**Informations essentielles sur la compétition :**

1. **Objectif de la compétition** :
   * La compétition vise à explorer les facteurs prédictifs de santé liés au **syndrome des ovaires polykystiques (SOPK)**, une maladie endocrinienne courante affectant la santé reproductive et la qualité de vie des femmes.
   * L'objectif est d'analyser l'impact des choix de mode de vie (alimentation, exercice, gestion du stress, etc.) sur le SOPK en utilisant des méthodes basées sur les données.
2. **Dates** :
   * **Début** : 25 janvier 2025
   * **Fin** : 13 février 2025
3. **Évaluation** :
   * Les soumissions sont évaluées sur la base de l'**aire sous la courbe ROC (AUC-ROC)** entre la probabilité prédite et la cible observée.
4. **Prix** :
   * **Total des prix disponibles** : 1$ (symbolique, probablement pour encourager la participation).
   * **1ère place** : Une surprise si la performance (AUC-ROC) dépasse 0.9.
5. **Jeu de données** :
   * Le jeu de données est généré à partir d'un modèle de deep learning entraîné sur des données réelles de SOPK.
   * Les distributions des caractéristiques sont proches, mais pas identiques, à celles du jeu de données original.
   * Le but est de faciliter la recherche sur l'impact du mode de vie sur le SOPK, en permettant :
     + L'analyse exploratoire des données (EDA).
     + La modélisation prédictive.
     + La recherche en santé pour comprendre comment l'alimentation, l'exercice et le stress influencent la prévalence du SOPK.

**Choix d'implémentation en Python :**

1. **Préparation des données** :
   * **Gestion des valeurs manquantes** : Utilisation de IterativeImputer pour imputer les valeurs manquantes dans les colonnes numériques.
   * **Gestion des outliers** : Deux méthodes sont proposées : la méthode **IQR** (Interquartile Range) et la méthode **Z-score** pour détecter et traiter les valeurs aberrantes.
   * **Encodage des variables catégorielles** : Utilisation de LabelEncoder pour encoder les variables catégorielles, avec une gestion des catégories non vues dans le jeu de test.
   * **Regroupement des âges et des types d'exercice** : Les variables catégorielles comme l'âge et le type d'exercice sont regroupées en catégories plus larges pour simplifier l'analyse.
2. **Feature Engineering** :
   * **Création de nouvelles caractéristiques** : Plusieurs interactions entre les variables sont créées, comme l'interaction entre le sommeil et l'exercice, ou entre l'âge et l'exercice.
   * **Calcul du BMI** : Le BMI est calculé en supposant une taille moyenne de 1,6 m pour les femmes.
   * **Création de caractéristiques polynomiales et logarithmiques** : Des caractéristiques comme le carré du poids, la fréquence d'exercice au carré, ou le logarithme du BMI sont ajoutées pour capturer des relations non linéaires.
3. **Gestion du déséquilibre des classes** :
   * **SMOTE (Synthetic Minority Over-sampling Technique)** : Utilisé pour équilibrer les classes de la cible (SOPK vs non-SOPK) en générant des échantillons synthétiques pour la classe minoritaire.
4. **Modélisation** :
   * **Plusieurs modèles testés** : Les modèles incluent la régression logistique, les forêts aléatoires, XGBoost, LightGBM, CatBoost, SVM, KNN, Gradient Boosting, AdaBoost, et un réseau de neurones MLP.
   * **Validation croisée** : Une validation croisée en 5 folds est utilisée pour évaluer les performances des modèles.
   * **Sélection du meilleur modèle** : Le modèle avec le meilleur AUC-ROC moyen est sélectionné pour l'optimisation.
5. **Optimisation des hyperparamètres** :
   * **Optuna** : Utilisé pour optimiser les hyperparamètres du meilleur modèle sélectionné. L'optimisation est basée sur la maximisation de l'AUC-ROC.
   * **Hyperparamètres optimisés** : Les hyperparamètres varient selon le modèle (par exemple, n\_estimators, max\_depth, learning\_rate pour XGBoost).
6. **Évaluation finale** :
   * **Courbe ROC et matrice de confusion** : Les performances du modèle sont évaluées à l'aide de la courbe ROC et de la matrice de confusion, avec des pourcentages affichés pour une meilleure interprétation.
   * **Prédictions sur le jeu de test** : Les prédictions finales sont générées pour le jeu de test et sauvegardées dans un fichier de soumission.

**Points forts de l'implémentation :**

* **Approche robuste de préparation des données** : La gestion des valeurs manquantes, des outliers et l'encodage des variables catégorielles sont bien structurés.
* **Feature Engineering approfondi** : La création de nombreuses interactions et caractéristiques dérivées permet de capturer des relations complexes dans les données.
* **Validation croisée et optimisation** : L'utilisation de la validation croisée et d'Optuna pour l'optimisation des hyperparamètres assure une sélection rigoureuse du meilleur modèle.
* **Gestion du déséquilibre des classes** : L'utilisation de SMOTE permet de mieux gérer les classes déséquilibrées, ce qui est crucial pour un problème de classification comme le SOPK.