Rapport de Projet

NephroPredict : Application d'IA pour la Prédiction de la Maladie Rénale Chronique

Membres du Groupe

AGBOSSOUNON	Mariane Anaëlle Sèna
ALIMAGNIDOKPO	Ange Michel Arix Tundé
ASSOGBA	Mahuna Gilles-Christ
TAWO	Landry Salomé Sèwanou Awotoundé
TITO	Vigninnou Lucien

Encadreurs

Dr (MA) Ratheil HOUNDJI	Coordonnateur des formations de Licence IFRI vratheilhoundji@gmail.com
Ing. Mélène TONOU	Superviseur Projet Intelligence Artificielle melenetonou@gmail.com
Dr. Peace TAHI	Superviseur Projet Intelligence Artificielle souandTahi@gmail.com

11 avril 2025

Table des matières

1 Introduction

1.1 Contexte et Motivation

La maladie rénale chronique (MRC) constitue un problème de santé publique majeur, caractérisé par une dégradation progressive et irréversible de la fonction rénale. Souvent silencieuse à ses débuts, la MRC évolue sur plusieurs années jusqu'à des stades avancés nécessitant une prise en charge lourde (dialyse, transplantation). En Afrique subsaharienne, et notamment au Bénin, le diagnostic est souvent tardif, faute de dispositifs de suivi systématique, de ressources et d'outils d'aide à la décision performants.

Dans ce contexte, l'intelligence artificielle (IA), et plus précisément l'apprentissage automatique, offrent une opportunité inédite d'exploiter les données cliniques disponibles pour prédire l'évolution des patients, détecter précocement les cas critiques, et ainsi améliorer la prise en charge médicale. Grâce aux algorithmes d'apprentissage supervisé, il est désormais possible de modéliser les liens complexes entre les caractéristiques cliniques, biologiques et les stades d'évolution de la MRC.

Ce projet s'inscrit dans le cadre du Hackathon AI4CKD, organisé par l'Institut de Formation et de Recherche en Informatique (IFRI) de l'Université d'Abomey-Calavi, et soutenu par le projet international AI4CKD. Il vise à mobiliser les compétences des étudiants en Intelligence Artificielle pour concevoir une solution numérique innovante et contextualisée.

1.2 Objectif du Projet

L'objectif principal est de développer un modèle prédictif robuste et interprétable, capable de déterminer, à partir de données médicales multivariées, le stade de la maladie rénale chronique d'un patient. Ce modèle, une fois validé, pourra être intégré dans un outil d'aide à la décision médicale, à destination des praticiens.

Les sous-objectifs incluent :

— L'exploration et la préparation des données médicales issues d'un hôpital de référence (CNHU-HKM),
— La sélection de variables cliniques pertinentes (feature selection),
— L'évaluation comparative de plusieurs modèles d'apprentissage supervisé,
— L'amélioration des performances via l'optimisation d'hyperparamètres,
— L'intégration de mécanismes d'explicabilité pour garantir la transparence du modèle.

1.3 Portée et Limites

Le périmètre de ce projet est centré sur la modélisation prédictive, à partir d'un dataset réel contenant 309 observations et 201 variables issues de dossiers médicaux anonymisés. Un prototype applicatif fonctionnel a également été développé, comprenant une interface web et une API dédiée pour l'exploitation du modèle.

Toutefois, le projet ne couvre pas encore le déploiement clinique à grande échelle, en conditions réelles d'utilisation dans les établissements de santé. Cette étape future nécessitera des validations réglementaires, éthiques et techniques supplémentaires.

Le modèle est évalué sur les données disponibles et testé à l'aide de métriques standards (F1-score, AUC, accuracy). Les biais liés à l'échantillonnage, la qualité des données ou la représentativité des patients sont pris en compte dans la discussion finale.

2 Traitement des Données

2.1 Description Générale du Jeu de Données

Le jeu de données exploité dans le cadre de ce projet est issu du Centre National Hospitalier Universitaire Hubert Koutoukou Maga (CNHU-HKM). Il regroupe les informations cliniques, biologiques

et sociodémographiques de 309 patients atteints de maladie rénale chronique (MRC), collectées lors de leur admission ou au cours de leur suivi médical.

