



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

RAPPORT DE PROJET DE FIN D'ANNÉE

*4^{ème} 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

Chapitre I — Présentation du projet	3
1.1 Introduction	4
1.2 Contexte et sujet du projet	4
1.3 Problématique	4
1.4 Objectifs du projet	5
1.5 User Stories	5
1.5.1 Acteur : Étudiant	6
1.5.2 Acteur : Conseiller d'orientation	6
1.5.3 Acteur : Administrateur IA	6
1.5.4 Acteur : Utilisateur Web	6
1.6 Périmètre fonctionnel du projet	6
Conclusion	7
Chapitre II — Méthodologie de travail	8
2.1 Méthodologie Agile	9
2.2 Cadre Scrum	10
2.3 Les artefacts Scrum	10
2.4 Les cérémonies Scrum	10
2.5 Organisation des sprints et planification Scrum	11
2.5.1 Sprint 1 — Module étudiant et prédiction initiale	11
2.5.2 Sprint 2 — Fonctionnalités avancées et module Conseiller d'orientation	12
2.5.3 Sprint 3 — Gestion du modèle et supervision par l'administrateur IA .	13
Chapitre III — Conception du système d'orientation académique	14
3.1 Diagramme de cas d'utilisation	15
3.2 Diagramme de séquence	16
3.3 Diagramme d'activités	18
Chapitre IV — Architecture du Système et Pipeline d'Apprentissage Automatique	20
4.1 Architecture générale du système	21
4.1.1 Couche Présentation – Interface Web Streamlit	22
4.1.2 Couche Logique Métier – Backend FastAPI	22

4.1.3	Couche Données – MySQL et Stockage de fichiers	23
4.2	Justification du choix de l'architecture	23
4.3	Pipeline d'apprentissage automatique (IA)	25
Chapitre V — Réalisation et Implémentation		27

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 exprimés par les différents acteurs du système et permettent de définir les fonctionnalités attendues de manière simple et centrée utilisateur.

1.5.1 Acteur : Étudiant

- **US1** : En tant qu'étudiant, je veux saisir mes informations pour obtenir une orientation personnalisée.
- **US2** : Je veux voir plusieurs filières proposées avec un pourcentage de compatibilité.
- **US3** : Je veux visualiser mes points forts (logique, communication, créativité, etc.).
- **US4** : Je veux comprendre les critères ayant influencé ma prédiction.

1.5.2 Acteur : Conseiller d'orientation

- **US5** : Consulter les orientations générées par l'IA.
- **US6** : Filtrer les étudiants selon leur profil ou leur orientation.
- **US7** : Consulter les tendances globales des orientations.

1.5.3 Acteur : Administrateur IA

- **US8** : Importer ou mettre à jour le dataset.
- **US9** : Entraîner plusieurs modèles et comparer les performances.
- **US10** : Sauvegarder le meilleur modèle et les métriques associées.
- **US11** : Réentraîner le modèle en cas de mise à jour du dataset.

1.5.4 Acteur : Utilisateur Web

- **US12** : Accéder à une interface simple via Streamlit.
- **US13** : Soumettre mes données et obtenir une prédiction immédiate.
- **US14** : Visualiser les résultats sous forme de graphiques.
- **US15** : Télécharger un rapport PDF contenant l'analyse complète.

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.
- Espace conseiller : consultation des résultats avec filtres.

- Espace administrateur IA : gestion du dataset et entraînement du modèle.

Fonctionnalités exclues

- Application mobile (non prévue dans cette version).
- Système d'authentification avancé multi-rôle.
- API REST complète (non nécessaire dans la version Streamlit).
- Notifications automatiques (e-mail, SMS).

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 exposé les user stories représentant les attentes des différents acteurs impliqués.

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 (étudiant, conseiller, administrateur IA) ;
 - exigences fonctionnelles (collecte des données, prédition, 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 prédition 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 — Module étudiant et prédition initiale

Le premier sprint a été consacré à la construction du socle fonctionnel du projet. L'objectif principal était de permettre à l'étudiant de saisir ses informations et d'obtenir une première prédition générée par un modèle d'apprentissage automatique.

