

# Rapport de Projet — Plateforme Django de Diagnostic de Tension (ML)

## 1. Contexte et Objectifs

**Problématique** : Aider au screening rapide de l'hypertension via des valeurs de tension artérielle (TA).

**Objectif** : Développer une mini-plateforme Web (Django) permettant de saisir *systolic* et *diastolic* et d'obtenir un diagnostic indicatif, d'abord par règles, puis via un modèle ML entraîné.

## 2. Données

**CSV interne** : diagnostic/data/bp\_dataset.csv

**Colonnes** : systolic, diastolic, label

**Classes** : Normal, Élevée, Hypertension Stade 1, Hypertension Stade 2, Crise hypertensive.

**Remarque** : Dataset d'exemple (synthétique) conforme aux catégories usuelles de TA.

## 3. Méthode et Architecture

**Stack** : Django 5, scikit-learn (Logistic Regression), joblib.

**Approche** : Règles de base (fallback) dans diagnostic/ml.py; modèle ML entraîné et chargé si présent.

### *Fichiers clés*

- diagnostic/ml.py → predict\_bp(), chargement modèle et fallback
- diagnostic/train\_model.py → pipeline d'entraînement et sauvegarde
- diagnostic/forms.py → BloodPressureForm
- diagnostic/views.py → home()
- diagnostic/templates/diagnostic/home.html → interface utilisateur
- diagnostic/data/bp\_dataset.csv → dataset d'exemple
- projet\_m1Sir/settings.py → INSTALLED\_APPS inclut diagnostic
- projet\_m1Sir/urls.py → include('diagnostic.urls') à la racine

## 4. Implémentation ML

**Entrées** : [[systolic, diastolic]].

**Labels** : ['Normal', 'Élevée', 'Hypertension Stade 1', 'Hypertension Stade 2', 'Crise hypertensive'].

**Pipeline** : StandardScaler + LogisticRegression(multi\_class='multinomial').

**Persistance** : diagnostic/model/bp\_model.joblib via joblib.

## 5. Résultats et Évaluation

Après entraînement, un rapport de classification est affiché (précision, rappel, f1). Sur données synthétiques, performance élevée attendue.

## 6. Démonstration

- 1 Installer les dépendances : `pip install -r requirements.txt`
- 2 Entraîner le modèle : `python diagnostic/train_model.py`
- 3 Lancer le serveur : `python manage.py migrate && python manage.py runserver`
- 4 Tester : `http://127.0.0.1:8000/` puis saisir systolic/diastolic

## 7. Avertissements et Éthique

Outil indicatif, non destiné au diagnostic médical. Pas de stockage patient implémenté. Les performances sur données réelles requièrent un dataset cliniquement représentatif.

## 8. Limites et Améliorations

- Variables limitées (uniquement TA). Ajouter d'autres facteurs cliniques.
- Dataset de démo synthétique. Intégrer des données réelles anonymisées.
- Améliorer la robustesse (détection anomalies, unités, valeurs manquantes).
- Élargir les modèles (RF, XGBoost), calibration des probabilités, tuning.