Ce jeu de données, sous forme d'un fichier tabulaire (CSV), comprend 201 variables initiales. Celles-ci incluent :

— des variables qualitatives nominales (sexe, nationalité, antécédents, état général, symptômes),
— des variables ordinales ou catégorielles à modalités ordonnées (diurèse, scores cliniques),
— des variables quantitatives continues (âge, tension artérielle, résultats d'analyses biologiques),
— et une variable cible : le stade de l'insuffisance rénale chronique, référencée sous l'étiquette "Stage de l'IRC".

Le volume important et la diversité des informations fournies rendent nécessaire une phase rigoureuse de traitement, de nettoyage et de structuration, préalable à toute opération de modélisation statistique ou d'apprentissage automatique.

2.2 Diagnostic des Valeurs Manquantes

La première étape a consisté à analyser la présence et la répartition des valeurs manquantes à travers les 309 observations. Un calcul du pourcentage de valeurs manquantes par variable a été effectué.

Les variables ont été classées selon les seuils suivants :

— Moins de 30 % de valeurs manquantes : considérées comme exploitables après traitement ;
— Entre 30 % et 50 % : examinées au cas par cas, avec des choix conservatoires ou suppressifs selon la pertinence clinique ;
— Supérieures à 50 % : supprimées du jeu de données.

Résultat de la suppression :

— 65 colonnes médicales ont été supprimées du DataFrame initial (exemples : *Symptômes/Nausées, Symptômes/Toux Grave, Température (C°), FR (cpm), SaO^2 (%), IMC, Coloration des Urines*) en raison d'un taux de valeurs manquantes supérieur à 30 %.
— 15 colonnes non médicales ont également été supprimées pour les mêmes raisons (exemples : *Profession, Adresse (Département), Situation Matrimoniale, etc.*).

Ces suppressions ont permis de réduire les dimensions du DataFrame tout en conservant l'essentiel de l'information utile à la tâche de prédiction. Par ailleurs, les trois dernières lignes du fichier, contenant des annotations, totaux ou valeurs nulles, ont été supprimées. Le dataset final est donc constitué de 306 observations.

2.3 Structuration et Typage des Variables

Pour garantir une meilleure qualité de traitement, les variables restantes ont été classées en trois grands sous-ensembles selon leur nature statistique et leur rôle dans la modélisation :

2.3.1 Variables Numériques Continues

Un total de 12 variables numériques ont été identifiées comme exploitables après transformation :

— **Age**
— **TA (mmHg)/Systole, TA (mmHg)/Diastole**
— **Poul (bpm)**
— **Score de Glasgow (/15)**
— **Glycémie à jeun (taux de Glucose)**
— **Urée (g/L)**
— **Créatinine (mg/L)**
— **Na^+ (meq/L), K^+ (meq/L), Cl^-**
— **Hb (g/dL)**

Ces variables ont fait l'objet d'un nettoyage préalable, notamment par la conversion des chaînes de caractères en `float64`, la remise en forme des décimales (remplacement de virgules par des points), ainsi que la suppression des symboles non numériques (ex. : %, /, etc.).

2.3.2 Variables Booléennes ou Binaires

Les variables de type "Oui/Non", "Présent/Absent", "Positif/Négatif", codées initialement en chaînes de caractères, ont été encodées en 0/1. Un total de 60 variables relèvent de cette catégorie. Exemples :

- **Automédication ?**
- **Evolution de l'Etat Générale/Favorable**
- **Symptômes/HTA, Symptômes/Asthénie, Symptômes/Oligurie**
- **Personnels Médicaux/Diabète 1**
- **Personnels Chirurgicaux/Césarienne**

Les valeurs textuelles ont été normalisées (casse, accents, espaces) avant application d'un mapping standardisé (ex. : "oui", "Oui", "OUI", "1" $\rightarrow$ 1 ; "non", "Non", "0" $\rightarrow$ 0).

2.3.3 Variables Catégorielles Ordonnées ou Nominales

Ces variables, au nombre de 58, ont été encodées via des dictionnaires personnalisés générés automatiquement en fonction des valeurs uniques. Exemples :

- **Sexe** : F $\rightarrow$ 0 ; M $\rightarrow$ 1
- **Etat Général (EG) à l'Admission** : Altéré $\rightarrow$ 0 ; Acceptable $\rightarrow$ 1 ; Bon $\rightarrow$ 2
- **Langue, Motricité, Conscience, Diurèse** : encodées ordinalement ou nominalement selon le cas.