Durant ce sprint, nous avons mis en place :

- l'interface initiale permettant la saisie des données étudiantes;
- la génération d'une prédition 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 – Entrainer 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 module Conseiller d’orientation

Le second sprint a visé l’enrichissement fonctionnel du système en ajoutant les fonctionnalités destinées au conseiller d’orientation 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 pour les différents étudiants ;
- 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 des résultats pour le conseiller d’orientation.

2.5.3 Sprint 3 — Gestion du modèle et supervision par l'administrateur IA

Le troisième sprint s'est concentré sur les fonctionnalités dédiées à l'administrateur IA, afin d'assurer la maintenance et l'évolution continue du modèle. Ce sprint a introduit :

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

- 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 prédition, outils d'analyse pour le conseiller, puis supervision du modèle par l'administrateur IA. La méthode adoptée s'est ainsi révélée parfaitement adaptée aux exigences d'un projet d'Intelligence Artificielle 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

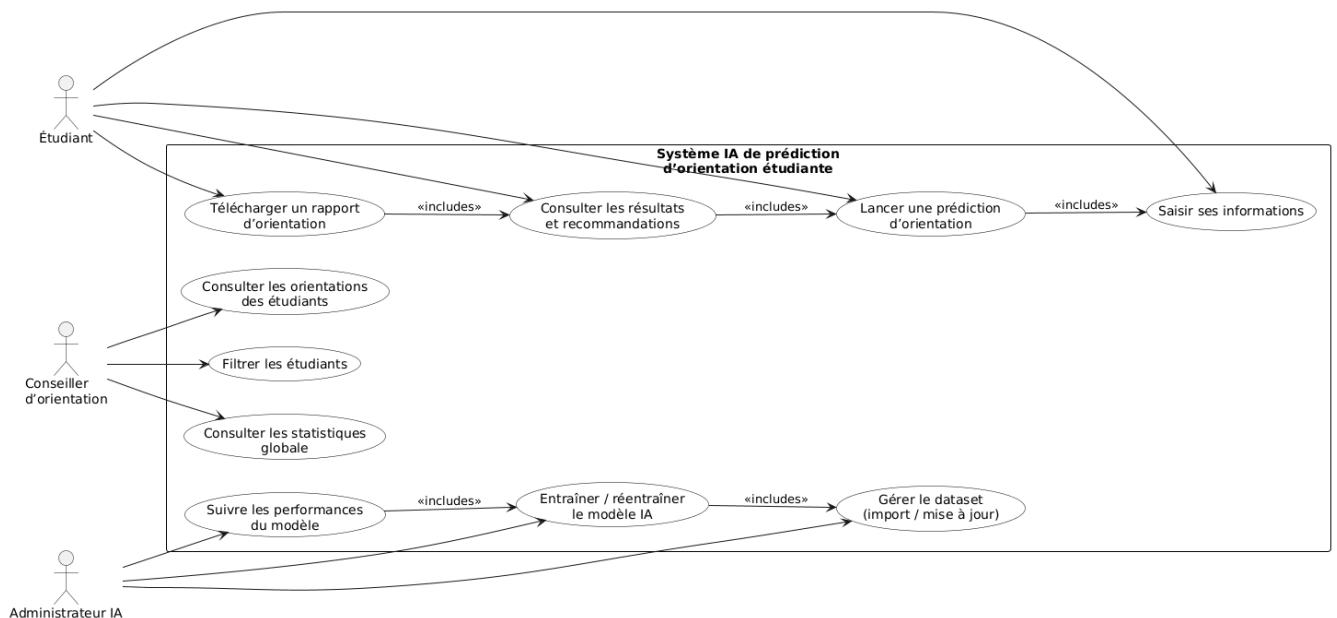


FIGURE 3.1 – Diagramme de cas d'utilisation du système d'orientation académique

>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 prédition, 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

