RAPPORT TECHNIQUE

Système de Stratification Probabiliste des Réponses Ovariennes en Fécondation In Vitro

Code source GitHub : https://github.com/YesmineZhioua/Tanit$_ML - Data.git$

7 décembre 2025

Table des matières

1 INTRODUCTION ET CONTEXTE

1.1 Contexte du Projet

La fécondation in vitro (FIV) représente une technologie médicale complexe où la prédiction de la réponse ovarienne aux stimulations hormonales constitue un défi clinique majeur. Ce projet vise à développer un système de classification probabiliste permettant de stratifier les patientes selon leur réponse attendue : faible (low), optimale (optimal) ou élevée (high).

1.2 Objectifs du Projet

Objectif Principal

Développer un modèle de classification interprétable capable de prédire avec précision la réponse ovarienne des patientes, en fournissant des probabilités associées à chaque classe (ex : «68% de chance que cette patiente soit high responsive»).

Objectifs Secondaires

— Extraire et structurer les données médicales non structurées (PDF)
— Anonymiser les données patients selon les normes RGPD
— Identifier les biomarqueurs prédictifs significatifs
— Développer une interface d'inférence utilisable cliniquement
— Assurer l'interprétabilité des prédictions via SHAP/LIME

1.3 Enjeux Cliniques

La stratification précoce et précise des patientes permet :

Optimisation thérapeutique : Adaptation des protocoles de stimulation selon le profil
Réduction des risques : Prévention du syndrome d'hyperstimulation ovarienne (SHSO)
Amélioration des résultats : Augmentation des taux de réussite par cycle
Personnalisation des soins : Traitement individualisé basé sur les biomarqueurs
Optimisation économique : Réduction des coûts et du temps de traitement

2 DESCRIPTION DU DATASET ET DES VARIABLES

2.1 Sources de Données

Données Structurées : Fichier CSV contenant les dossiers des patients.
Données Non Structurées : Document PDF représente les données médical d'un patient.

2.2 Variables du Dataset

Variable	Type	Description Clinique
Patient_id	Catégoriel	Identifiant anonymisé unique
Cycle_number	Numérique entier	Numéro de tentative de FIV (historique)
Age	Numérique	Âge de la patiente au moment du cycle
Protocol	Catégoriel	Type de protocole de stimulation ovarienne : fixed antagonist / flexible antagonist / agonist
AMH	Numérique continu	Hormone Anti-Müllérienne (ng/mL)
N_Follicles	Numérique entier	Nombre de follicules au dernier jour monitoré
E2_day5	Numérique continu	Niveau d'estradiol au jour 5 de stimulation
AFC	Numérique entier	Compte de follicules antraux à l'échographie
Patient_Response	Catégoriel (CIBLE)	Stratification de la réponse ovarienne :low / optimal / high

TABLE 1 – Variables du dataset

3 Architecture Global

PDF → Extraction → EDA → Nettoyage → Entraînement → Évaluation → Déploiement .

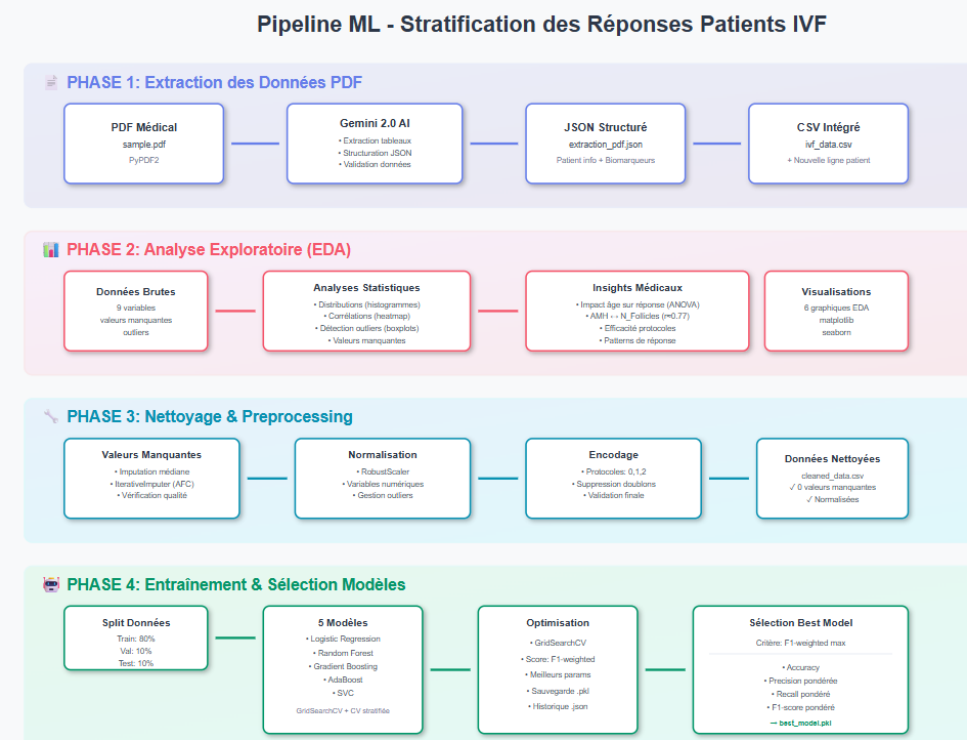

FIGURE 1 – Pipeline

3.1 Technologie utilisées

Extraction : PyPDF2 , Google Gemini 2.0 AI
Traitement : Pandas , Numpy , Scikit-learn
visualisation : Matplotlib , Seaborn
Modélisation : Scikit-learn (Logistic Regression, Random Forest, Gradient Boosting, AdaBoost, SVC)
Back : Python , Flask
Front : Streamlit

3.2 Architecture du Projet

4 Phase1 : EXTRACTION DES DONNÉES PDF

4.1 Les etapes de l'extraction

L'extraction des données du document PDF médical nécessite une approche hybride combinant :

— **Etape 1 : Extraction du texte brut avec PyPDF2**
— **Etape 2 : Structuration intelligente avec Gemini 2.0 :**
Nous avons utilisé Google Gemini 2.0 avec un prompt engineering précis pour identifier

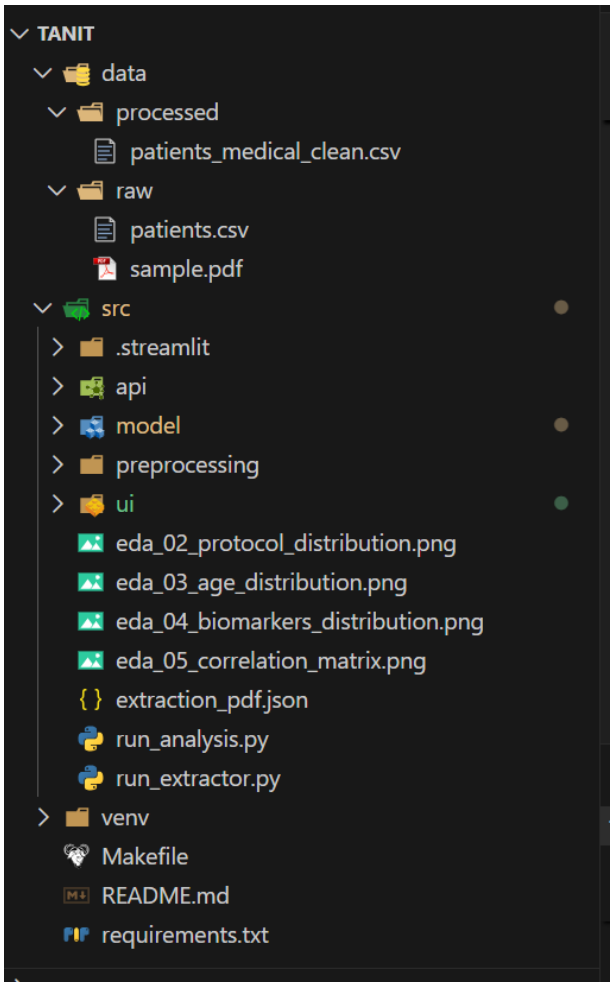

FIGURE 2 – Architecture du code source

les en-têtes de tableaux, Extraire les valeurs exactes(préservation des formats "10/8", "225UI") et générer un JSON structuré validé

— **Étape 3 : Validation et nettoyage du JSON**

— **Étape 4 : Intégration dans le CSV :**
Mapping intelligent des données extraites vers les colonnes du dataset :
-Génération automatique du format patientID en (format 25XXX)
-Extraction depuis "2nd " –> 2
-Calcul age depuis date de naissance
-Standariser le Protocol

