**Mini-Projet de Machine Learning**

**Baazaoui Mohamed Ali 1DNI1**

**🌾 Contexte du Dataset**

Ce dataset est destiné à **recommander la meilleure culture** (ex : riz, maïs, pois chiches, etc.) à un agriculteur en fonction des **caractéristiques du sol et des conditions climatiques**.

On va utiliser un modèle de machine learning qui, à partir de certaines données d'entrée (mesures du sol et du climat), **prévoit quelle culture est la plus adaptée** pour une parcelle donnée.

**📊 Description des colonnes (features)**

| **Colonne** | **Description** |
| --- | --- |
| N | Teneur en **azote** dans le sol (en mg/kg) |
| P | Teneur en **phosphore** dans le sol (en mg/kg) |
| K | Teneur en **potassium** dans le sol (en mg/kg) |
| temperature | **Température moyenne** dans la région (en degrés Celsius) |
| humidity | **Humidité relative** de l'air (en %) |
| ph | **pH du sol**, mesure de l’acidité ou alcalinité du sol (entre 0 et 14) |
| rainfall | **Précipitations annuelles** moyennes (en mm) |

Ces 7 colonnes sont les **features**, c’est-à-dire les données d’entrée qu’on utilise pour entraîner un modèle.

**🎯 Colonne cible (target)**

| **Colonne** | **Description** |
| --- | --- |
| label | La **culture recommandée** (ex: rice, maize, coconut...). C’est la **variable à prédire**. |

Il y a **22 types de cultures** différents dans la colonne label.

**🧠 Exemple concret**

| **N** | **P** | **K** | **temperature** | **humidity** | **ph** | **rainfall** | **label** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 90 | 42 | 43 | 20.87 | 82.0 | 6.5 | 202.9 | **rice** |

➡️ Cela signifie :  
Avec ces conditions (N, P, K, température, etc.), **la meilleure culture à planter serait du riz**.

Pour le **Crop Recommendation Dataset**, comme la **cible (label) est une variable catégorique** (classification multi-classes), nous allons utiliser un **modèle de classification supervisée**.

Voici les **modèles les plus adaptés** pour ce genre de problème 👇 :

**✅ Modèles de classification possibles**

| **Modèle** | **Avantages** | **Inconvénients** |
| --- | --- | --- |
| **Random Forest** 🌳 | Très précis, gère bien les features numériques, résistant au surapprentissage | Plus lent si le dataset est énorme |
| **K-Nearest Neighbors (KNN)** 👥 | Simple et efficace pour les petits datasets | Lent si beaucoup de données |
| **Logistic Regression** 📈 | Rapide, facile à interpréter | Moins performant pour des données non linéaires |
| **SVM (Support Vector Machine)** | Très efficace pour des données séparables | Moins efficace sur de grands ensembles |
| **XGBoost / LightGBM** 🚀 | Performant, rapide, utilisé en compétition Kaggle | Un peu plus complexe à configurer |

**🎯 Meilleur choix ici : Random Forest Classifier**

Pourquoi ?

* Gère bien les features numériques comme N, P, K, température, etc.
* Gère les problèmes de sur-apprentissage grâce à l’agrégation d’arbres.
* Très bon pour la classification multi-classes.
* Peu de préparation nécessaire (pas besoin de scaling).