Artificielle, ainsi que de suivre les performances du modèle afin d'évaluer la qualité des prédictions produites.

Certaines fonctionnalités présentent des dépendances logiques dans le processus global. La saisie des informations constitue une étape préalable au lancement d'une prédiction. La consultation des résultats permet l'accès au téléchargement du rapport d'orientation. Par ailleurs, la gestion du dataset conditionne l'entraînement et le réentraînement du modèle, tandis que le suivi des performances permet d'analyser la fiabilité et l'évolution du système.

Ce diagramme fournit ainsi une vue synthétique et globale des fonctionnalités essentielles du système d'orientation académique, depuis la collecte des données jusqu'à l'évaluation continue du modèle d'apprentissage automatique.

## 3.2 Diagramme de séquence

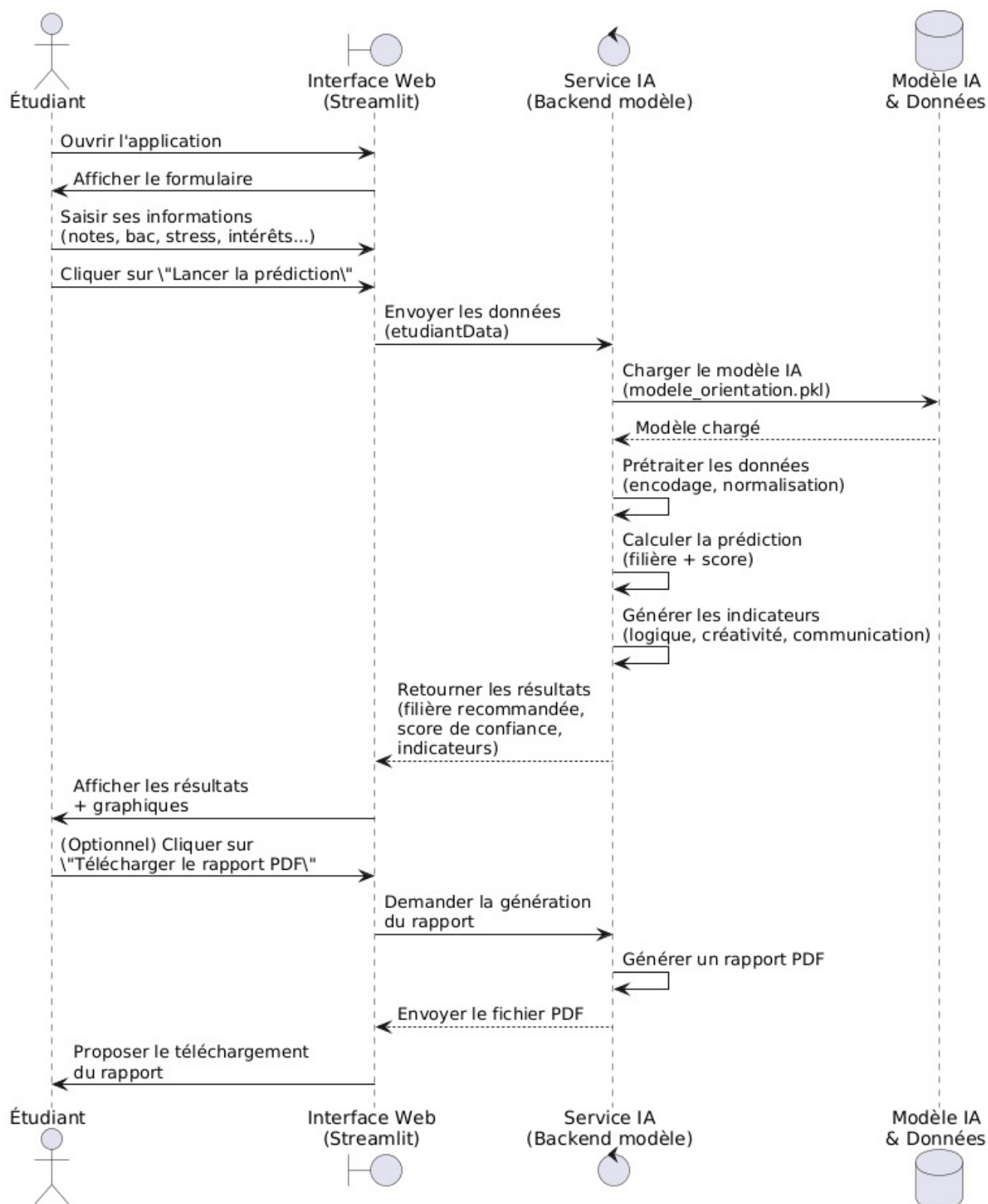


FIGURE 3.2 – Diagramme de séquence du processus de prédiction d'orientation

## **=>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

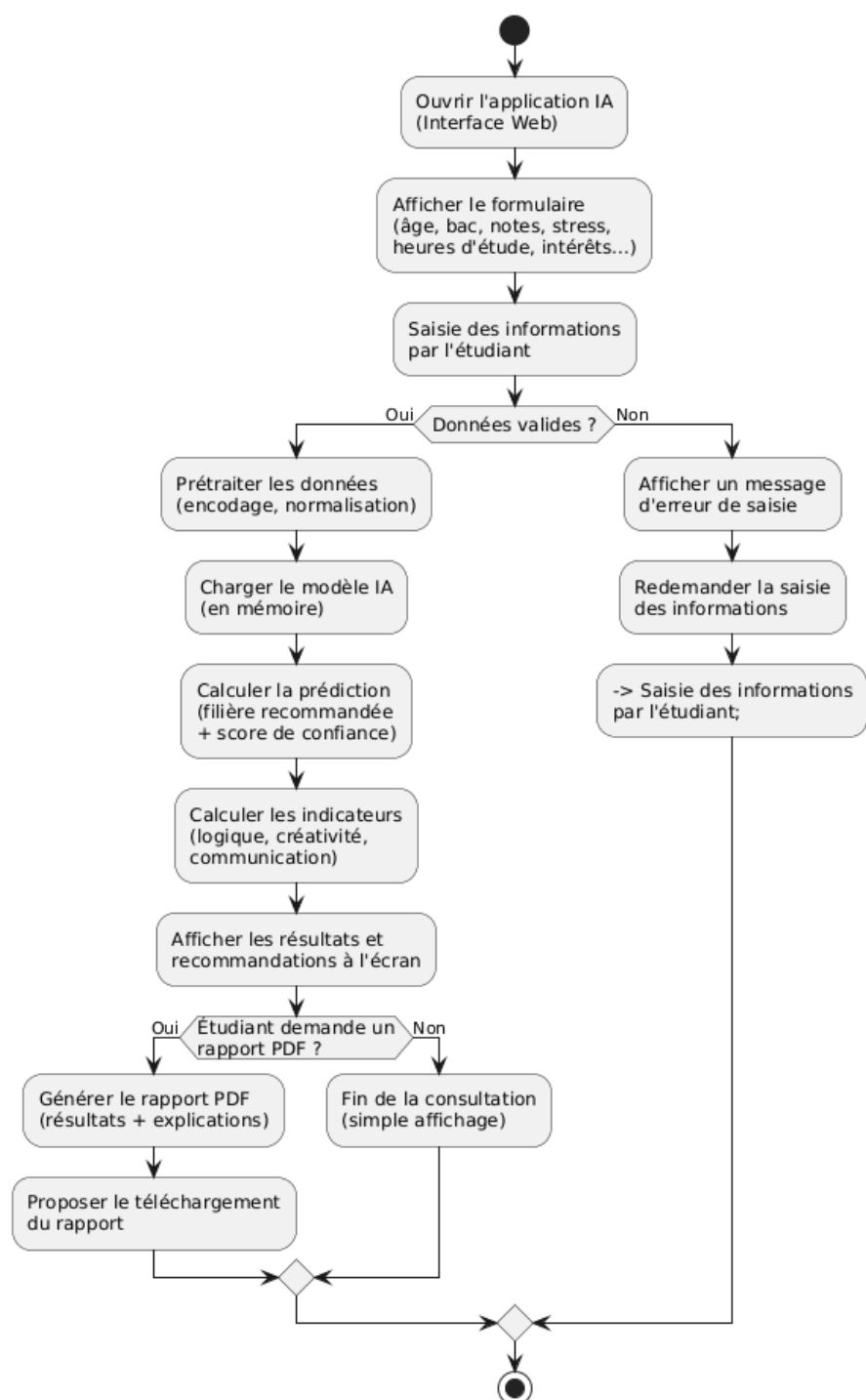


FIGURE 3.3 – Diagramme d'activités du processus de prédiction d'orientation

## **=>Description du diagramme d'activités**

Ce diagramme d'activités illustre le déroulement complet du processus de prédiction 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