4.2 Structuration et Intégration

Une fois les données extraites du PDF :

1. Conversion en format structuré (CSV/JSON)
2. Validation du schéma de données (types, plages)
3. Ajout d'une nouvelle ligne au CSV existant avec tous les champs requis
4. Vérification de la cohérence avec les données existantes

4.3 Visualisation des résultats

```
yesmine@Yassou MINGW64 ~/Desktop/Tanit/src (yasmine)
$ python run_extractor.py
================================================================================

📂 Lecture du JSON : extraction_pdf.json
✓ Données extraites pour le patient : 25502
✓ CSV existant trouvé : C:\Users\yesmine\Desktop\Tanit\data\raw\patients.csv

✍️Ajout de la ligne au CSV...
  ✓ Patient ajouté : 25502

✅ DONNÉES AJOUTÉES AVEC SUCCÈS !
📁 Fichier CSV mis à jour : C:\Users\yesmine\Desktop\Tanit\data\raw\patients.csv

📊 DONNÉES EXTRAITES ET AJOUTÉES :
--------------------------------------------------------------------------------
  • Patient_id          : 25502
  • Cycle_number        : 1
  • Age                 : 30
  • Protocol            : flexible antagonist
  • AMH                 : 3.64 ng/mL
  • N_Follicles         : 9
  • E2_day5             : 350 pg/mL
  • AFC                 : ❌ Non trouvé
  • Patient_Response    : optimal
--------------------------------------------------------------------------------
```

FIGURE 3 – Extraction depuis PDF

4.4 Défis de cette phase :

Variabilité des formats PDF : Nécessité d'un prompt robuste

Valeurs composées : Gestion des formats "10/8" nécessitant un parsing spécifique

Abréviations médicales non standardisées

Présence de données manuscrites ou annotations

Tables multi-pages avec en-têtes répétés

Valeurs manquantes ou non détectées

5 Phase 2 : Analyse Exploratoire des Données (EDA)

5.1 Objectifs

Comprendre la distribution des variables

Identifier les patterns médicaux

Détecter les anomalies et outliers

Analyser les corrélations

5.2 Statistiques Descriptives

Distribution de la Variable Cible

— Proportion Low / Optimal / High responders

— Vérification de l'équilibre des classes

Statistiques Univariées

— Mesures de tendance centrale (moyenne, médiane, mode)

— Dispersion (écart-type, IQR, min/max)

— Distribution (histogrammes, box plots)

Statistiques descriptives :

	cycle_number	Age	n_Follicles	AFC
count	501.000000	495.000000	468.000000	262.000000
mean	1.441118	31.735354	16.551282	12.286260
std	0.659565	4.676656	10.541421	6.679164
min	1.000000	24.000000	1.000000	3.000000
25%	1.000000	28.000000	8.000000	7.000000
50%	1.000000	32.000000	15.000000	12.000000
75%	2.000000	35.000000	24.000000	16.000000
max	3.000000	45.000000	46.000000	29.000000

FIGURE 4 – Enter Caption

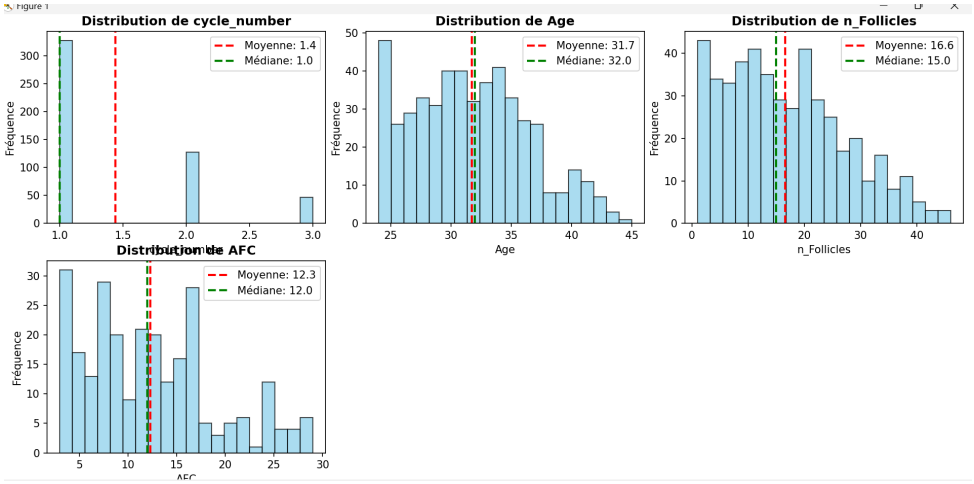

FIGURE 5 – Enter Caption

5.3 Analyse des Corrélations

Matrice de Corrélation

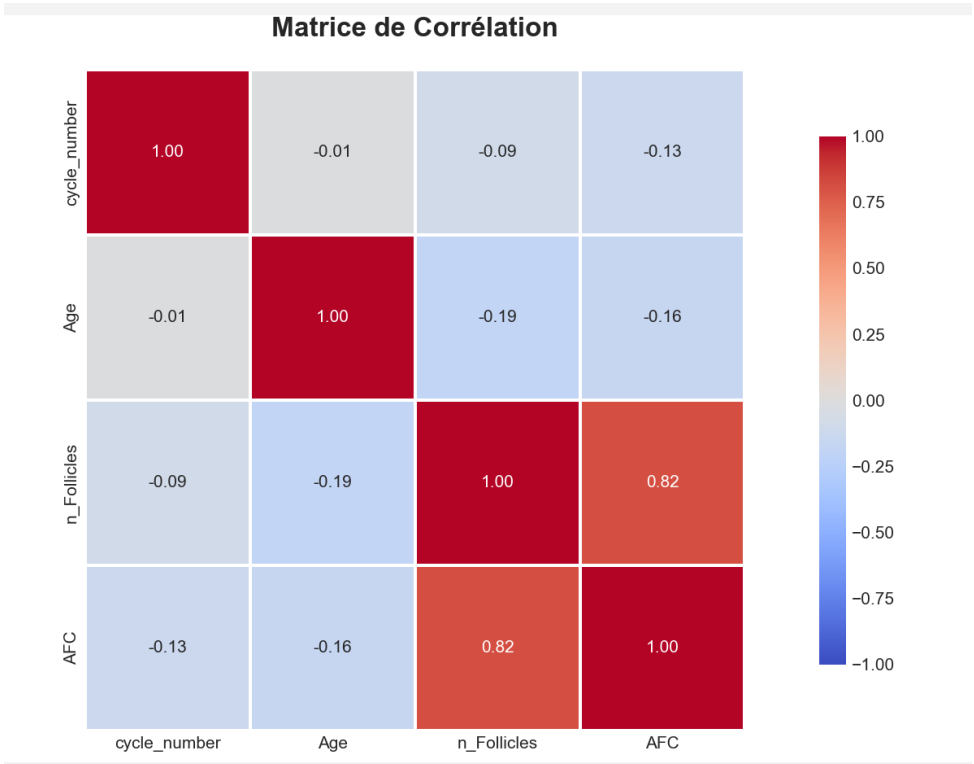

FIGURE 6 – Matrice de corrélation

— AMH ↔ AFC : forte corrélation positive
— Age ↔ AMH : corrélation négative (déclin avec l'âge)
— N_Follicles ↔ E2_day5 : corrélation positive
— AMH/AFC ↔ Patient_Response : corrélations significatives

