

Analyse Complète

Projet de Diagnostic d'Appendicite Pédiatrique

Équipe Médicale & Data Science

12 mars 2025

Résumé

Information

Ce document présente une analyse détaillée de la structure et des fonctionnalités de chaque composant d'un projet de diagnostic d'appendicite pédiatrique. L'application utilise l'apprentissage automatique pour aider les médecins à évaluer le risque d'appendicite chez les patients pédiatriques en utilisant diverses caractéristiques cliniques et résultats de tests.

Table des matières

1	Fichiers racine	2
2	Dossiers principaux	2
2.1	DATA (Données)	2
2.2	src (Code source)	2
2.3	models (Modèles entraînés)	3
2.4	tests (Tests)	3
2.5	Autres dossiers	3
3	Fonctionnement général de l'application	3
4	Caractéristiques principales utilisées par le modèle	4

1 Fichiers racine

- **Dockerfile** : Définit l'environnement Docker pour l'application, spécifiant l'image de base Python et l'installation des dépendances.
- **docker-compose.yml** : Configure le déploiement de l'application, définit les variables d'environnement et les ports exposés.
- **.dockerignore** : Liste les fichiers et dossiers à exclure lors de la création de l'image Docker (comme les fichiers temporaires et les environnements virtuels).
- **requirements.txt** : Liste toutes les bibliothèques Python nécessaires (Flask, pandas, scikit-learn, etc.).
- **setup.py** : Script permettant d'installer le projet comme un package Python, définissant les métadonnées et les dépendances.
- **README.md** : Documentation principale expliquant l'objectif du projet, son installation et son utilisation.
- **fix_templates.py** : Script utilitaire créé pour corriger les problèmes de variable dans les templates.

2 Dossiers principaux

2.1 DATA (Données)

- **DATA/raw** : Contient les données brutes de patients, non traitées.
- **DATA/processed** : Stocke les données après nettoyage et prétraitement, prêtes pour l'entraînement.
- **DATA/external** : Données provenant de sources externes (potentiellement d'autres ensembles de données médicales).

2.2 src (Code source)

- **src/app.py** : Point d'entrée principal qui initialise et lance l'application Flask.
- **src/api** : Module contenant l'application web Flask :
 - *app.py* : Configure l'application Flask, définit les routes et gère les requêtes.
 - *templates/* : Contient les fichiers HTML pour l'interface utilisateur (formulaire de diagnostic, page de résultats).
 - *static/* : Fichiers statiques (CSS, JavaScript, images) pour l'interface utilisateur.
- **src/models** : Gère les modèles de Machine Learning :
 - Contient le code pour entraîner, évaluer et sauvegarder les modèles de prédiction.
 - Implémente le RandomForestClassifier utilisé pour les prédictions d'appendicite.
- **src/data_processing** : Traitement des données :
 - Scripts pour nettoyer, transformer et préparer les données pour l'entraînement.
 - Gestion des valeurs manquantes et normalisation des caractéristiques.
- **src/explainability** : Module pour l'interprétabilité du modèle :
 - Outils pour comprendre les prédictions du modèle.
 - Visualisations montrant l'importance des caractéristiques.
- **src/features** : Extraction et sélection de caractéristiques :
 - Définit les caractéristiques importantes pour le diagnostic (symptômes, tests sanguins, scores cliniques).

- Transforme les données brutes en caractéristiques utilisables par le modèle.
- **src/utls** : Fonctions utilitaires :
 - Outils généraux réutilisés dans différentes parties du projet.
 - Fonctions de journalisation, gestion de configuration, etc.
- **src/visualization** : Outils de visualisation :
 - Création de graphiques et visualisations des données et résultats.
 - Aide à l'interprétation des résultats pour les médecins.

2.3 models (Modèles entraînés)

- **models/best_model.pkl** : Modèle RandomForest entraîné et sérialisé, prêt à être utilisé pour les prédictions.
- **models/configs** : Fichiers de configuration des hyperparamètres du modèle.
- **models/model_history** : Historique des versions précédentes du modèle pour comparaison.