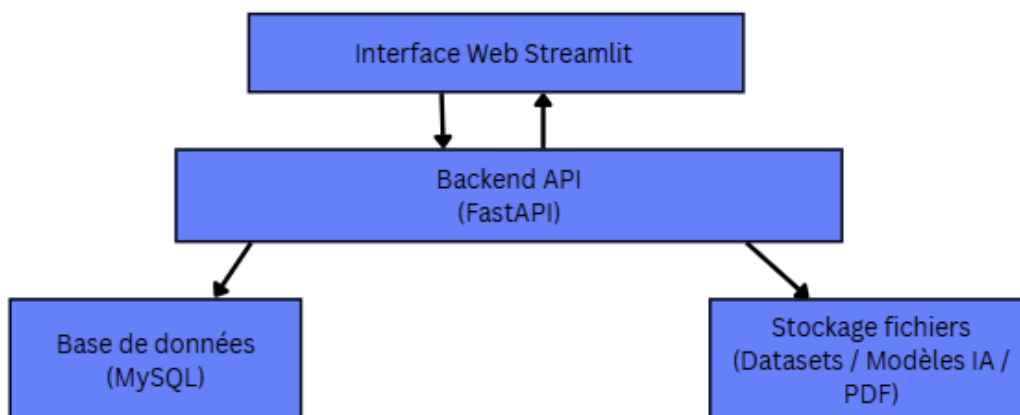
Ce chapitre présente l'architecture technique globale du système ainsi que le pipeline d'apprentissage automatique qui constitue le cœur du moteur de recommandation. L'objectif est d'expliquer la structure logicielle du projet, les technologies utilisées et la manière dont les différentes couches interagissent 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, réentraînement 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 bibliothèques 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)