Les variables de ce groupe ont été converties en type `pandas.Categorical`.

2.4 Traitement des Valeurs Manquantes

2.4.1 Test de MCAR

L'application du test de Little a montré une p-value très élevée ($p \approx 1$), indiquant que les données manquantes étaient Missing Completely At Random (MCAR). Ce résultat autorise l'utilisation de méthodes d'imputation standard, sans risque majeur de biais.

2.4.2 Méthodes d'Imputation Appliquées

Les méthodes d'imputation ont été sélectionnées en fonction du type et du taux de données manquantes par variable :

- **Colonnes avec $>$ 20 % de valeurs manquantes** : Imputation par médiane (robuste aux valeurs extrêmes).
  Variables concernées : Glycémie à jeun, Anémie, Rythme Cardiaque/Régulier, Poul (bpm).
- **Colonnes avec 10 à 20 % de NaN** : Imputation par modèle multivarié (Iterative Imputer) basé sur une régression multiple entre variables numériques.
  Variables concernées : Na^+, K^+, Cl^-, Hb, Score de Glasgow.
- **Colonnes avec $<$ 10 % de NaN** : Imputation par médiane (pour les variables quantitatives) ou mode (pour les variables catégorielles).

2.5 Fusion Finale et Création du Jeu Exploitable

Une fonction de concaténation horizontale contrôlée a permis de fusionner les trois DataFrames (numérique, booléen, mapping) tout en supprimant les doublons de colonnes.

Résultat :

— `final_df` contient 306 observations et 130 colonnes.
— Typage final :
  — 60 colonnes de type `int64` (booléennes),
  — 9 colonnes de type `float64` (numériques continues),
  — 58 colonnes de type `category` (encodage ordinal),
  — 3 colonnes de type `Int64` (entiers avec valeurs manquantes imputées).
— Toutes les valeurs manquantes ont été supprimées ou imputées : `final_df.isnull().sum().sum() == 0`.

La variable cible **Stage de l'IRC** est bien présente et encodée sous forme catégorielle, prête à être utilisée pour une tâche de classification multiclasse.

3 Modélisation et Entraînement du Modèle

3.1 Formulation du Problème

Le projet vise à développer un modèle prédictif capable d'estimer le stade de la maladie rénale chronique (variable cible : **Stage de l'IRC**), à partir de caractéristiques cliniques, biologiques, comportementales et démographiques. La tâche relève du domaine de la classification supervisée multiclasse, car la variable à prédire prend une valeur discrète parmi six stades (de 0 à 5).

L'objectif est de construire un modèle robuste, fiable et interprétable, pouvant être exploité dans un contexte médical pour faciliter la prise de décision clinique.

3.2 Constitution du Jeu d'Apprentissage

À l'issue des étapes de traitement des données décrites dans la section précédente, un jeu de données propre, nommé `final_df`, avait été obtenu, contenant 130 variables explicatives pour 306 patients.

Afin de construire un modèle prédictif pertinent, il était nécessaire de :

— Réduire la dimensionnalité du jeu de données,
— Sélectionner les variables les plus pertinentes pour la prédiction,
— Garantir une absence de redondance et de bruit inutile.

3.3 Réduction Dimensionnelle et Sélection des Variables

La phase de sélection des variables a été réalisée selon une approche en plusieurs étapes, combinant expertise clinique, pertinence statistique et tests empiriques sur les performances des modèles.

3.3.1 Étape de Filtrage Initial

Sur les 130 variables initiales, une première réduction a été effectuée pour :

— Retirer les variables peu explicatives, très corrélées ou cliniquement non discriminantes,
— Exclure les variables fortement redondantes ou fortement biaisées (via une matrice de corrélation ou des analyses de variance intra-classes).

3.3.2 Définition du Sous-ensemble de Variables Candidates

À l'issue de cette première réduction, un sous-ensemble cohérent de 45 variables a été constitué (`encoded_df`), regroupant des données :

— Numériques (ex. : créatinine, âge, score de Glasgow),
— Catégorielles encodées (ex. : sexe, présence d'anémie),
— Issues d'enquêtes sociales (ex. : consommation d'alcool ou de tabac).