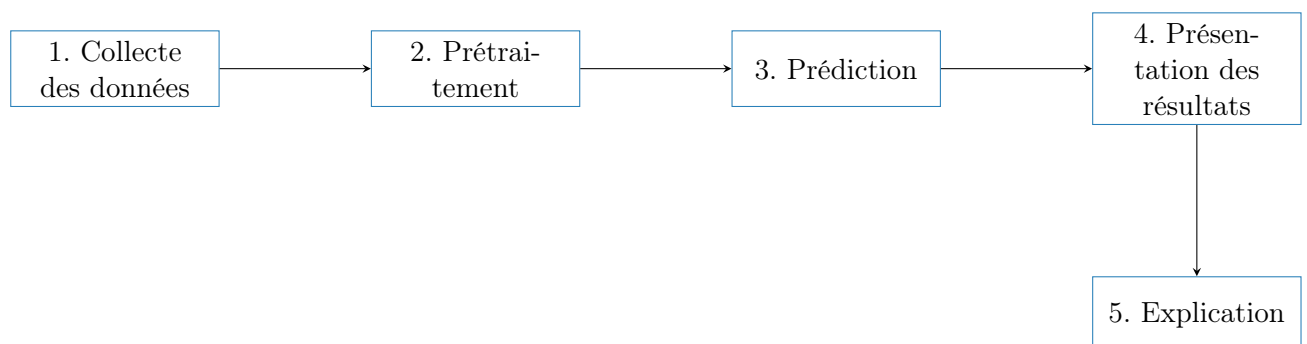
2.4 tests (Tests)

- **tests/run_tests.py** : Script central qui découvre et exécute tous les tests du projet.
- **tests/unit** : Tests unitaires qui vérifient les fonctionnalités individuelles :
 - *test_model.py* : Vérifie le chargement du modèle, ses caractéristiques et ses capacités de prédiction.
- **tests/integration** : Tests d'intégration qui vérifient l'interaction entre les composants :
 - *test_api_model_integration.py* : Teste l'intégration entre l'API Flask et le modèle de ML.
- **tests/api** : Tests spécifiques à l'API web.
- **tests/data_processing** : Tests pour les fonctions de traitement de données.
- **tests/explainability** : Tests des outils d'explication du modèle.

2.5 Autres dossiers

- **docs** : Documentation du projet :
 - *api* : Documentation technique de l'API.
 - *user_guide* : Guide d'utilisation pour les médecins.
- **notebooks** : Notebooks Jupyter pour l'exploration de données et l'expérimentation.
- **logs** : Fichiers journaux de l'application.
- **config** : Fichiers de configuration du projet.
- **scripts** : Scripts utilitaires pour diverses tâches.

3 Fonctionnement général de l'application



- 1. Collecte des données** Le médecin saisit les informations du patient via l'interface web (données démographiques, symptômes, résultats de tests).
- 2. Prétraitement** Les données sont transformées au format attendu par le modèle.
- 3. Prédiction** Le modèle RandomForest analyse les caractéristiques et calcule la probabilité d'appendicite.
- 4. Présentation des résultats** L'application affiche le niveau de risque (faible, moyen, élevé) et la probabilité estimée.
- 5. Explication** Des visualisations aident à comprendre quelles caractéristiques ont influencé la prédiction.

4 Caractéristiques principales utilisées par le modèle

Catégorie	Caractéristiques
Démographiques	Âge, sexe
Symptômes cliniques	Durée de la douleur, migration de la douleur, anorexie, nausées, vomissements, douleur au quadrant inférieur droit, fièvre, rebond de sensibilité
Tests sanguins	Nombre de globules blancs, pourcentage de neutrophiles, protéine C-réactive
Scores cliniques	Score d'appendicite pédiatrique (PAS), score d'Alvarado

TABLE 1 – Caractéristiques utilisées dans le modèle de prédiction d'appendicite

Information

Remarque importante :

Le modèle a été entraîné sur une population pédiatrique spécifique et doit être utilisé comme un outil d'aide à la décision uniquement. Les décisions cliniques finales doivent toujours être prises par un médecin qualifié en tenant compte de la situation clinique complète du patient.