5.4 Insights Cliniques et Patterns

Effet de l'Âge

— Déclin de la réserve ovarienne après 35 ans

— Interaction âge-AMH : poids prédictif variable selon l'âge

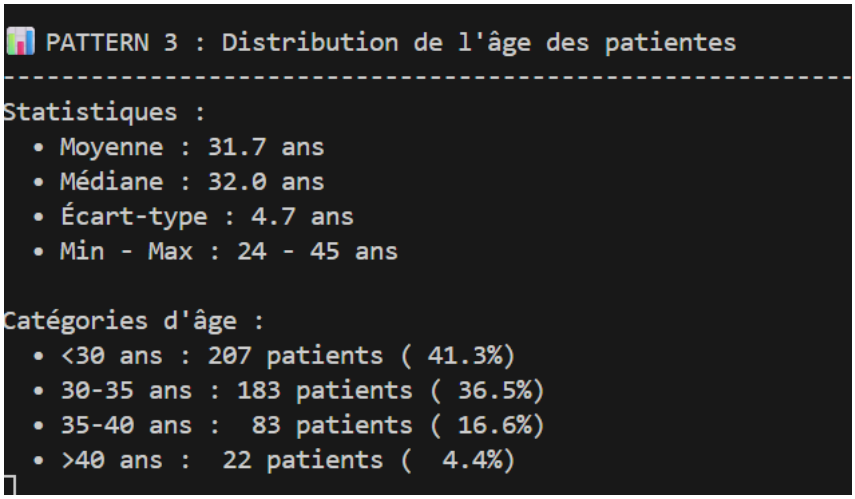

FIGURE 7 – DistributionAge

Impact des Protocoles

— Efficacité différentielle selon le profil de réserve

— Fixed antagonist : adapté aux réponses normales

— Flexible antagonist : meilleur contrôle des high responders

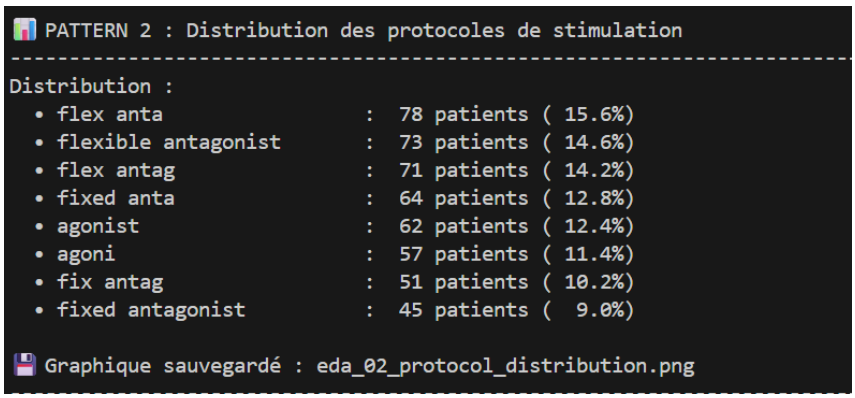

FIGURE 8 – Enter Caption

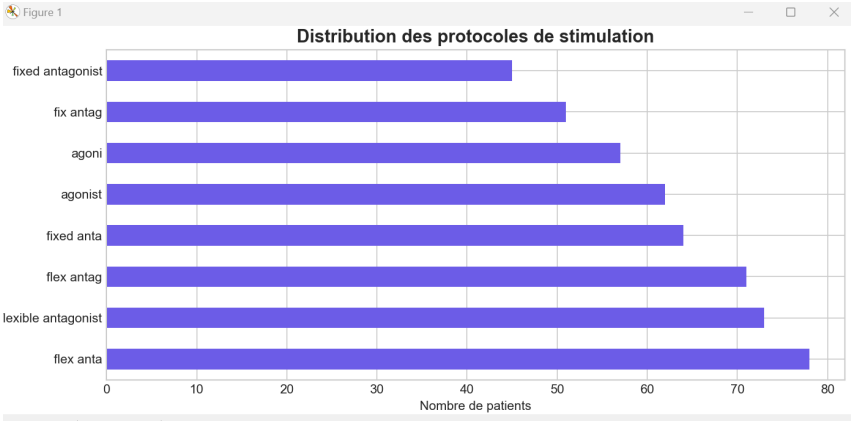

FIGURE 9 – Enter Caption

5.4.1 Visualisation des Distributions

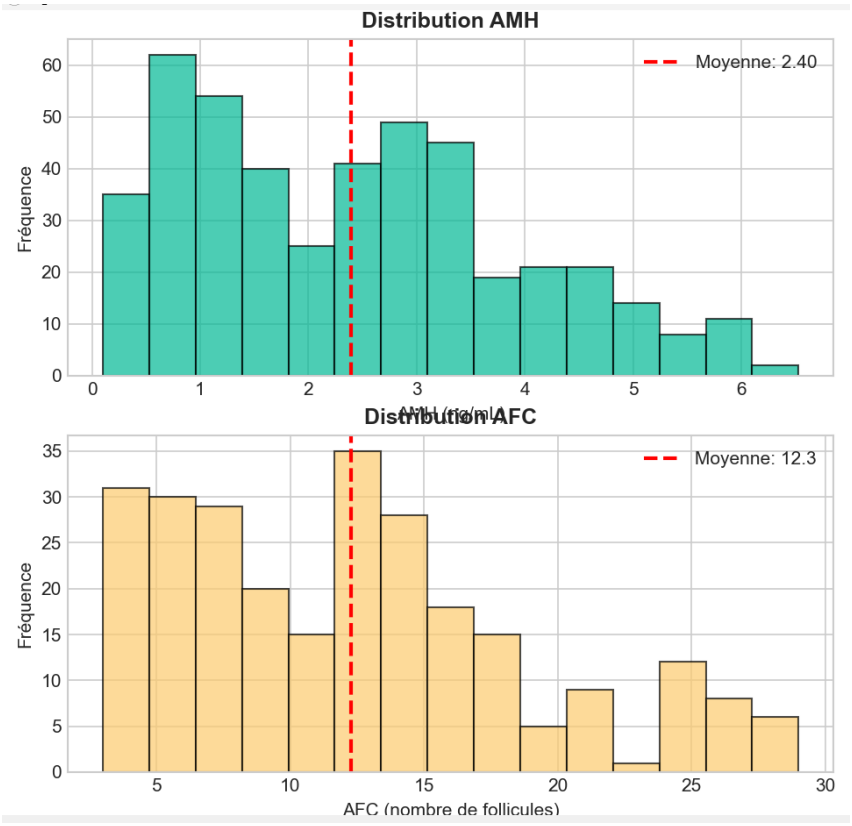

FIGURE 10 – Enter Caption

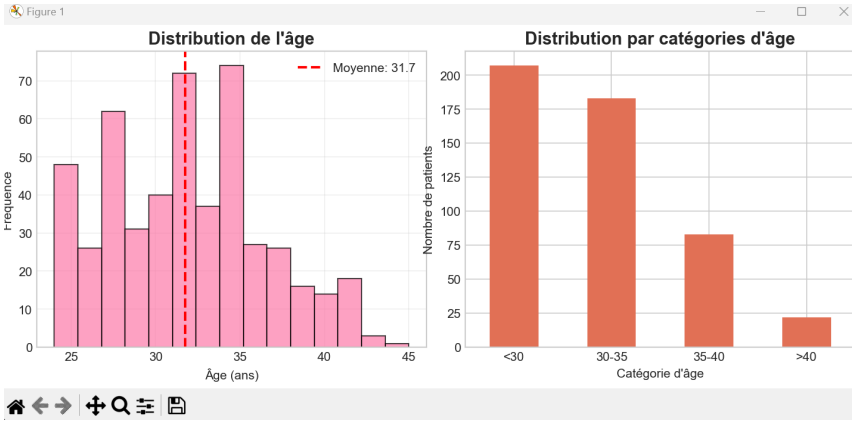

FIGURE 11 – Enter Caption

Résumé de cette phase

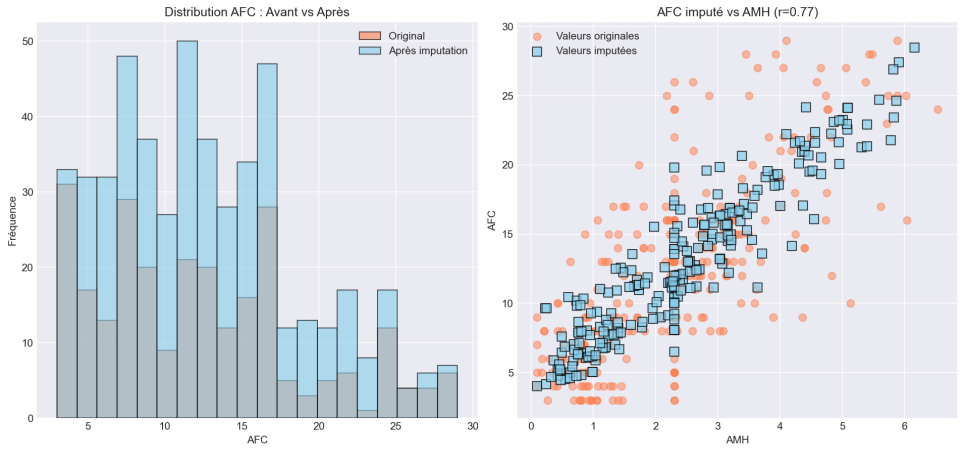

FIGURE 12 – Enter Caption

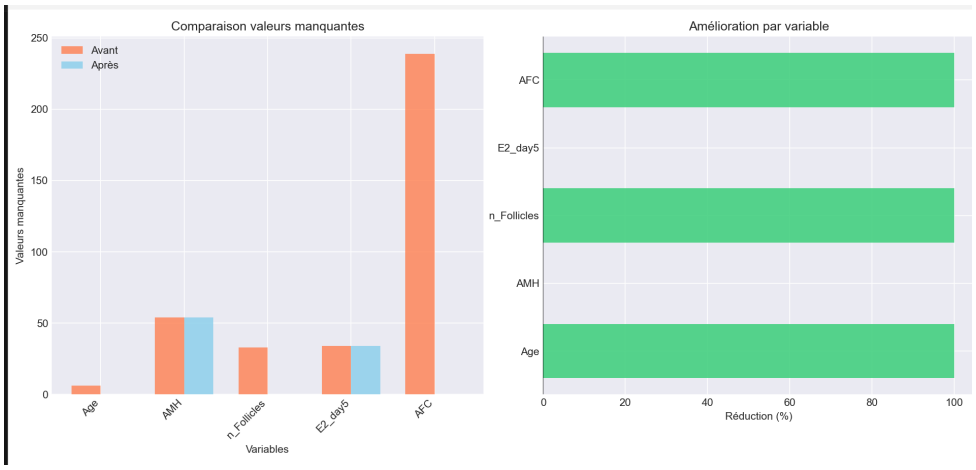

FIGURE 13 – Aucune valeurs manquante

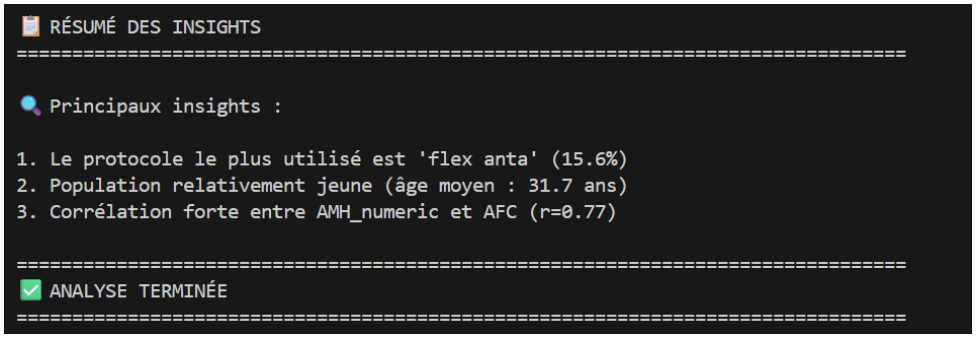

FIGURE 14 – Enter Caption

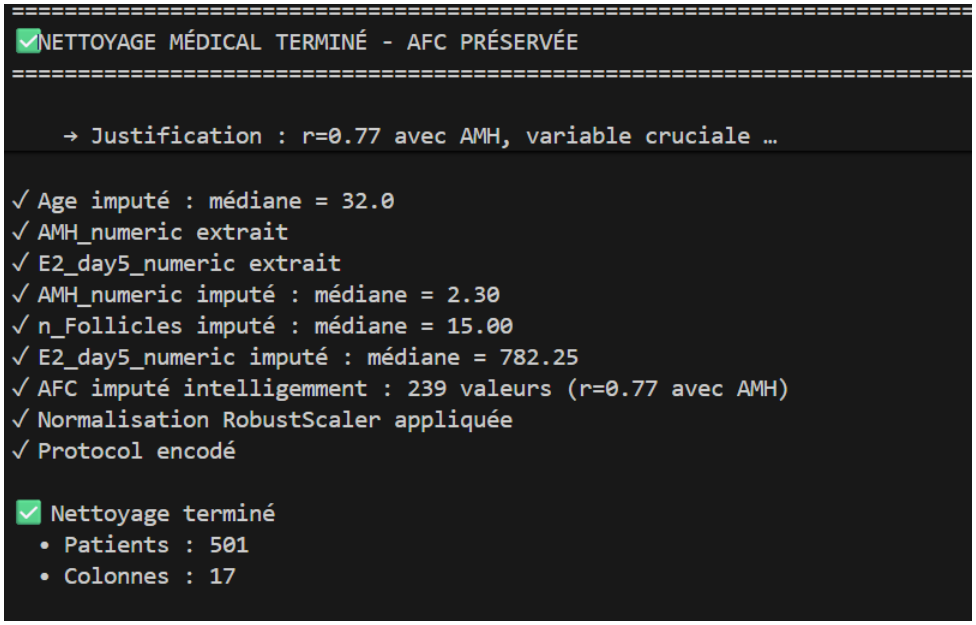

FIGURE 15 – Enter Caption

6 Phase3 : Prétraitement et Nettoyage

6.0.1 Détection et Gestion des Valeurs Manquantes

— Analyse du taux de valeurs manquantes par variable

```
============================================================
2️⃣ ANALYSE DES VALEURS MANQUANTES
============================================================

⚠️ VALEURS MANQUANTES DÉTECTÉES :
------------------------------------------------------------
    Colonne  Valeurs manquantes  Pourcentage (%)
        AFC                239        47.704591
        AMH                 54        10.778443
    E2_day5                 34         6.786427
n_Follicles                 33         6.586826
        Age                  6         1.197605
------------------------------------------------------------

💾 Graphique sauvegardé : eda_01_missing_values_analysis.png
```

— Stratégies selon le type de variable :

— Variables numériques : imputation par médiane (robuste aux outliers)

```
================================================================================
ÉTAPE 2 : IMPUTATION DE AGE (MÉDIANE UNIQUEMENT)
================================================================================

🔧 Imputation de Age :
--------------------------------------------------------------------------------
  • Valeurs manquantes : 6 (1.2%)
  • Médiane de Age : 32.00 ans
  • Méthode : MÉDIANE (règle médicale stricte)
  ✓ 6 valeurs imputées avec la médiane
--------------------------------------------------------------------------------
```

FIGURE 16 – Imputations de l'age avec mediane

— Variables catégorielles : mode ou création d'une catégorie «Unknown»

— Imputation Avancée (IterativeImputer) : Exploite la forte corrélation AFC-AMH (r=0.77) pour une imputation plus précise et médicalement cohérente.

