Rapport

Septembre 2025

Table des matières

1 Introduction et Présentation des données

La consommation de café est une habitude courante à travers le monde. Cependant, ses effets sur la santé humaine restent un sujet de débat parmi les chercheurs et les professionnels de la santé. Certaines études suggèrent que la consommation modérée de café peut avoir des effets bénéfiques, tels que l'amélioration de la vigilance mentale et la réduction du risque de certaines maladies chroniques. D'autres études mettent en garde contre les effets négatifs potentiels, notamment l'augmentation de l'anxiété, des troubles du sommeil et des problèmes cardiovasculaires.

Notre sujet d'apprentissage statistique porte donc sur la prévision des problèmes de santé d'une personne en fonction de co-variables explicitées dans la suite. Notre base de données est issue de *Global Coffee Health Dataset*.

Le but de ce projet est de prédire la variable cible `Health_Issues` en fonction des autres variables et de classifier au mieux la dernière classe ("Severe").

1.1 Variables qualitatives

- `Health_Issues` : État de santé du participant (0 : None, 1 : Mild, 2 : Moderate, 3 : Severe)
- `Country` : Pays du participant
- `Sleep_Quality` : Qualité du sommeil (0 : Poor, 1 : Fair, 2 : Good, 3 : Excellent)
- `Stress_Level` : Niveau de stress (0 : Low, 1 : Moderate, 2 : High, 3 : Very High)
- `Smoking` : Tabagisme (0 : Non-fumeur, 1 : Fumeur)
- `Alcohol_Consumption` : Consommation d'alcool (0 : Non-buveur, 1 : Buveur)
- `Occupation` : Statut professionnel (Other, Student, Employed, Service, Healthcare)
- `Gender` : Genre (Male, Female, Other)

1.2 Variables quantitatives

- `Caffeine_mg` : Quantité de caféine en mg : 1 cup $\approx$ 95 mg
- `Coffee_Intake` : Consommation de tasses de cafés par jour (0.0–8.2 tasses)
- `Coffee_mg` : Quantité estimée de café en mg (0.0–780.3)
- `Age` : Âge du participant en années (18–80)
- `BMI` : Indice de masse corporelle (15.0–38.2)

— `Physical_Activity_Hours` : Durée moyenne d'activité physique quotidienne en heures (0.0–15.0)
— `Sleep_Hours` : Durée moyenne de sommeil quotidienne en heures (3.0–10.0)
— `Heart_Rate` : Fréquence cardiaque du participant en battements par minute (50–109)

1.3 Quelques visualisations des données

Comme nous nous intéressons principalement à la variable cible `Health_Issues`, nous présentons ici quelques visualisations en lien avec cette variable.

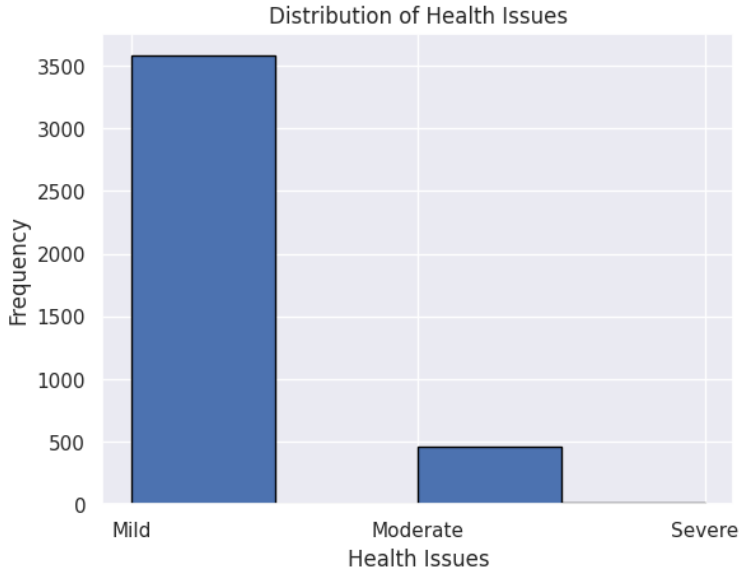

FIGURE 1 – Distribution des problèmes de santé dans le dataset

Nous remarquons que les deux premières classes sont très dominantes et la dernière quasi inexistante. Nous pouvons avancer que sa prédiction sera plus difficile.

De plus, nous avons réalisé une analyse statistique pour identifier les variables ayant un impact significatif sur `Health_Issues`. Nous avons utilisé le test de Kruskal-Wallis pour évaluer l'importance des variables quantitatives.

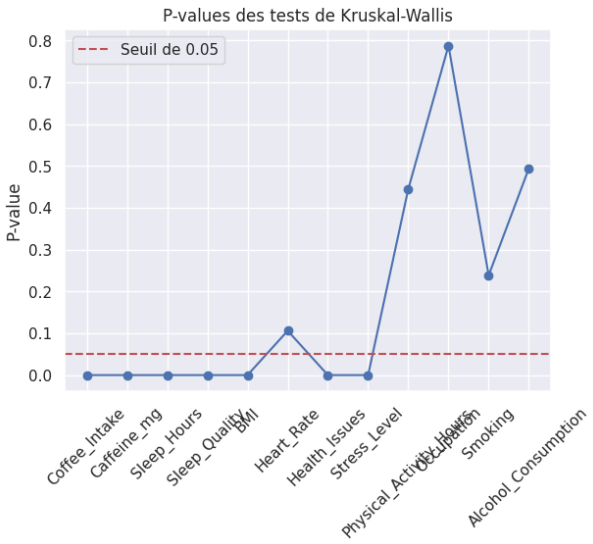

FIGURE 2 – P-values des tests de Kruskal-Wallis pour chaque variable quantitative

Ce test nous révèle que la variable `Health_Issues` est en rejet de l'hypothèse nulle selon laquelle les échantillons indépendants ont la même tendance et sont issus de la même population.