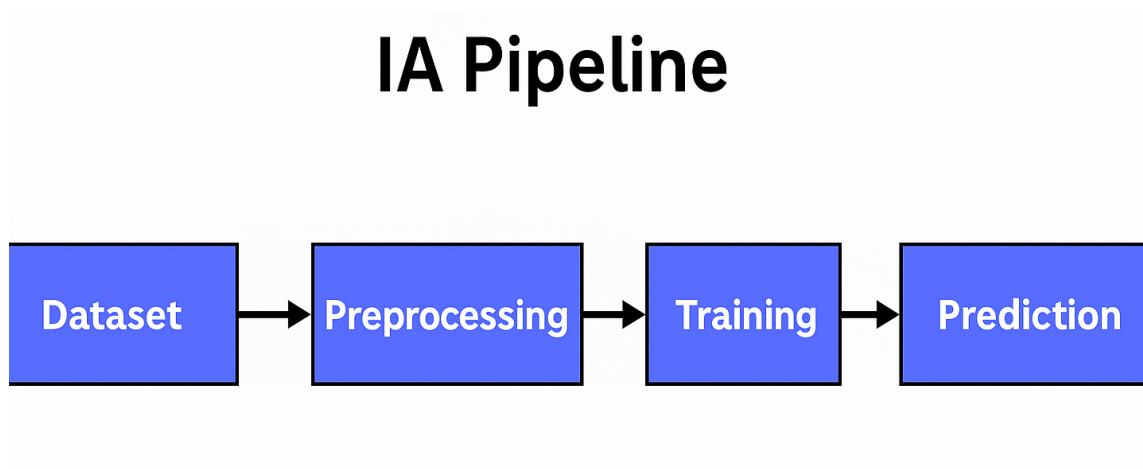


FIGURE 4.5 – Pipeline global du processus d'apprentissage automatique

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.

### 4.3.1 Modèle Random Forest

Le modèle **Random Forest** est un algorithme d'apprentissage supervisé basé sur le principe des méthodes ensemblistes (*ensemble learning*). Il repose sur la construction de plusieurs arbres de décision entraînés sur différentes sous-parties du dataset, puis sur l'agrégation de leurs prédictions afin d'obtenir un résultat final plus robuste et plus stable.

Dans le cadre du projet *Orientation-IA*, Random Forest a été utilisé comme l'un des modèles principaux du pipeline d'apprentissage automatique. Cet algorithme est particulièrement adapté à la problématique de l'orientation académique, car il permet de traiter efficacement des données hétérogènes, combinant des variables numériques (notes scolaires, scores cognitifs) et des variables catégorielles (type de baccalauréat, centres d'intérêt).

Un avantage majeur de Random Forest réside dans sa capacité à limiter le sur-apprentissage (*overfitting*) par rapport à un arbre de décision unique, tout en conservant de bonnes performances prédictives. De plus, il offre la possibilité d'analyser l'importance relative des variables,

contribuant ainsi à une meilleure interprétabilité des recommandations générées.

Les performances du modèle Random Forest ont été évaluées à l'aide de métriques classiques telles que l'accuracy et le F1-score, puis comparées à celles d'autres algorithmes testés dans le pipeline. Les résultats obtenus ont montré un bon compromis entre précision, stabilité et explicabilité, justifiant ainsi son intégration dans le système de recommandation d'orientation académique.

Dans le contexte du pipeline d'apprentissage automatique mis en place, le modèle Random Forest a été retenu car il offre un compromis optimal entre performance prédictive, robustesse face aux données hétérogènes et capacité d'explication des résultats, répondant ainsi aux exigences de fiabilité et d'interprétabilité du système Orientation-IA.



**FIGURE 4.6** – Logos : Random Forest

## 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 Environnement et outils de développement

Le développement du système *Orientation-IA* a été réalisé à l'aide de plusieurs outils logiciels permettant une implémentation efficace, collaborative et maintenable.

L'éditeur de code principal utilisé est **Visual Studio Code (VS Code)**, un environnement de développement léger et extensible, particulièrement adapté aux projets Python et aux applications orientées Data Science. Il a été utilisé pour le développement du backend FastAPI, du frontend Streamlit ainsi que pour la gestion des scripts liés au pipeline d'apprentissage automatique. VS Code offre plusieurs avantages :



**FIGURE 5.1** – Logo Visual Studio Code

- une prise en charge native du langage Python ;
- une intégration efficace avec les environnements virtuels ;
- des extensions dédiées au développement IA (linting, debugging, gestion des dépendances) ;
- une meilleure organisation et lisibilité du code source.