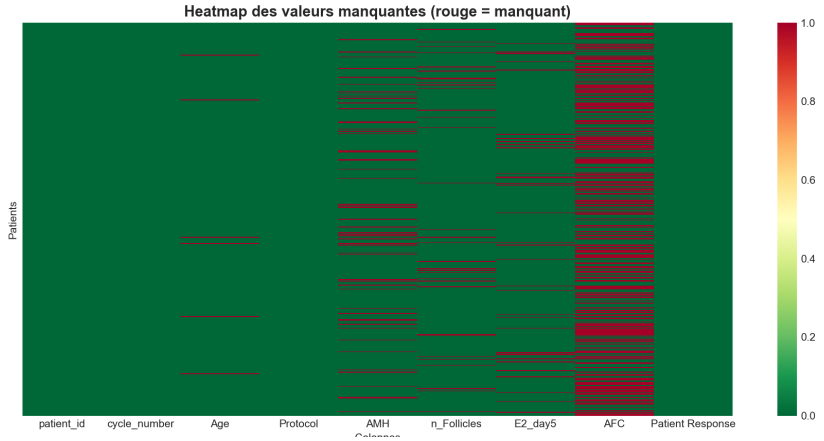

FIGURE 17 – Visualidation des valeurs manquantes

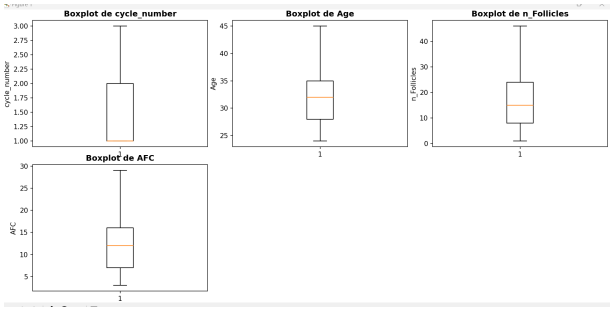

FIGURE 19 – Détection valeurs aberrantes

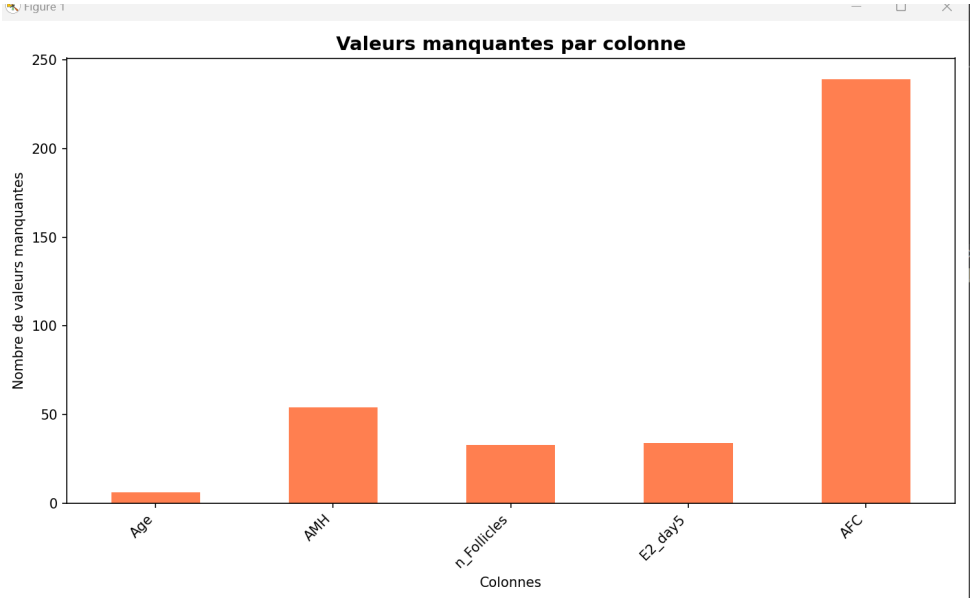

FIGURE 18 – Valeurs manquantes par colonnes

6.0.2 Détection des Valeurs Aberrantes

— Méthode IQR (Interquartile Range) : valeurs hors [Q1-1.5×IQR, Q3+1.5×IQR]

— Z-score : valeurs avec $|z| > 3$

— Validation clinique : vérification que les plages correspondent aux normes biologiques

— Décision : correction, suppression ou conservation selon le contexte clinique

6.0.3 Détection des Doublons

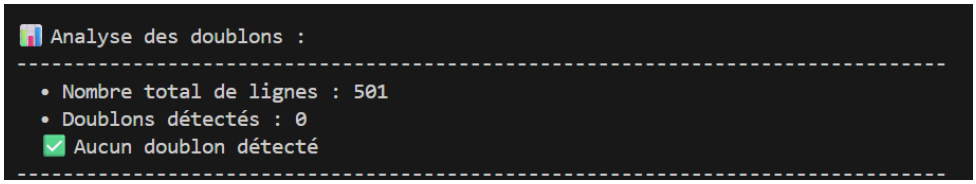

FIGURE 20 – Détection des Doublons

6.0.4 Normalisation des Variables Numériques

— StandardScaler : $(x - \mu)/\sigma \rightarrow$ distribution centrée réduite
— Ou MinMaxScaler : $x \in [0, 1]$ si distribution non gaussienne
— RobustScaler : résistant aux outliers (médiane et IQR)
— Normalisation par variable pour éviter domination par échelle

```
ÉTAPE 4 : NORMALISATION AVEC ROBUSTSCALER
================================================================================

🔬 RÈGLE MÉDICALE STRICTE :
--------------------------------------------------------------------------------
  ❌ MinMaxScaler : INTERDIT (détruit info biologique)
  ❌ StandardScaler : INTERDIT (sensible aux outliers)
  ✅ RobustScaler : SEUL AUTORISÉ (basé sur médiane/IQR)
--------------------------------------------------------------------------------

📊 Variables à normaliser : Age, AMH_numeric, n_Follicles, E2_day5_numeric

🔄 Age :
--------------------------------------------------------------------------------
  Avant  : min=24.00, max=45.00, median=32.00
  Médiane : 32.00
  IQR : 7.00
  Après  : median=0.00, IQR=1.00
--------------------------------------------------------------------------------

🔄 AMH_numeric :
--------------------------------------------------------------------------------
  Avant  : min=0.10, max=6.53, median=2.30
  Médiane : 2.30
  IQR : 2.02
  Après  : median=0.00, IQR=1.00
```

–>Choix du RobustScaler :

— Utilise la médiane et l'IQR (Interquartile Range)
— Robuste aux outliers (important pour les biomarqueurs)
— Préserve les valeurs extrêmes médicalement significatives (high responders)

6.0.5 Encodage des Variables Catégorielles

— Protocol : Encodage ordinal

```
  🔹 Protocoles uniques trouvés : ['flexible antagonist' 'fix antag' 'flex anta' 'fixed
anta' 'agonist'
 'flex antag' 'agoni' 'fixed antagonist']
Agonist : ['agonist']
Flexible Antagonist : ['flexible antagonist', 'flex anta', 'flex antag']
Fixed Antagonist : ['fix antag', 'fixed anta', 'fixed antagonist']
✓ Protocol encodé : valeurs uniques après encodage -> [ 1.  2.  0. -1.]
```

FIGURE 21 – Encodage du Protocol

— Patient_Response (cible) : Label Encoding (0 :low, 1 :optimal, 2 :high)

7 Phase 4 : Entraînement des ML

7.1 Objectif

L'objectif de cette phase est de **prédire la réponse des patientes au traitement IVF** selon trois catégories cliniques : *low*, *optimal* et *high*. Pour cela, plusieurs algorithmes de classification supervisée ont été entraînés, optimisés et comparés. L'objectif final est de :

— identifier le modèle le plus performant selon le **F1-score pondéré**, adapté aux classes déséquilibrées ;

— sauvegarder le meilleur modèle et ses hyperparamètres optimaux ;

— fournir une base fiable pour les phases d'interprétabilité, d'évaluation clinique et de déploiement.

7.2 Algorithmes Utilisés

Cinq modèles supervisés ont été testés afin de couvrir plusieurs familles d'algorithmes (linéaires, arbres, ensembles, marges maximales) :

Régression Logistique Multinomiale

— Modèle linéaire simple et fortement interprétable.

— Adapté à la classification multi-classes.

— Fournit directement des probabilités.

Random Forest

— Ensemble d'arbres de décision construit par bagging.

— Robuste aux données bruitées et aux outliers.

— Réduit la variance et améliore la stabilité des prédictions.

Gradient Boosting

— Approche de boosting séquentiel corrigeant les erreurs des arbres précédents.

— Excellent comportement sur données tabulaires.

— Souvent le modèle le plus performant dans des contextes cliniques structurés.

AdaBoost

— Boosting adaptatif basé sur des "faibles apprenants".

— Donne plus de poids aux erreurs pour affiner la frontière de décision.

— Particulièrement efficace lorsque les données sont bien séparables.

Support Vector Classifier (SVC)

— Apprentissage basé sur des hyperplans maximisant la marge entre classes.
— Kernel RBF utilisé pour capturer la non-linéarité.
— Probabilités calibrées via la méthode de Platt.

7.3 Étapes Principales de l'Entraînement

1. Chargement et Prétraitement du Dataset

— Les données nettoyées proviennent du module `IVFDataset`.
— Elles incluent les valeurs biologiques, hormonales, antécédents médicaux et variables cliniques.
— Les features et labels sont extraits sous forme de (X, y).

2. Division des Données

— Split : 80% train, 10% validation, 10% test.
— Split stratifié pour conserver la distribution Low/Optimal/High.
— Fixation du random seed pour reproductibilité.

3. Préparation des Données

— Option de normalisation via StandardScaler.
— Pipeline automatique pour garantir l'absence de fuite d'information (data leakage).

4. Initialisation des Modèles et Grilles d'Hyperparamètres

Pour chaque modèle :
— définition d'une grille d'hyperparamètres,
— intégration dans un pipeline Scikit-learn,
— préparation pour une optimisation systématique via GridSearchCV.

5. Entraînement avec GridSearchCV

— Validation croisée stratifiée (StratifiedKFold).
— Score principal : **F1-score pondéré**.
— Recherche exhaustive des hyperparamètres optimaux.

6. Sélection et Sauvegarde du Meilleur Modèle par Algorithme

Pour chaque algorithme :
— sauvegarde du meilleur modèle (`.pkl`),
— sauvegarde des hyperparamètres optimaux (`.json`),
— archivage dans un dossier avec horodatage.

7. Historisation

— Création d'un historique complet : nom du modèle, hyperparamètres, score final, date.
— Stockage dans la variable `training_history`.

