Dossier Technique - Projet Machine Learning Vin 🍷

# Table des matières

1. Présentation du projet

2. Architecture et organisation du code

3. Description des modules

4. Scripts utilitaires

5. Fonctionnalités implémentées

6. Mode d’emploi pour l’utilisateur

7. Ressources et références

# 1. Présentation du projet

Ce projet vise à développer une application interactive en Python avec Streamlit qui présente un pipeline complet de Machine Learning appliqué à un dataset de vins (vin.csv). L’application permet :  
- L’exploration des données  
- Le prétraitement avancé (gestion des valeurs manquantes par différentes stratégies, suppression de colonnes)  
- La modélisation avec différents algorithmes ML (Logistic Regression, Random Forest, SVM) et Deep Learning (Keras)  
- L’évaluation détaillée des modèles via métriques, visualisations, et comparaison rapide avec LazyPredict  
- L’optimisation automatique des hyperparamètres avec GridSearchCV  
- La prédiction interactive sur de nouvelles données, avec gestion automatique des valeurs manquantes

# 2. Architecture et organisation du code

├── data/  
│ └── vin.csv # Jeu de données CSV (à fournir manuellement)  
  
├── models/ # Modèles ML sauvegardés (joblib / Keras)  
│ ├── logistic\_regression.joblib  
│ ├── random\_forest.joblib  
│ ├── label\_encoder.joblib  
│ ├── scaler.pkl  
│ ├── encoder.pkl  
│ ├── model\_dl.h5 # Modèle Deep Learning Keras  
│ └── feature\_names.txt  
  
├── pages/  
│ ├── exploration.py # Exploration interactive des données  
│ ├── pretraitement.py # Prétraitement avancé (imputation multiple, suppression)  
│ ├── training.py # Entraînement ML & DL, sauvegarde des artefacts  
│ ├── evaluation.py # Évaluation, prédiction avec gestion des valeurs manquantes  
│ ├── deep\_learning.py # Module dédié au Deep Learning (Keras)  
│ └── app.py # Point d’entrée principal (navigation Streamlit)  
  
├── setup.py # Script d’installation des dépendances  
├── run.py # Script de lancement de l’application Streamlit  
├── requirements.txt # Liste des packages requis  
├── README.md # Documentation utilisateur / notes projet  
└── .gitignore # Fichiers/dossiers ignorés par Git

# 3. Description des modules

## pages/exploration.py

* - Affichage du DataFrame et statistiques descriptives
* - Visualisations interactives (histogrammes, pairplot, matrice de corrélation)
* - Sélection dynamique des colonnes numériques pour l’analyse

## pages/pretraitement.py

* - Nettoyage des données
* - Affichage et gestion des valeurs manquantes
* - Imputation configurable (moyenne, médiane, mode, ou stratégie avancée à venir)
* - Suppression des colonnes choisies par l’utilisateur

## pages/training.py

* - Chargement et prétraitement des données
* - Sélection automatique et/ou manuelle des features
* - Entraînement des modèles ML classiques (Logistic Regression, Random Forest, SVM)
* - Entraînement de modèles Deep Learning avec Keras (multi-couches, dropout, early stopping)
* - Optimisation des hyperparamètres via GridSearchCV
* - Comparaison rapide avec LazyPredict (tous les modèles ML en un clic)
* - Sauvegarde des modèles, encodeurs, scalers et noms de features dans `models/`

## pages/evaluation.py

* - Chargement des artefacts ML & DL
* - Interface utilisateur pour saisie manuelle ou import CSV pour prédiction
* - Imputation automatique des valeurs manquantes (médiane par défaut, mode ou moyenne possible)
* - Prédiction avec affichage des probabilités et décodage des labels
* - Gestion robuste des erreurs et validation des entrées utilisateur

## pages/deep\_learning.py

* - Module dédié à la construction, entraînement, et évaluation du modèle Keras
* - Sauvegarde/chargement des artefacts Deep Learning (modèle `.h5`, scaler, encodeur)
* - Intégration complète dans le workflow Streamlit avec gestion de session

## pages/app.py

* - Point d’entrée Streamlit
* - Navigation fluide via sidebar entre Exploration, Prétraitement, Modélisation, Évaluation, Deep Learning

# 4. Scripts utilitaires

## setup.py

Script d’installation automatique des dépendances Python via pip  
Installe et prépare l’environnement virtuel  
  
Usage :  
python setup.py

## run.py

Script principal pour démarrer l’application Streamlit  
Lance le module pages/app.py  
Usage :  
python run.py  
ou  
streamlit run run.py

# 5. Fonctionnalités implémentées

* - Chargement dynamique du dataset
* - Exploration interactive avec graphiques et statistiques
* - Prétraitement avancé : gestion fine des valeurs manquantes avec plusieurs stratégies (moyenne, médiane, mode)
* - Suppression de colonnes selon choix utilisateur
* - Modélisation ML complète avec split, fit, prédiction
* - Comparaison automatique de modèles ML avec LazyPredict
* - Optimisation hyperparamètres avec GridSearchCV
* - Entraînement Deep Learning avec Keras (réseau fully connected, dropout, early stopping)
* - Sauvegarde et chargement des artefacts (modèles, scalers, encodeurs)
* - Prédiction interactive sur de nouvelles données, avec imputation automatique des valeurs manquantes
* - Interface utilisateur fluide avec navigation par sidebar

# 6. Mode d’emploi pour l’utilisateur

1. Cloner le dépôt Git
2. Installer les dépendances :  
    python setup.py
3. Lancer l’application :  
    python run.py  
    ou  
    streamlit run run.py
4. Utiliser la sidebar pour naviguer entre les pages :  
    - Exploration  
    - Prétraitement  
    - Modélisation (ML classique + Deep Learning)  
    - Évaluation et Prédiction
5. Explorer les options, entraîner les modèles, effectuer des prédictions interactives

# 7. Ressources et références

* - Documentation officielle Streamlit : https://docs.streamlit.io
* - Documentation scikit-learn : https://scikit-learn.org
* - Keras Documentation : https://keras.io
* - Cours LinkedIn Learning : Python pour la Data Science
* - LazyPredict GitHub : https://github.com/shankarpandala/lazypredict