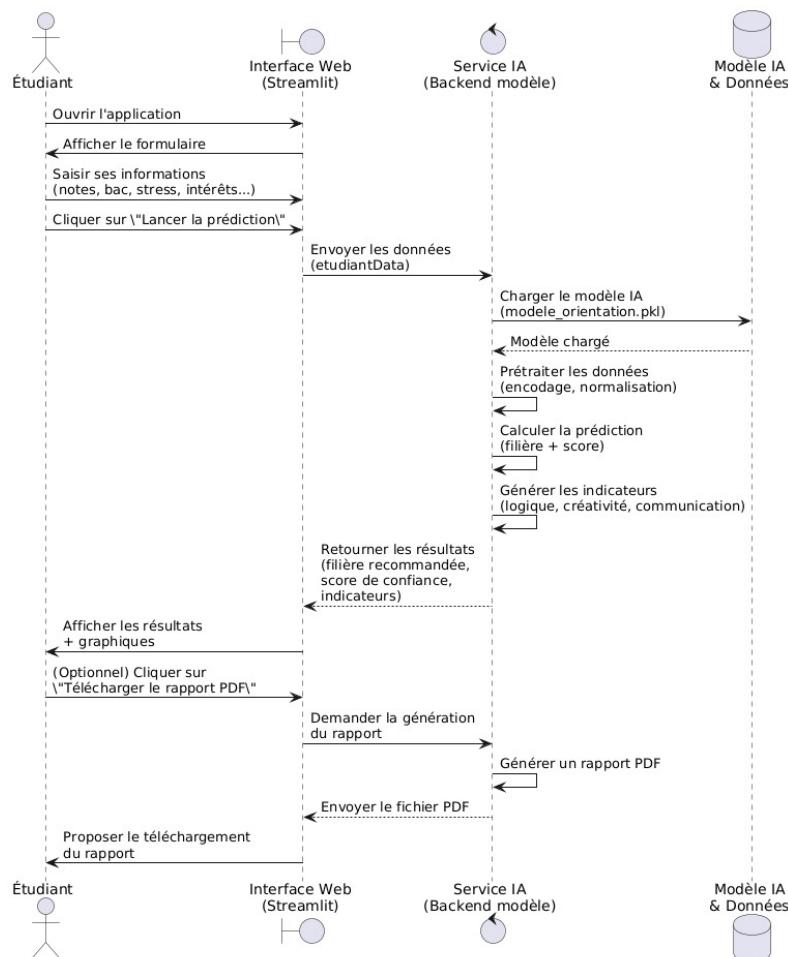


FIGURE 3.2 – Diagramme de séquence du processus de prédition d'orientation

->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

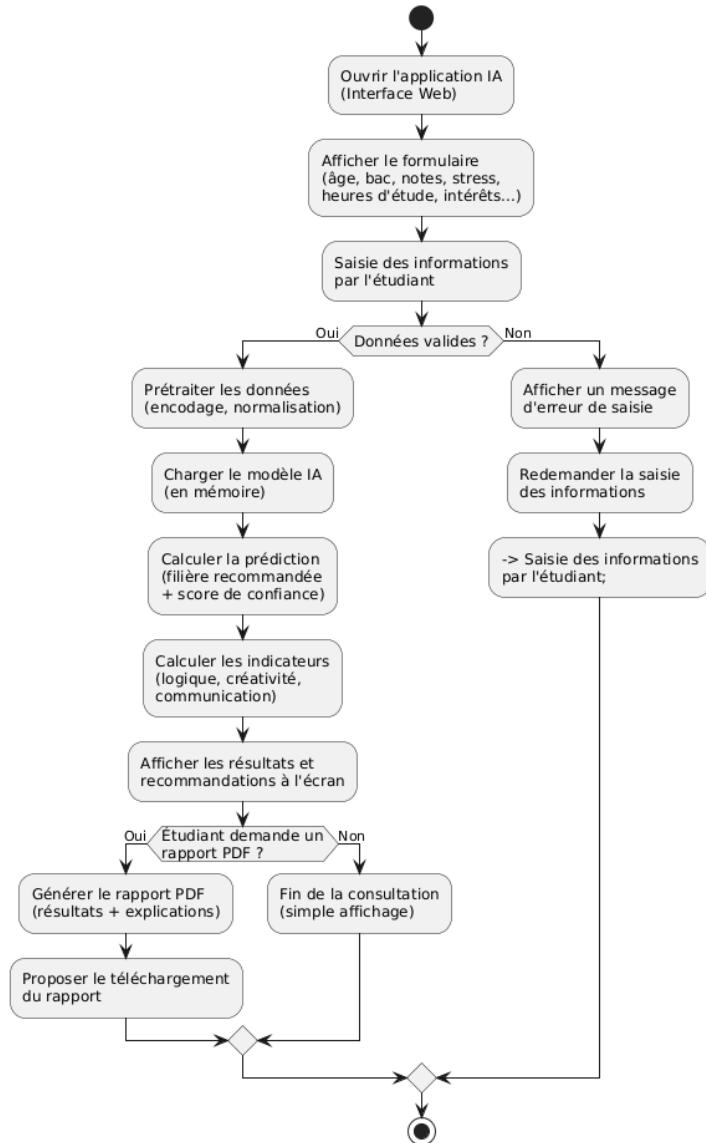


FIGURE 3.3 – Diagramme d'activités du processus de prédition d'orientation

->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.

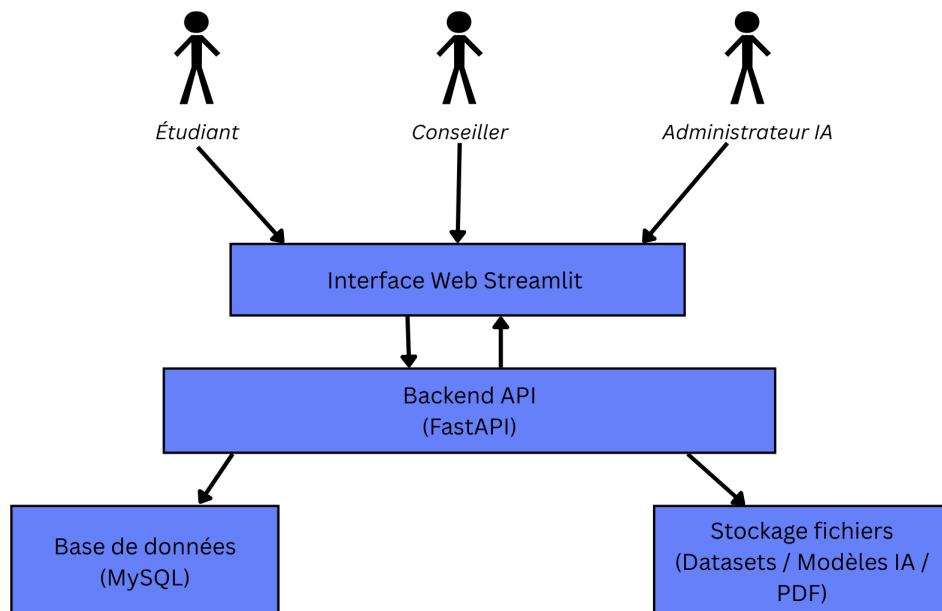


FIGURE 4.1 – Architecture générale du système d'orientation académique

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.



Technologies utilisées : 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.



Technologies utilisées : 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.



Technologies utilisées : 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

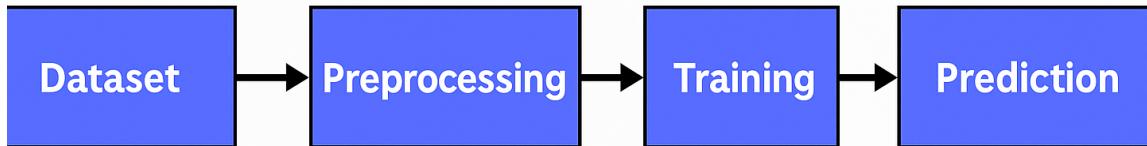


FIGURE 4.2 – Pipeline global du processus d'apprentissage automatique

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