7.4 Résultat Attendu

À l'issue de cette phase :

— tous les modèles candidats sont entraînés et sauvegardés ;
— les hyperparamètres optimaux sont documentés ;
— une comparaison objective pourra être effectuée lors de la phase d'évaluation ;
— un modèle "gagnant" sera sélectionné pour déploiement selon son F1-score pondéré.

8 Phase 5 : Évaluation et Choix du Modèle

8.1 Stratégie d'Évaluation

L'objectif de cette phase est d'évaluer les différents modèles entraînés afin d'identifier celui offrant la meilleure capacité de prédiction de la réponse des patientes au traitement IVF (classes : *low*, *optimal*, *high*). L'évaluation repose principalement sur le **F1-score pondéré**, adapté aux jeux de données déséquilibrés.

Jeu de Test

L'ensemble de test est généré par le module `IVFDataset` selon la configuration définie lors de l'entraînement :

— Split : 80% entraînement, 10% validation, 10% test

— Stratification pour préserver la distribution des classes

— Données normalisées si demandé

— Variables utilisées : `X_test` (features), `y_test` (labels), `class_names` (noms des classes)

Le jeu de test n'est **jamais utilisé pendant l'entraînement**, garantissant une évaluation non biaisée.

8.2 Modèles Évalués

Les cinq modèles suivants, préalablement entraînés et sauvegardés au format `.pkl`, sont évalués :

Modèle	Description
Logistic Regression	Modèle linéaire multinomial, baseline interprétable.
Random Forest	Ensemble d'arbres de décision robuste, réduit la variance.
Gradient Boosting	Boosting séquentiel corrigeant les erreurs précédentes.
AdaBoost	Boosting adaptatif basé sur des estimateurs faibles.
SVC	Classifieur à hyperplans maximisant les marges, kernels linéaire/RBF.

Chaque modèle est évalué sans réentraînement, en utilisant les poids optimisés obtenus via `GridSearchCV`.

8.3 Métriques d'Évaluation

Pour chaque modèle, les métriques suivantes sont calculées :

— **Accuracy :** proportion de bonnes prédictions.

— **Precision pondérée :** moyenne pondérée selon l'importance des classes.

— **Recall pondéré :** taux de vrais positifs pondéré.
— **F1-score pondéré : métrique principale** pour le choix du meilleur modèle.

$$F1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

— **Matrice de confusion :** analyse des erreurs par classe.
— **Classification Report :** inclut précision, rappel, F1 et support.

```
==================================================
Evaluating gradient_boosting
==================================================
INFO:__main__:Accuracy:  0.8515
INFO:__main__:Precision: 0.8563
INFO:__main__:Recall:    0.8515
INFO:__main__:F1-score:  0.8524

Classification Report:
              precision    recall  f1-score   support

         low       0.79      0.87      0.83        31
     optimal       0.84      0.84      0.84        45
        high       0.95      0.84      0.89        25

    accuracy                           0.85       101
   macro avg       0.86      0.85      0.86       101
weighted avg       0.86      0.85      0.85       101

INFO:__main__:Sample 0: {'low': '0.4%', 'optimal': '99.1%', 'high': '0.5%'}
INFO:__main__:Sample 1: {'low': '2.6%', 'optimal': '80.4%', 'high': '17.0%'}
INFO:__main__:Sample 2: {'low': '97.2%', 'optimal': '0.9%', 'high': '1.9%'}
INFO:__main__:Sample 3: {'low': '4.4%', 'optimal': '6.2%', 'high': '89.4%'}
INFO:__main__:Sample 4: {'low': '4.5%', 'optimal': '90.0%', 'high': '5.4%'}
INFO:__main__:
```

FIGURE 22 – Evaluate GredientBoost

```
Evaluating random_forest
==================================================
INFO:__main__:Accuracy:  0.8713
INFO:__main__:Precision: 0.8734
INFO:__main__:Recall:    0.8713
INFO:__main__:F1-score:  0.8715

Classification Report:
              precision    recall  f1-score   support

         low       0.82      0.90      0.86        31
     optimal       0.88      0.84      0.86        45
        high       0.92      0.88      0.90        25

    accuracy                           0.87       101
   macro avg       0.87      0.88      0.87       101
weighted avg       0.87      0.87      0.87       101
```

FIGURE 23 – RandomForest

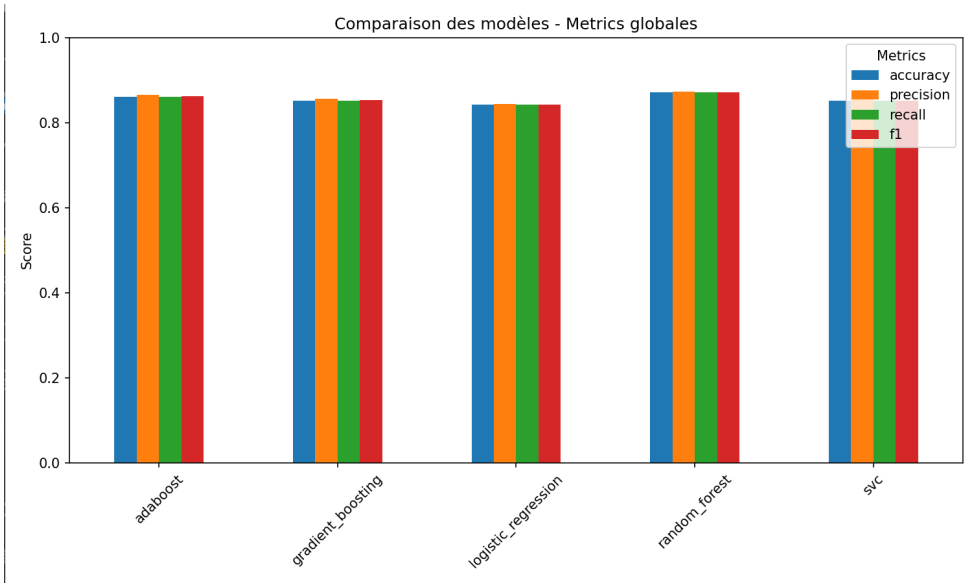

FIGURE 25 – Comparaison entre les modéls

8.4 Analyse et Visualisations

— **Bar Charts Comparatifs :** comparaison directe des modèles sur Accuracy, Precision weighted, Recall weighted et F1-score weighted.

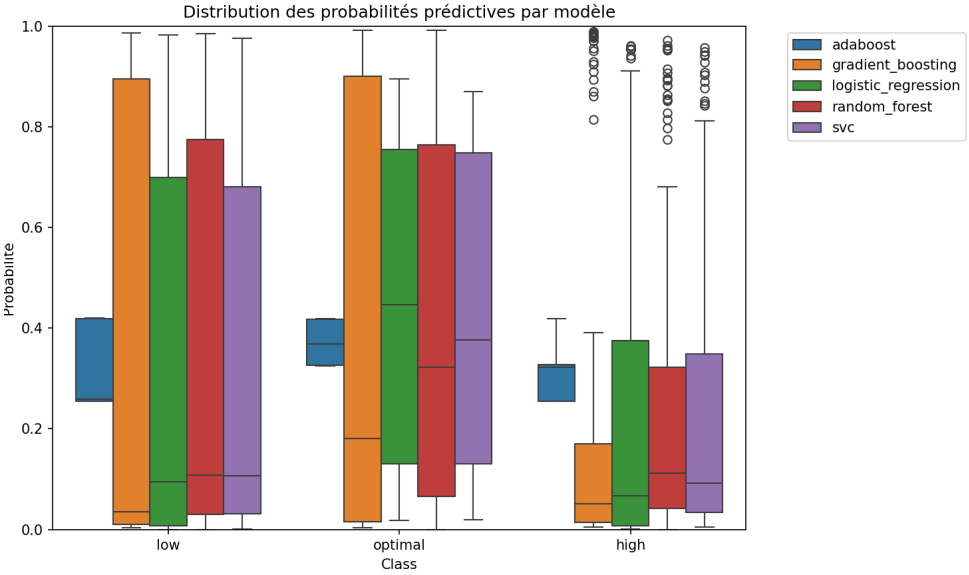

FIGURE 24 – Comparaison entre les models

— **Heatmaps de Matrices de Confusion :** permet d'observer les classes bien prédites et les confusions.