En complément, les outils suivants ont été utilisés :

- **Git** pour la gestion de versions ;



**FIGURE 5.2** – Logo Git

- **Jira** pour le suivi Agile des sprints ;



FIGURE 5.3 – Logo Jira

## 5.2 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.

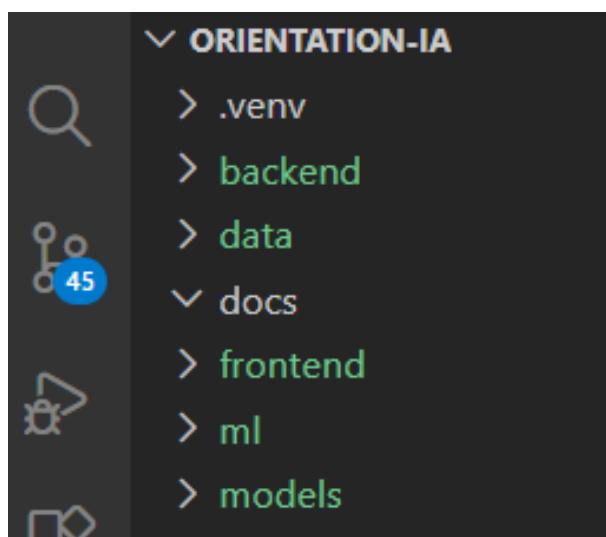


FIGURE 5.4 – Vue globale de l'arborescence du projet Orientation-IA

### 5.2.1 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.

### 5.2.2 Organisation du backend (FastAPI)

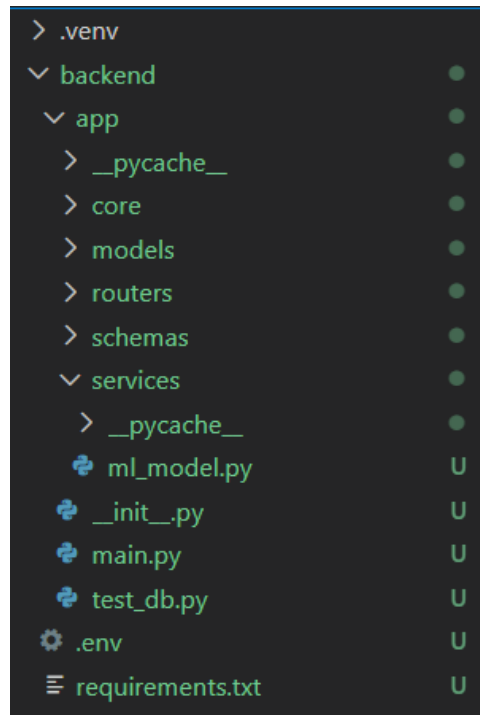


FIGURE 5.5 – Organisation interne du backend FastAPI

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.

### 5.2.3 Organisation du frontend (Streamlit)

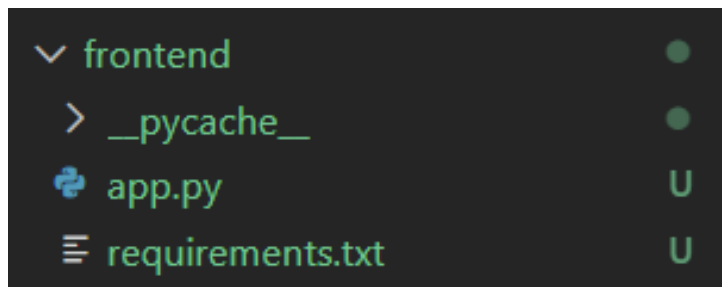


FIGURE 5.6 – Organisation du frontend Streamlit

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.

### 5.2.4 Organisation du module IA et des modèles

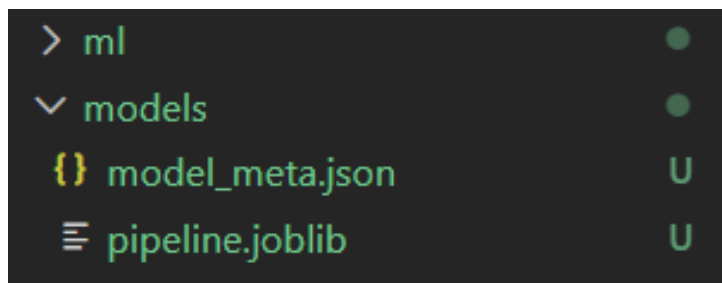


FIGURE 5.7 – Organisation du module ia et des modèles

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.

