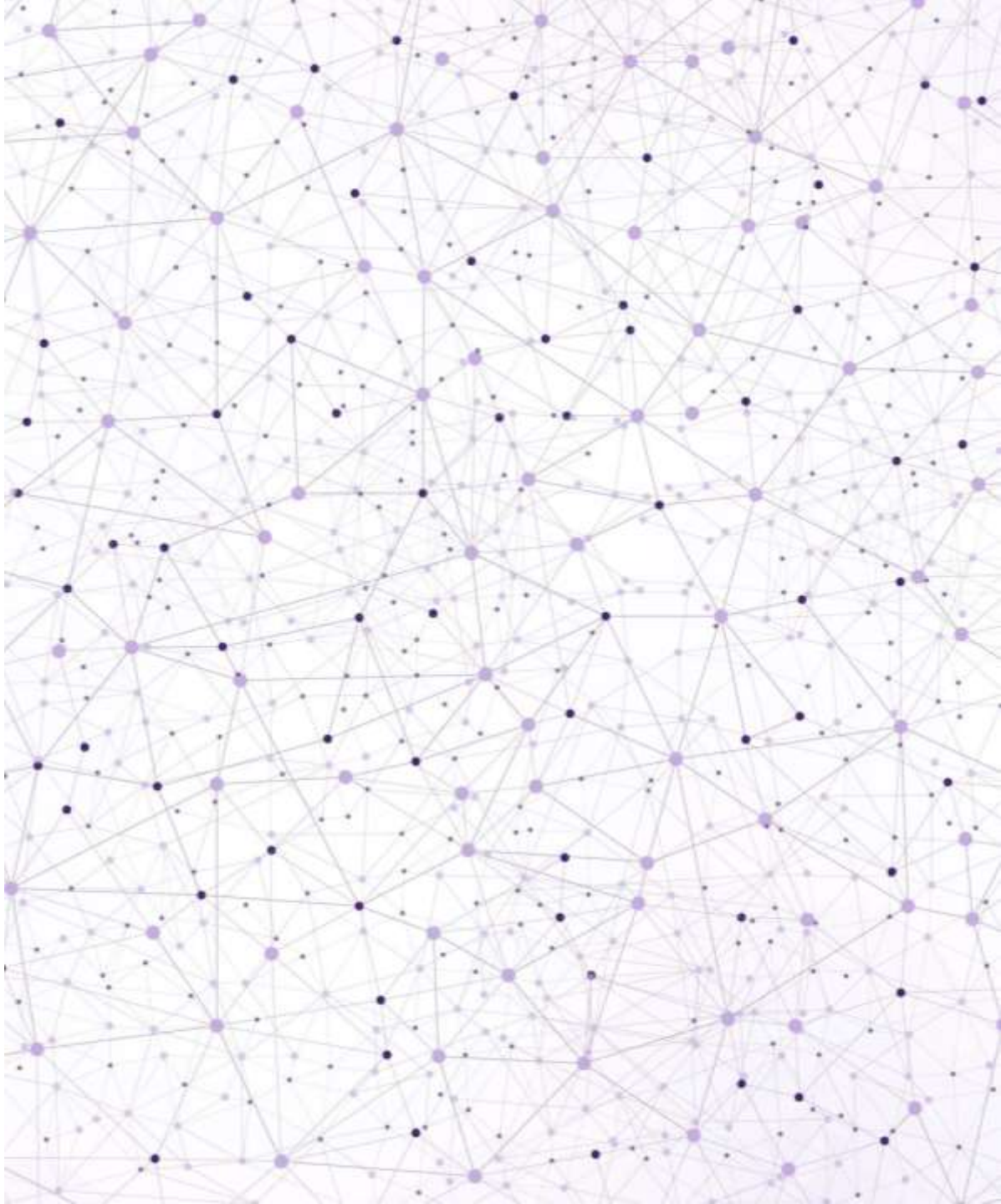
PRÉDICTION DU NIVEAU D'OBÉSITÉ

Application de modèles d’apprentissage machine (UCI Dataset)

Lyes Babou – A00235273

INFO4301 Apprentissage Machine – AUT2025



Introduction

Problématique:

Prédire le niveau d'obésité à partir d'habitudes alimentaires et de caractéristiques physiques.