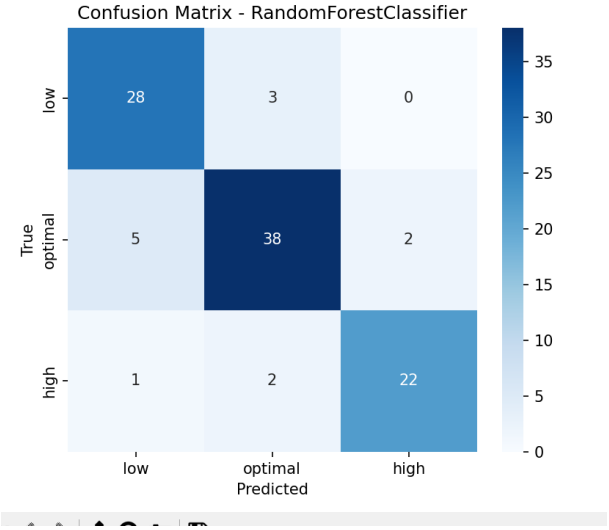

FIGURE 26 – Matrice de confusion de RandomForest

— **Analyse des Probabilités :** boxplots et quelques prédictions avec probabilités pour vérifier la calibration et la confiance des modèles.

8.5 Sélection du Meilleur Modèle

Critère de Sélection

Le meilleur modèle est défini comme celui ayant le **F1-score pondéré le plus élevé sur le jeu de test**, justifié par :

— la présence éventuelle de déséquilibre des classes,

— le compromis entre précision et rappel,

— la robustesse du F1-score pour les problèmes multiclasses.

```
✅ Meilleur modèle sélectionné : random_forest avec F1-score pondéré = 0.8715
INFO:__main__:Description du meilleur modèle (random_forest): RandomForestClassifier(max_depth=20, min_samples_split=5, n_jobs=-1,
                  random_state=42)
INFO:__main__:Le meilleur modèle a été sauvegardé dans : saved_models\best_model.pkl
```

FIGURE 27 – Choix du meilleurModel

9 Phase 6 : Prédiction

9.1 Objectif

Cette phase consiste à prédire la réponse des patientes au traitement IVF en utilisant le modèle sélectionné lors de la phase d'évaluation. La réponse est stratifiée en trois classes : *low*, *optimal* et *high*. L'objectif est de fournir pour chaque patiente :

— La classe prédite (*predicted response*)

— La confiance de la prédiction (*confidence*)

— La distribution des probabilités pour chaque classe

— Une interprétation clinique basée sur les valeurs de probabilité

— Des recommandations cliniques personnalisées

9.2 Chargement du Modèle

Le modèle utilisé pour la prédiction est le meilleur modèle sauvegardé (`best_model.pkl`) accompagné des préprocesseurs :

— **Scalage des variables** : `scaler.pkl`

— **Encodage des classes** : `label_encoder.pkl`

— **Informations sur les features** : `feature_info.pkl`

— **Métadonnées du modèle** : `best_model_metadata.json` (optionnel)

9.3 Prétraitement des Données

Avant la prédiction, les données patient sont prétraitées :

— Conversion en `DataFrame` si nécessaire

— Vérification et création des features attendues par le modèle

— Application du scaler sur les variables numériques (*Age, AMH, N_Follicles, E2_day5, AFC*)

— Encodage des variables catégorielles, notamment *Protocol*

— Remplissage automatique des features manquantes avec des valeurs par défaut

9.4 Prédiction

Le modèle prédit pour chaque patiente :

— La classe prédite : `predicted_response`

— L'indice de classe : `predicted_class_index`

— Les probabilités par classe : `probabilities`

— La confiance maximale : `confidence`

Des fonctions spécifiques permettent :

— **predict_single_patient** : prédiction détaillée pour un patient avec interprétation clinique et recommandations

— **predict_batch** : prédiction pour plusieurs patientes et sauvegarde des résultats dans un fichier CSV

9.5 Interprétation Clinique et Recommandations

Pour chaque patiente, le modèle fournit une interprétation et des recommandations personnalisées :

— **Interprétation clinique** : résumé de la probabilité et de la confiance, avec classification en haute, moyenne ou faible confiance.

— **Recommandations** :

  — *Low responders* : augmenter la dose initiale, suivi plus fréquent, évaluer la réserve ovarienne, considérer protocoles alternatifs.

  — *High responders* : réduire la dose pour limiter les risques d'OHSS, planifier un déclencheur agoniste, suivi étroit, envisager stratégie de *freeze-all*.

  — *Optimal responders* : protocole standard, suivi régulier, bonne probabilité de succès.

9.6 Exemple de Prédiction

— Patient : 29 ans, cycle numéro 1, protocole *flex anta*, AMH 5.62, 15 follicules, E2_day5 699.82, AFC 17

— Résultat :

  — Classe prédite : *high*

  — Confiance : 92%

  — Probabilités : low 5%, optimal 3%, high 92%

  — Interprétation : *Haute probabilité d'une réponse élevée au traitement, suivi recommandé pour OHSS.*

  — Recommandations : dose initiale plus faible, déclencheur agoniste, surveillance étroite des niveaux d'estradiol.

9.7 Résumé

La phase de prédiction permet de transformer un modèle machine learning optimisé en un outil décisionnel clinique, offrant à la fois :

— Des prédictions quantitatives (classe et probabilités)

— Des interprétations cliniques compréhensibles

— Des recommandations personnalisées pour la prise en charge de chaque patiente

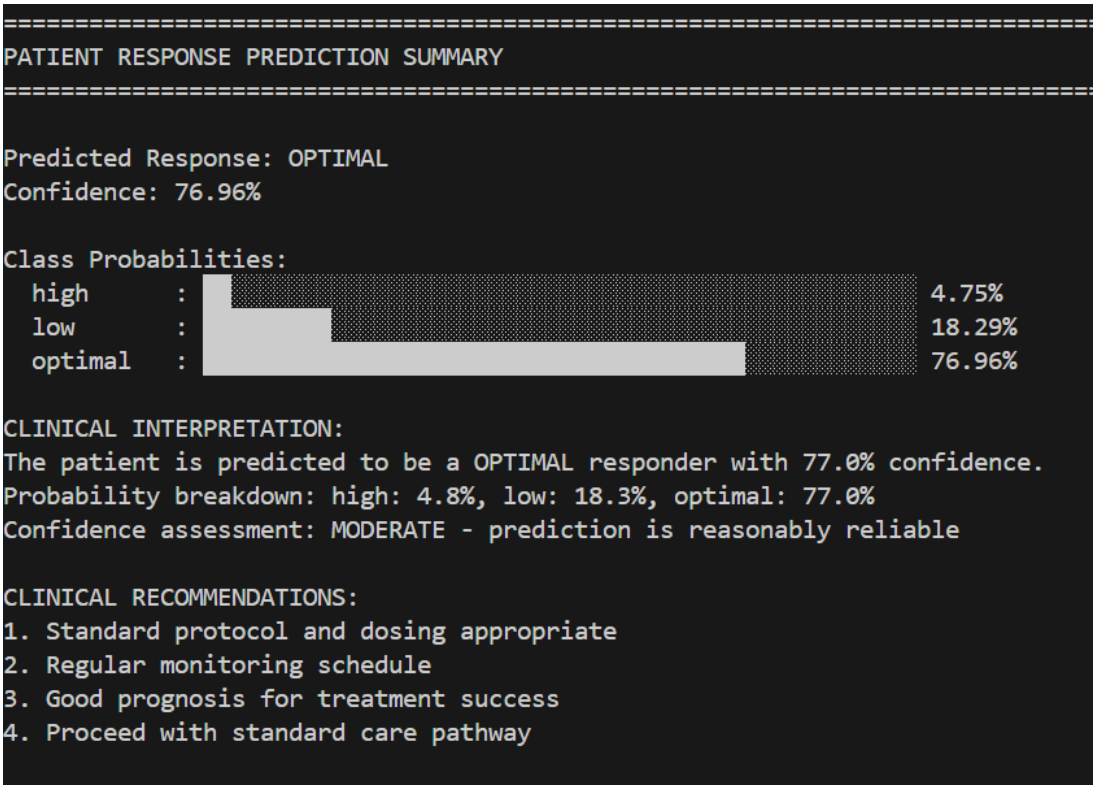

FIGURE 28 – Enter Caption

10 Phase 7 : Interface Utilisateur et Intégration du Modèle

10.1 Description

Cette partie présente l'interface développée pour l'interaction avec le modèle de prédiction. Elle combine un **backend Flask** pour la gestion des API et des requêtes et un **frontend Streamlit** pour l'affichage interactif des résultats.

