■ ANALYSE DU PROJET

Système de Classification IA pour Données CAN

Date d'analyse: 08/07/2025

Type de projet: Intelligence Artificielle - Machine Learning

■ OBJECTIF PRINCIPAL

Ce projet est un **système de classification basé sur l'intelligence artificielle** pour analyser et prédire des classes à partir de données CAN (Controller Area Network) - typiquement utilisées dans les systèmes automobiles ou industriels.

■ COMPOSANTS DU PROJET

1. Analyse et Comparaison de Modèles IA

- Notebook Jupyter (IA_Model_Comparison.ipynb) pour comparer différents algorithmes
- Utilisation de Random Forest et XGBoost pour la classification
- Optimisation des hyperparamètres avec des outils comme Optuna

2. API Web de Prédiction

- Application Flask (app.py) qui expose une API REST
- Interface web (client.html) pour tester les prédictions en temps réel
- Endpoint /predict qui accepte 8 valeurs numériques (D0 à D7)

3. Pipeline de Données

- Prétraitement des données hexadécimales CAN (conversion en entiers)
- Normalisation avec StandardScaler
- Gestion des valeurs manquantes

4. Modèles Entraînés

- Modèle XGBoost optimisé (xgb_optimized.pkl)
- Modèle Random Forest (random_forest_model.pkl)
- Scaler pré-entraîné pour la normalisation

■ FONCTIONNALITÉS

- **Prédiction en temps réel:** Interface web pour saisir des données et obtenir des prédictions
- **API REST:** Service web pour intégrer les prédictions dans d'autres applications
- **Script de prédiction:** Outil en ligne de commande pour traiter des fichiers CSV
- **Tests:** Validation des fonctionnalités de l'API

■ CAS D'USAGE PROBABLE

Ce projet semble être conçu pour:

- **Détection d'anomalies** dans les réseaux CAN
- **Classification de messages** automobiles ou industriels
- **Surveillance en temps réel** de systèmes embarqués
- **Analyse prédictive** de données IoT ou véhiculaires

■■ ARCHITECTURE TECHNIQUE

Fichier	Description	Technologie
app.py	API Flask principale	Python/Flask
client.html	Interface utilisateur web	HTML/CSS/JavaScript
predict.py	Script de prédiction CLI	Python
IA_Model_Comparison.ipynb	Notebook d'analyse	Jupyter/Python
models/xgb_optimized.pkl	Modèle XGBoost entraîné	Pickle/Scikit-learn
models/scaler.pkl	Normalisateur de données	Pickle/Scikit-learn
requirements.txt	Dépendances Python	pip

■■ TECHNOLOGIES UTILISÉES

- **Backend:** Flask, Python
- **Machine Learning:** Scikit-learn, XGBoost, Pandas, NumPy
- **Visualisation:** Matplotlib, Seaborn
- **Optimisation:** Optuna, GridSearchCV
- **Frontend:** HTML, CSS, JavaScript
- **Validation:** Pydantic

■ CONCLUSION

Ce projet combine **recherche** (notebook), **développement** (API) et **déploiement** (interface web) dans un workflow complet de machine learning. Il représente une solution complète pour la classification automatisée de données CAN avec une approche moderne et des outils de pointe en intelligence artificielle.