### 5.2.5 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.

### 5.2.6 Interfaces et écrans de l'application

**FIGURE 5.8** – Interface principale de saisie des informations étudiantes

La Figure 5.8 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 prédiction. 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.

**Orientation IA**

Navigation

- Orientation
- Dashboard
- Historique

## Orientation des étudiants

Remplis le formulaire pour obtenir une recommandation personnalisée.

### Informations étudiant

Nom: Charty Prénom: Malak Email: charty@gmail.com

Ville: Casablanca Type de Bac: Sciences Math A

**FIGURE 5.9** – Saisie des informations personnelles de l'étudiant

La Figure 5.9 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.

**Orientation IA**

Navigation

- Orientation
- Dashboard
- Historique

### Notes (sur 20)

Mathématiques	Physique / Chimie (PC)	SVT
20,00	19,00	15,00
Anglais	Économie	Français
13,00	15,00	16,00
Philosophie	Arabe	Informatique
10,00	13,00	16,00
Moyenne générale (si 0 -> auto): 15,22		

### Scores (sur 100)

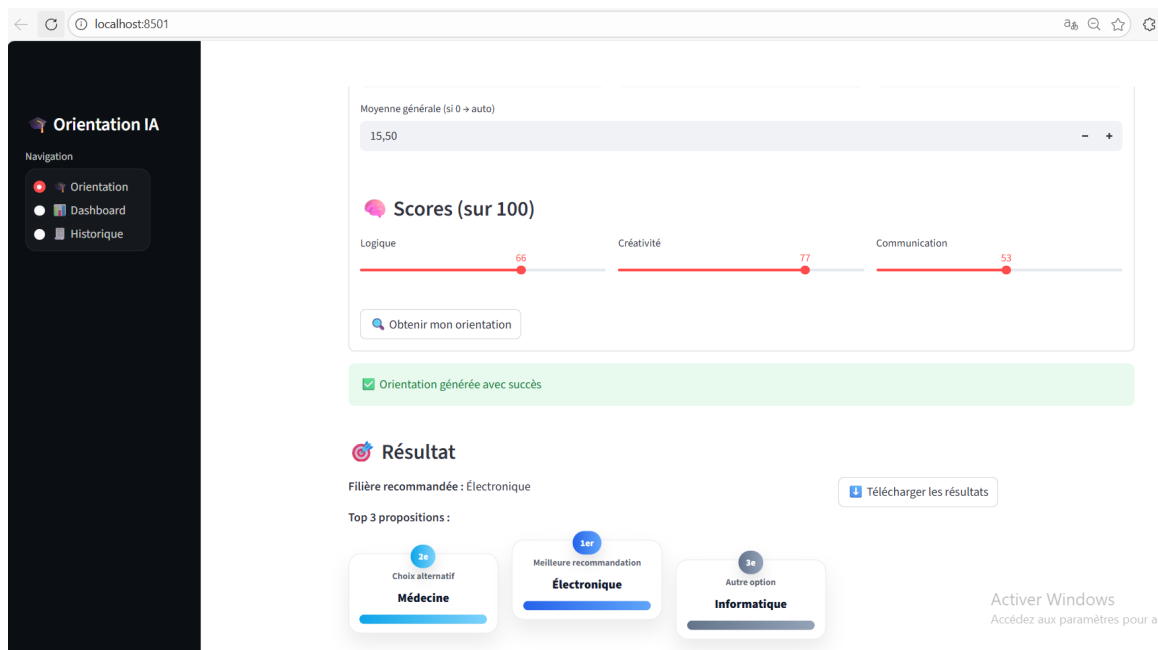
Logique: 66 Créativité: 77 Communication: 53

Obtenir mon orientation

Activer Windows  
Accédez aux paramètres pour activer Windows.