— **Backend Flask** :
  — Sert de serveur API pour recevoir les données patient
  — Gère la communication avec le modèle de prédiction
  — Retourne les résultats au frontend en format JSON

— **Frontend Streamlit** :
  — Interface utilisateur simple et interactive
  — Formulaires pour saisir les caractéristiques des patientes
  — Visualisation des résultats : classe prédite, probabilités, interprétation clinique et recommandations

— **Intégration du modèle** :
  — Le modèle de prédiction est chargé via le backend Flask
  — Streamlit envoie les données utilisateur à Flask
  — Les résultats retournés sont affichés en temps réel sur le frontend

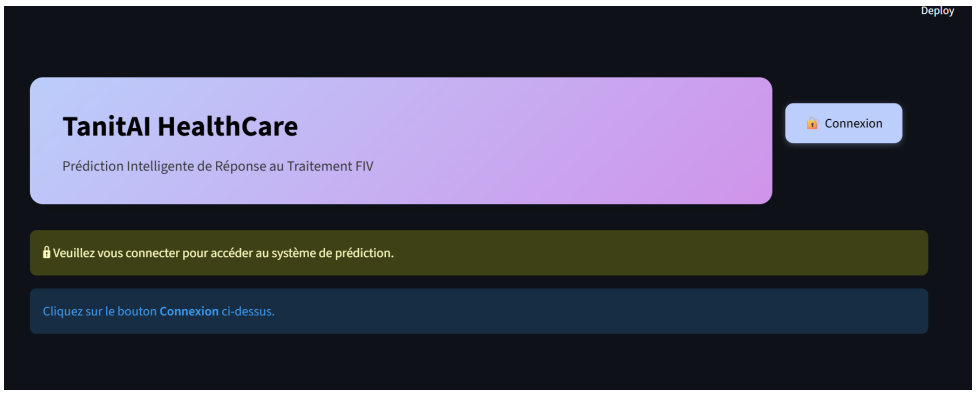

FIGURE 29 – Home Page

FIGURE 31 – Formulaire du client

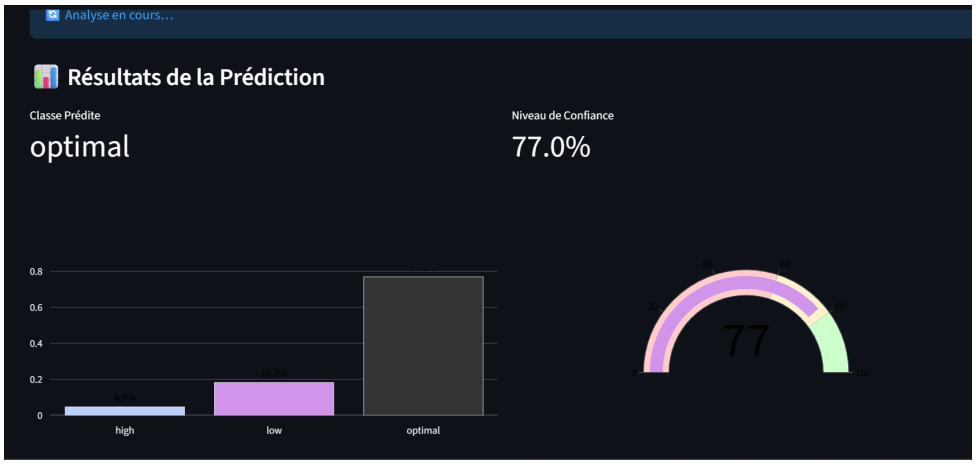

FIGURE 32 – Resultat de prediction

FIGURE 30 – Page de prediction

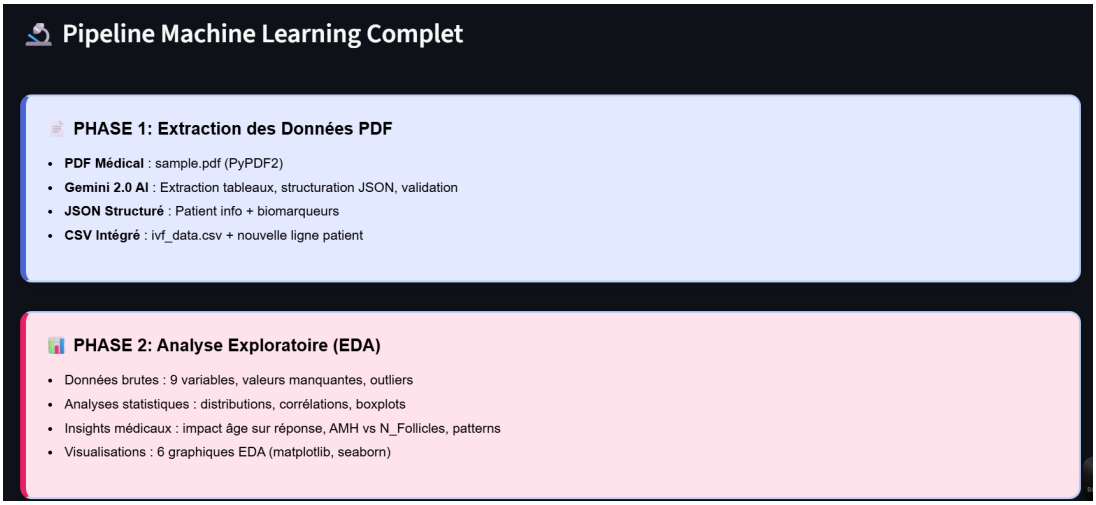

FIGURE 33 – Pipeline

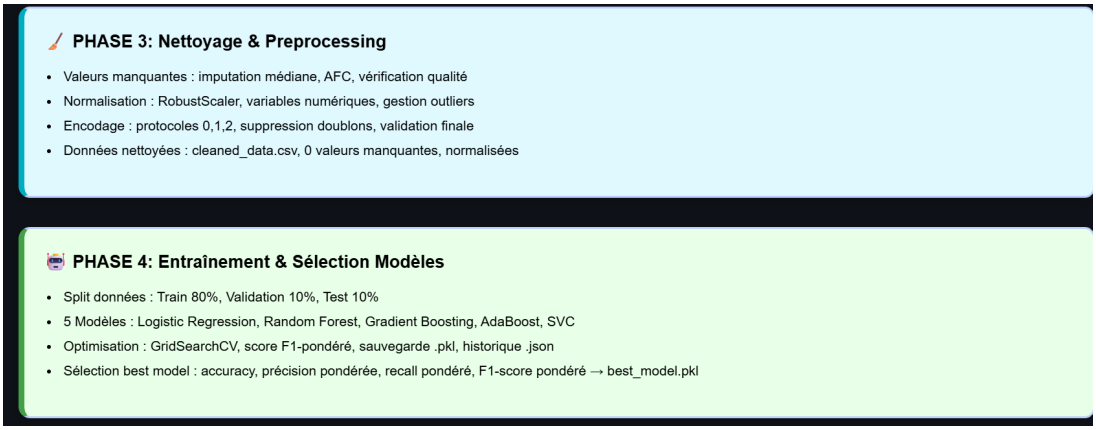

FIGURE 34 – Enter Caption

FIGURE 35 – Information sur Modele

11 Perspectives Importantes

11.1 Recommandation avec LLM

Dans les prochaines évolutions du projet, la phase de **recommandation finale** pourrait être réalisée à l'aide d'un **Large Language Model (LLM)**. L'idée serait que le LLM prenne en entrée :

— L'historique médical des patientes

— Les prédictions générées par le modèle de machine learning

— Les données médicales contextuelles et cliniques

et fournisse des recommandations personnalisées plus fiables.

Cette approche présente un gain potentiel en **précision et contextualisation clinique**, mais elle nécessite des ressources computationnelles importantes et du temps d'exécution, ce qui n'a pas permis de finaliser cette phase dans le temps imparti.

11.2 Visualisation du pipeline

Pour améliorer la compréhension et la traçabilité du processus, il serait intéressant d'intégrer une **visualisation complète du pipeline**, incluant :

— Le téléchargement et la préparation des données (PDF, CSV, etc.)

— L'extraction et le prétraitement des données

— L'analyse via les modèles de machine learning

— La génération des prédictions et des recommandations

— La production des conclusions et rapports

11.3 Documentation et mots-clés

Une documentation complète serait également essentielle pour :

— Décrire chaque étape du pipeline

— Fournir des mots-clés et définitions dans le domaine médical et informatique

— Faciliter la maintenance et l'évolution du projet

11.4 Version et contraintes

Ce travail peut être considéré comme une **version 0** du projet, réalisée en **moins de 48 heures** en raison des contraintes de devoirs et de validation. Les perspectives présentées ici serviront de base pour les améliorations futures et l'intégration des fonctionnalités avancées.