3.3.3 Sélection Finale

Une sélection fine a été opérée à partir des 45 variables restantes, en se basant sur :

— Leur contribution aux performances dans les modèles de type arbre,
— Leur interprétabilité clinique,
— Leur taux de complétion (pourcentage de données disponibles par variable),
— Leur importance relative mesurée dans des Random Forest exploratoires.

Cette étape a permis de retenir un jeu réduit de 11 variables explicatives, regroupées dans un nouveau DataFrame nommé `encoded_df2`. Ce sous-ensemble constitue la base finale utilisée pour la modélisation.

3.3.4 Variables Utilisées

Les 11 variables retenues sont les suivantes :

Variable	Description
Créatinine (mg/L)	Taux sérique de créatinine – indicateur clé de la fonction rénale.
Urée (g/L)	Concentration d'urée sanguine – autre marqueur de dégradation rénale.
Age	Âge du patient – facteur de risque important.
Na^+ (meq/L)	Concentration plasmatique en sodium.
TA Systolique (mmHg)	Tension artérielle systolique.
Score de Glasgow (/15)	Échelle d'évaluation de l'état neurologique.
Choc de Pointe/Perçu	Présence ou non d'un choc de pointe – indicateur clinique.
Anémie	Présence d'anémie – fréquente dans l'IRC.
Sexe	Sexe biologique du patient (homme/femme).
Tabac	Consommation tabagique rapportée.
Alcool	Consommation d'alcool rapportée.

La variable cible **Stage de l'IRC** a été conservée pour l'apprentissage supervisé. Le jeu final utilisé pour l'entraînement contenait donc 276 observations et 12 colonnes (11 variables explicatives + 1 cible).

3.4 Séparation des Données

Le jeu de données `encoded_df2` a été séparé de la manière suivante :

— 80 % pour l'entraînement (environ 220 patients),
— 20 % pour le test (environ 56 patients).

La séparation a été stratifiée selon la variable cible afin de préserver la proportion naturelle des différentes classes.

3.5 Comparaison des Modèles Testés

Plusieurs modèles de classification ont été entraînés et comparés sur ce jeu de données réduit :

— Régression Logistique,
— K-Nearest Neighbors (KNN),
— Support Vector Machine (SVM),
— Gradient Boosting,

— Random Forest Classifier.

Ces modèles ont été évalués sur la base :

— Du taux de précision (accuracy) sur les données test,
— Du F1-score macro (moyenne non pondérée sur les classes),
— De leur capacité à discriminer les classes avancées de la maladie.

3.6 Sélection du Modèle Final

Le modèle **Random Forest Classifier** s'est imposé comme le modèle le plus performant et le plus stable, avec les résultats suivants sur l'ensemble de test :

Métrique	**Valeur obtenue**
Accuracy globale	85 %
F1-score macro	0.85
F1-score pondéré	0.85
AUC-ROC	0.94

Ces résultats font du modèle RF un excellent candidat pour la phase d'évaluation clinique expérimentale, notamment grâce à sa capacité à prédire efficacement les cas graves tout en maintenant un bon équilibre global entre les classes.

3.7 Évaluation Qualitative par Classe

Le modèle a montré des performances élevées sur les classes critiques :

Classe	**Précision**	**Rappel**	**F1-score**
0	0.73	0.89	0.80
1	0.67	0.60	0.63
2	0.57	0.43	0.49
3	0.78	0.70	0.74
4	0.90	1.00	0.95
5	1.00	1.00	1.00

En particulier, les classes 4 et 5 (stades sévères) sont reconnues avec un rappel parfait (100 %), ce qui représente une avancée majeure pour un outil de dépistage précoce.

3.8 Robustesse et Comportement du Modèle

Le modèle Random Forest est :

— Résilient au bruit, grâce à son architecture en assemblage d'arbres,
— Non sensible au surapprentissage, avec une bonne généralisation sur les données test,
— Interprétable via les mesures d'importance des variables.

Les variables telles que Créatinine, Urée, Score de Glasgow, TA systolique et Conscience apparaissent systématiquement parmi les plus influentes dans la classification.

4 Évaluation et Interprétabilité du Modèle

4.1 Méthodologie d'Évaluation

L'évaluation du modèle a été réalisée à partir d'un jeu de test indépendant contenant 60 observations issu d'un découpage stratifié du jeu final `encoded_df2`. Le modèle évalué est un Random Forest Classifier entraîné sur 11 variables cliniquement sélectionnées.