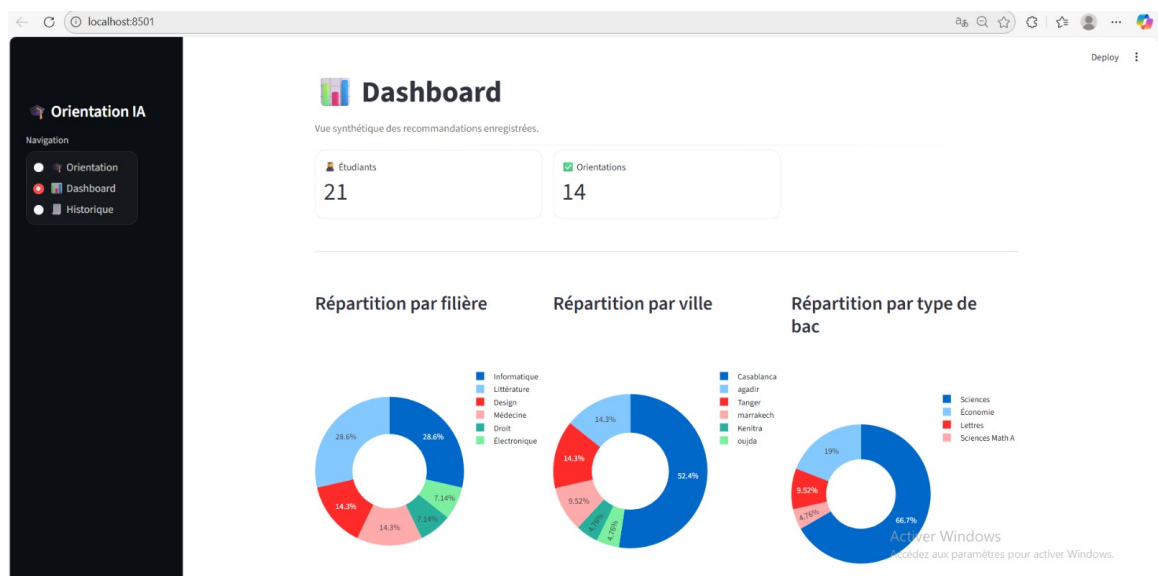
**FIGURE 5.10** – Saisie des notes scolaires et des scores comportementaux

La Figure 5.10 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.



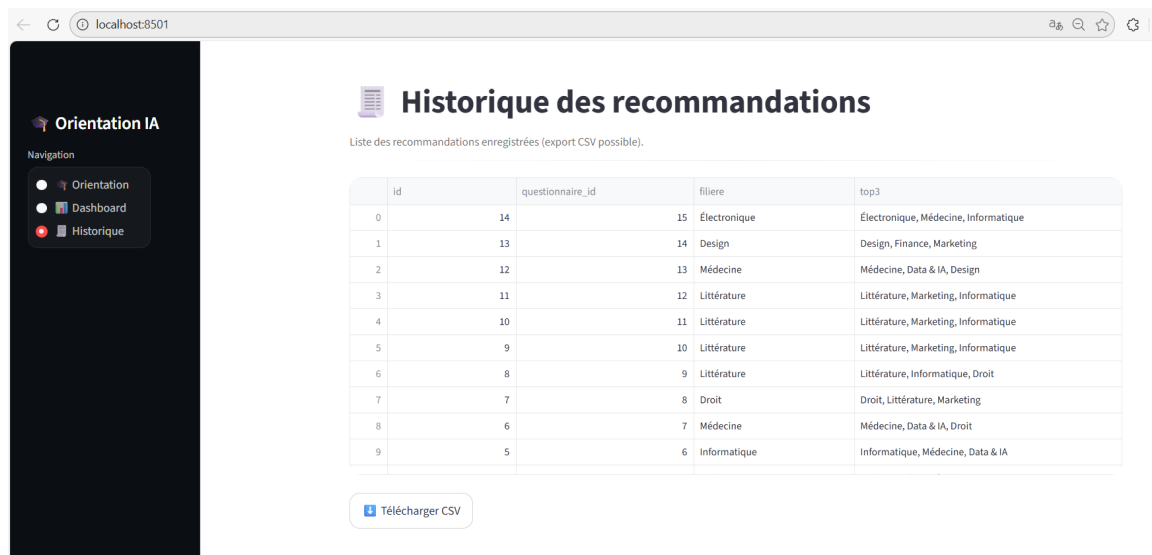
**FIGURE 5.11** – Affichage des résultats et recommandations d’orientation