Objectifs du projet:

- Explorer et comprendre le dataset
- Tester plusieurs modèles de classification
- Optimiser les meilleurs modèles
- Évaluer la performance finale
- Sélectionner le modèle le plus performant

Contexte:

L'obésité est un enjeu de santé majeur. Les données permettent de modéliser les facteurs associés.

Jeu de données

Source:

UCI Machine Learning Repository — Estimation of Obesity Levels Based on Eating Habits and Physical Condition

Taille: 2111 observations, 17 variables

Types de variables :

- Numériques : Age, Height, Weight, etc.
- Catégorielles : Gender, FAVC, SMOKE, MTRANS, etc.
- Cible : **NObeyesdad** (7 classes).

But : prédire une classe parmi les 7 catégories d'obésité.

Exploration des données (EDA)

Observations principales :

- Aucune valeur manquante dans le dataset
- Dataset propre et équilibré au niveau de la cible
- Âges majoritairement entre 18 et 30 ans
- Répartition des classes assez uniforme
- Distributions numériques cohérentes
- Aucun outlier majeur

Visuels recommandés sur cette slide :

- Histogramme Age
- Countplot Nobeyesdad

  (Prends ceux de ton notebook)

Préparation des données

Étapes réalisées :

- Séparation X / y
- Encodage des variables catégorielles (OneHotEncoder)
- Normalisation des variables numériques (StandardScaler)
- Encodage de la cible (LabelEncoder)
- Division des données :
  - 60 % entraînement
  - 20 % validation
  - 20 % test
- Utilisation d’un ColumnTransformer et Pipeline pour automatiser le prétraitement

Algorithmes testés

Modèles évalués :

- Régression logistique
- k-NN
- Arbre de décision
- Random Forest
- Gradient Boosting
- SVM
- MLPClassifier (réseau de neurones)

Pourquoi ces modèles ?

- Variété : modèles linéaires, basés distance, arbres, ensembles, réseaux
- Permet une comparaison complète
- Couvrent tout le contenu du cours

Optimisation des hyperparamètres

Méthode : GridSearchCV (cv=3)

Modèles optimisés :

- Gradient Boosting
- Random Forest
- MLPClassifier

Paramètres testés :

- GB : n_estimators, learning_rate, max_depth
- RF : n_estimators, max_depth, min_samples_split
- MLP : hidden_layer_sizes, activation, learning_rate_init, alpha

Résultat : les trois modèles ont amélioré leurs performances.

Résultats et performances

Validation (avant test) :

- MLP optimisé : ~96.7%
- Gradient Boosting optimisé : ~95.0%
- Random Forest optimisé : ~93.1%

Test final :

- MLP optimisé : Accuracy ≈ 97.16 %, F1 ≈ 97.07 %
- Gradient Boosting : Accuracy ≈ 93.6 %
- Random Forest : Accuracy ≈ 92.9 %

Modèle final sélectionné :

✓ MLPClassifier optimisé

	Model	Val_Accuracy	Val_F1
2	MLP_opt	0.966825	0.965399
0	GB_opt	0.950237	0.949260
1	RF_opt	0.931280	0.930220

	Model	Test_Accuracy	Test_F1
2	MLP_opt	0.971631	0.970779
0	GB_opt	0.936170	0.935665
1	RF_opt	0.929078	0.928379

Analyse critique

Points forts :

1. Modèles performants, surtout MLP
2. Très bonne généralisation (pas de surapprentissage)
3. Dataset propre et équilibré

Points faibles :

1. MLP peu interprétable
2. Modèles d'ensemble sensibles aux hyperparamètres
3. Dataset concentré sur une population jeune

Hypothèses :

- Les relations non linéaires avantagent MLP et Gradient Boosting
- Le scaling aide MLP et SVM

Conclusion et perspectives

Conclusion :

- Projet complet : EDA → préparation → modèles → optimisation → évaluation
- Le MLP optimisé donne les meilleurs résultats
- Très bonne performance globale (97 % accuracy)

Perspectives :

- Tester XGBoost ou LightGBM
- Étudier l'importance des variables
- Élargir le dataset à d'autres groupes d'âge
- Améliorer l'interprétabilité du modèle