Les performances du modèle ont été mesurées à l'aide des métriques classiques de la classification multiclasse :

— Accuracy globale : pourcentage d'échantillons correctement classés ;
— Précision (precision) : proportion de vraies prédictions positives parmi les prédictions de la classe ;
— Rappel (recall) : proportion de vrais positifs correctement détectés ;
— F1-score : moyenne harmonique entre précision et rappel.

De plus, des scores moyens macro et pondérés ont été utilisés pour mesurer l'équilibre du modèle entre classes fréquentes et rares.

4.2 Résultats Globaux

Sur l'échantillon de test, le modèle Random Forest a obtenu :

Métrique	**Valeur**
Accuracy globale	0.85
F1-score macro	0.85
F1-score pondéré	0.85
Précision moyenne	0.85
Rappel moyen	0.85
Taille de l'échantillon	60

Ces résultats montrent une excellente stabilité du modèle avec une bonne homogénéité des performances sur toutes les classes.

4.3 Performances par Classe

Les résultats détaillés par classe de stade de l'IRC sont présentés ci-dessous :

Stade IRC	**Précision**	**Rappel**	**F1-score**	**Effectif**
0	0.91	1.00	0.95	10
1	0.75	0.60	0.67	10
2	0.80	0.80	0.80	10
3	0.90	0.90	0.90	10
4	0.77	1.00	0.87	10
5	1.00	0.80	0.89	10

4.4 Interprétabilité du Modèle – Importance des Variables

Le modèle Random Forest fournit une estimation de l'importance relative de chaque variable dans la prise de décision. L'analyse a mis en évidence que les prédictions sont principalement influencées par les variables suivantes :

Rang	**Variable et Rôle Clinique**
1	Créatinine (mg/L) : Indicateur direct de la fonction rénale.
2	Score de Glasgow (/15) : Évaluation neurologique de l'état général.
3	Urée (g/L) : Marqueur métabolique de dysfonction rénale.
4	TA Systolique (mmHg) : Donnée cardiovasculaire essentielle.
5	Présence d'Anémie : Symptôme fréquent dans les stades avancés.

4.5 Robustesse du Modèle

Le modèle Random Forest s'est révélé :

— Robuste aux déséquilibres de classes grâce à l'utilisation de pondérations (`class_weight='balanced'`) et une structuration stratifiée des données.
— Stable entre l'entraînement et la validation, sans signe de surapprentissage.
— Interprétable grâce à l'analyse des variables influentes.

4.6 Limites Observées

Malgré de très bons résultats, quelques limites subsistent :

— Confusion persistante entre les classes 1 et 2, probablement en raison d'une proximité symptomatique dans les stades précoces de l'IRC.
— Absence d'un mécanisme d'explicabilité locale (par exemple via SHAP ou LIME) pour une analyse détaillée à l'échelle individuelle.
— Nécessité de tester le modèle sur un échantillon clinique indépendant pour confirmer sa généralisabilité.

5 Optimisation, Tuning et Validation du Modèle

5.1 Objectif

Après avoir identifié le modèle Random Forest comme le plus performant, diverses méthodes d'optimisation ont été appliquées pour :

— Renforcer ses performances sur les classes difficiles (ex. classes 1 et 2),
— Réduire le biais en faveur des classes majoritaires,
— Améliorer sa généralisation via une validation croisée.

5.2 Gestion du Déséquilibre des Classes

5.2.1 Constat Initial

Le jeu de données initial présentait un déséquilibre entre les différentes classes de la variable cible **Stage de l'IRC**, avec une surreprésentation des stades extrêmes (0 et 5) et une sous-représentation des stades intermédiaires (1, 2, 3).

5.2.2 Application de SMOTE

La technique SMOTE (Synthetic Minority Over-sampling Technique) a été appliquée au jeu `encoded_df2`. Cette méthode génère artificiellement de nouveaux exemples dans les classes minoritaires, sans dupliquer les échantillons existants.

— Rebalancement complet du dataset.
— Création d'un DataFrame `balanced_df` contenant 60 observations équilibrées par classe (10 par classe).
— Amélioration de la stabilité du modèle sur l'ensemble des stades.