La Figure 5.11 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 5.12** – Dashboard d’analyse

La Figure 5.12 illustre le tableau de bord d’analyse. 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 5.13** – Historique des recommandations

La Figure 5.13 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.

# 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.

# Perspectives et travaux futurs

Bien que le système *Orientation-IA* soit fonctionnel et réponde aux objectifs définis, plusieurs axes d'amélioration peuvent être envisagés afin d'augmenter la fiabilité des recommandations et d'élargir le périmètre d'usage.

- **Enrichissement et qualité du dataset** : intégrer davantage de profils étudiants, diversifier les établissements et les parcours, et améliorer la gestion des valeurs manquantes afin d'augmenter la représentativité des données d'entraînement.
- **Optimisation et robustesse des modèles** : mettre en place une validation croisée (*k-fold*), une recherche d'hyperparamètres (*Grid Search* / *Random Search*) et tester des modèles plus performants (ex. : *XGBoost*, *LightGBM*, ou ensembles hybrides) afin d'améliorer la précision et la stabilité des prédictions.
- **Explicabilité avancée des recommandations** : intégrer des méthodes d'interprétabilité (ex. : *SHAP* ou *LIME*) afin d'expliquer clairement les facteurs ayant influencé la recommandation et renforcer la confiance des utilisateurs.
- **Personnalisation et prise en compte du feedback** : permettre aux utilisateurs de fournir un retour (satisfaction, filière réellement choisie) et exploiter ce feedback pour un apprentissage continu et une amélioration progressive du système.
- **Sécurité et conformité** : renforcer la protection des données personnelles (contrôle d'accès, chiffrement, anonymisation), et formaliser une politique de conservation des données pour respecter les bonnes pratiques de confidentialité.
- **Déploiement et montée en charge** : déployer la solution sur une infrastructure cloud (containerisation Docker, CI/CD, monitoring) et optimiser les performances afin de supporter un plus grand nombre d'utilisateurs en production.
- **Extension fonctionnelle** : intégrer de nouvelles fonctionnalités, telles que la recommandation de métiers associés, des ressources pédagogiques personnalisées, ou un tableau de bord plus complet pour les conseillers d'orientation.

Ces perspectives permettront d'améliorer la précision des prédictions, de renforcer l'explicabilité des résultats et d'assurer une adoption plus large du système dans un contexte réel d'accompagnement à l'orientation.

# Bibliographie

- [1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.
- [2] J. R. Quinlan, *C4.5 : Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [3] F. Pedregosa et al., “Scikit-learn : Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [4] FastAPI Documentation, *FastAPI : Modern, fast (high-performance) web framework for building APIs with Python*. <https://fastapi.tiangolo.com/>
- [5] Streamlit Documentation, *Streamlit : The fastest way to build data apps in Python*. <https://docs.streamlit.io/>
- [6] MySQL Documentation, *MySQL 8.0 Reference Manual*. <https://dev.mysql.com/doc/>
- [7] Joblib Documentation, *Joblib : running Python functions as pipeline jobs*. <https://joblib.readthedocs.io/>
- [8] S. M. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [9] M. T. Ribeiro, S. Singh, and C. Guestrin, “*Why Should I Trust You ?* : Explaining the Predictions of Any Classifier,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.