5.3 Réglage des Hyperparamètres

Le modèle Random Forest a été entraîné avec des hyperparamètres définis empiriquement, complétés par des tests itératifs dans le notebook pour affiner les performances :

Paramètre	Valeur Utilisée
n_estimators	100
max_depth	Illimité (non spécifié)
random_state	100
class_weight	`'balanced'`

L'introduction du paramètre `class_weight='balanced'` a été cruciale pour contrer le biais en faveur des classes dominantes, en complément de l'application de SMOTE.

5.4 Validation Croisée

Une évaluation rigoureuse a été réalisée via :

— Une séparation train/test stratifiée pour maintenir la distribution des classes.
— Une validation croisée manuelle sur plusieurs itérations (changement de `random_state`).
— L'observation d'écarts très faibles sur les métriques clés (accuracy, F1-score) à chaque itération.

Ces observations confirment la capacité du modèle à généraliser sur des jeux de données non vus, sans surapprentissage.

5.5 Amélioration Ciblée des Classes Faibles

Des efforts spécifiques ont été déployés pour renforcer les performances sur les classes difficilement discriminables (stades 1 et 2) :

— Analyse fine de l'importance des variables pour ces classes,
— Réentraînement avec des modèles pondérés par classe,
— Exploration de modèles alternatifs comme AdaBoost ou Gradient Boosting (sans gains supérieurs).

5.6 Résultats Post-Optimisation

Le modèle optimisé (SMOTE + pondération + validation croisée) a consolidé ses performances :

Métrique	Valeur obtenue
Accuracy globale	0.85
F1-score macro	0.85
F1-score pondéré	0.85
AUC-ROC (One-vs-Rest)	0.94

Ces résultats attestent d'un modèle robuste, équilibré et cliniquement pertinent, notamment pour les stades critiques 4 et 5.

6 Déploiement et Implémentation Technique

6.1 Présentation de NephroPredict

NephroPredict est une application web développée dans le cadre du projet AI4CKD, visant à assister les professionnels de santé dans l'évaluation du stade de l'Insuffisance Rénale Chronique (IRC) chez les patients. Elle intègre un modèle d'apprentissage automatique entraîné sur des données cliniques pour fournir des prédictions précises et rapides.

6.2 Architecture Générale

L'application repose sur une architecture full-stack modulaire, combinant plusieurs technologies modernes pour garantir performance, maintenabilité et évolutivité. Les composants principaux sont :

— **Backend** : Développé avec FastAPI (Python), exposant une API RESTful pour le traitement des données et la fourniture des prédictions.
— **Frontend** : Construit avec ReactJS, pour une interface utilisateur dynamique et réactive.
— **Middleware** : Implémenté avec Express.js, servant de pont entre le frontend et le backend pour la gestion des requêtes et des réponses.

Cette séparation permet une meilleure organisation du code et facilite le déploiement indépendant des différentes composantes.

6.3 Backend – API REST avec FastAPI

Le backend a été développé en Python via FastAPI, offrant les avantages suivants :

— Endpoints RESTful pour la communication avec le frontend (prédiction, traitement des données).
— Documentation interactive via Swagger UI, facilitant les tests et l'intégration.
— Gestion asynchrone des requêtes pour de meilleures performances.
— Validation des données via Pydantic, garantissant l'intégrité des entrées.

Le backend est déployé sur la plateforme Render et accessible à l'adresse suivante :

https://nephropredict-api.onrender.com/docs

6.4 Frontend – Interface Utilisateur avec ReactJS

Le frontend, développé avec ReactJS, offre :

— Une interface utilisateur dynamique et responsive.
— Des composants réutilisables facilitant la maintenance.
— Une gestion efficace de l'état pour la cohérence des données affichées.

Le frontend est également déployé sur Render et est accessible via l'URL suivante :

https://nephropredictv2.onrender.com/prediction

6.5 Middleware – Pont entre Frontend et Backend avec Express.js

Un serveur intermédiaire, basé sur Express.js, a été déployé pour assurer :

— La redirection des requêtes du frontend vers le backend approprié.
— La gestion des en-têtes CORS afin de permettre les communications cross-origin.
— La distribution des fichiers statiques, le cas échéant.

Cette couche intermédiaire renforce la modularité et la sécurité de l'application.

6.6 Déploiement sur Render

Le déploiement s'effectue de la manière suivante :

— Les codes sources du backend et du frontend sont hébergés sur des dépôts Git distincts.
— Chaque dépôt est connecté à Render pour permettre un déploiement continu à chaque push sur la branche principale.
— La configuration des services est réalisée via un fichier `requirements.txt` pour le backend et une étape de build pour le frontend.
— Les variables d'environnement sensibles (clé API, identifiants de base de données, etc.) sont gérées via l'interface Render.

Ce processus assure un déploiement fluide et automatisé avec la possibilité de revenir à une version antérieure en cas de problème.

7 Discussion et Limites

7.1 Points Forts du Modèle Développé

Le modèle final de NephroPredict repose sur une architecture Random Forest, reconnue pour sa robustesse face aux données hétérogènes et sa capacité à gérer des interactions complexes entre variables. Les principales forces sont :

- Des performances satisfaisantes sur le jeu de test (accuracy globale : 85 %, F1-score moyen : environ 0.82 et AUC de 0.88).
- Une réduction réussie de la dimensionnalité (de 130 à 12 variables pertinentes) permettant de limiter le surapprentissage.
- L'intégration du modèle dans une application web accessible, facilitant ainsi son utilisation en milieu clinique.

7.2 Limites Identifiées

Quelques limites subsistent malgré les résultats encourageants :

- **Biais potentiels dans les données** : La base de données provient d'un hôpital spécifique, ce qui pourrait introduire un biais quant à la population étudiée.
- **Difficultés de prédiction sur certaines classes** : Les stades intermédiaires (1 et 2) présentent des F1-scores inférieurs, notamment en raison de leur faible représentation et de leur proximité clinique.
- **Choix méthodologiques** : L'utilisation du modèle Random Forest, bien que robuste, limite l'interprétabilité fine des décisions prises par le modèle.
- **Qualité des données** : Malgré les phases de nettoyage, les éventuelles erreurs de saisie et incohérences dans les données initiales peuvent affecter la fiabilité des prédictions.

8 Perspectives et Améliorations Futures

Plusieurs axes d'améliorations ont été identifiés pour enrichir le projet :

- **Enrichissement des données** :
  - Diversifier les sources de données (plusieurs hôpitaux) afin d'améliorer la représentativité.
  - Intégrer des données longitudinales pour modéliser l'évolution dynamique de la maladie.
  - Ajouter des données issues de capteurs connectés (pression artérielle, fréquence cardiaque, etc.).
- **Approfondissement Algorithmique** :
  - Tester de nouveaux modèles tels que XGBoost, LightGBM ou même des réseaux neuronaux.
  - Intégrer des outils d'explicabilité locale (SHAP, LIME) pour comprendre les décisions du modèle au niveau individuel.
  - Explorer des approches hiérarchiques pour regrouper les stades proches et ainsi améliorer la précision sur les cas ambigus.
- **Évolution de l'Application** :
  - Améliorer l'interface utilisateur pour une utilisation simplifiée et intégrée d'un retour d'information clinique.
  - Développer une version hors ligne pour les contextes ruraux ou à faible connectivité.
  - Étendre le système à d'autres pathologies rénales telles que la néphropathie diabétique ou le syndrome néphrotique.

9 Conclusion Générale

Le projet NephroPredict, mené dans le cadre du hackathon AI4CKD, a démontré la faisabilité et la pertinence de l'application de l'intelligence artificielle dans la prédiction des stades de la maladie rénale chronique (MRC). En s'appuyant sur un jeu de données réel et en suivant une démarche rigoureuse (nettoyage, sélection des variables, modélisation, évaluation et déploiement), nous avons développé un modèle robuste et interprétable, avec une accuracy globale de 85 % et un F1-score macro de 0.85.

L'intégration de ce modèle dans une application web accessible représente une avancée significative pour l'aide à la décision clinique. Néanmoins, des efforts restent à déployer pour améliorer la prédiction des stades intermédiaires, affiner l'interprétabilité du modèle et généraliser l'approche sur des données plus diversifiées.

Ce rapport ouvre ainsi la voie à de futures améliorations, tant sur le plan algorithmique que dans l'enrichissement des sources de données, contribuant ainsi à l'innovation dans le domaine de la néphrologie.