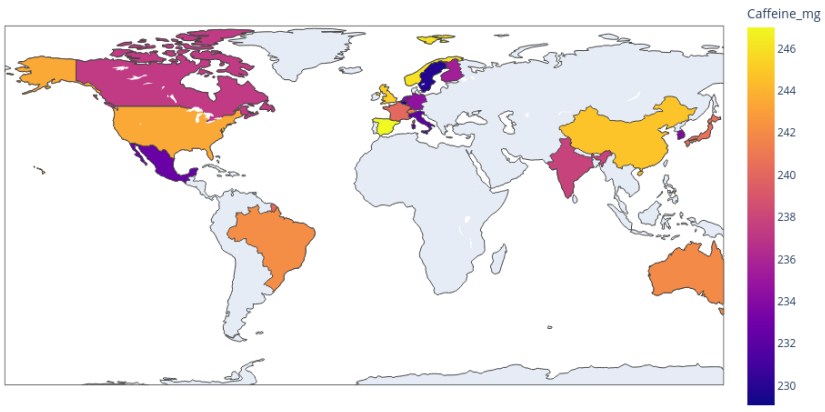

FIGURE 3 – Quantité de café consommé par pays dans le monde

On observe que dans les pays recensés, la consommation de café n'impacte pas directement les problèmes de santé. En posant un regard sur l'ensemble

des figures précédentes, des différences significatives sont présentes entre individus selon plusieurs variables exogènes par rapport à `Health_Issues` c'est pourquoi nous allons normaliser par standardisation avant de modéliser.

1.3.1 Matrice de corrélation et Pairplot

Dans la matrice de corrélation, `Health_Issues` est très corrélée négativement à `Stress_Level` ce qui indique des relations proches mais inversées. Cependant, cette méthode ne capte que des relations linéaires. C'est pourquoi nous appliquons un test de Spearman pour détecter les relations monotones non linéaires.

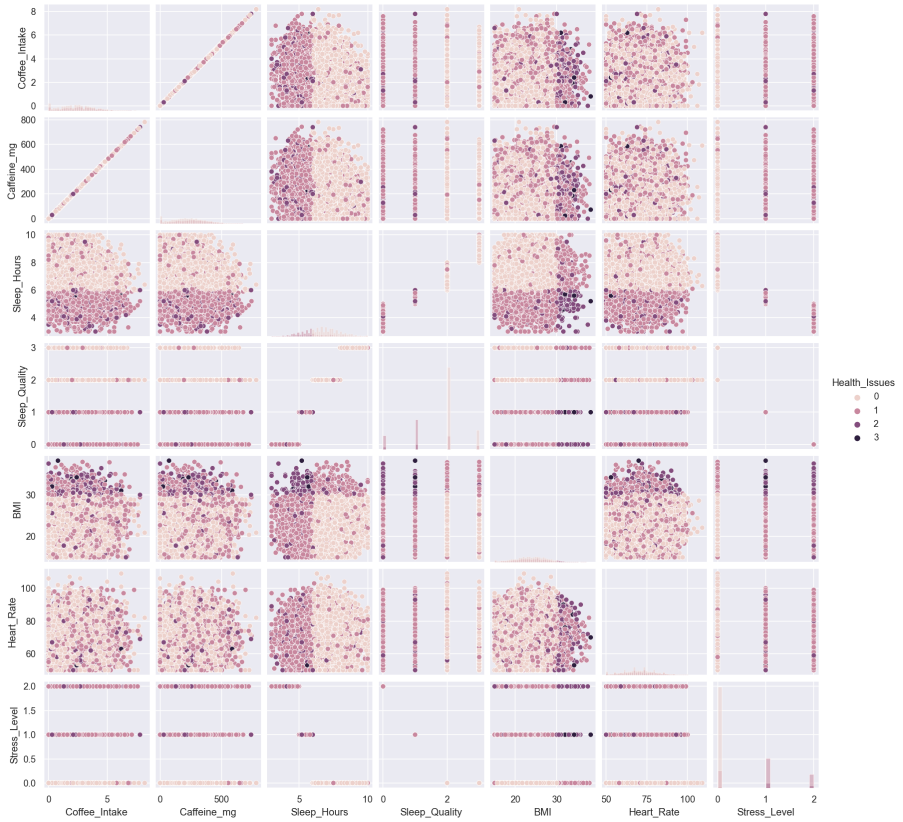

FIGURE 4 – Pairplot sur les corrélations autour de la variable cible

Les clusters ont été formés à partir de la variable cible. En effet, elle est très corrélée avec `Stress_Level` ce qui rejoint les observations faites avec la matrice.

1.3.2 Test de Spearman

Le test de Spearman ne prend pas en compte la linéarisation ni la normalisation ce qui capte de potentielles relations non linéaires avec `Health_Issues`.

Variable	**Coefficient de Spearman**	**p-value**
Coffee_Intake	0.109	5.5e-28
Caffeine_mg	0.109	4.6e-28
Sleep_Hours	−0.628	0.0
Sleep_Quality	−0.700	0.0
BMI	0.149	3.0e-51
Heart_Rate	0.022	0.0267
Health_Issues	1.0	0.0
Stress_Level	0.786	0.0
Physical_Activity_Hours	−0.008	0.394
Occupation	0.010	0.311
Smoking	−0.0045	0.649
Alcohol_Consumption	−0.011	0.257

TABLE 1 – Résultats du test de corrélation de Spearman

Les `Health_Issues` sont fortement liés à `Stress_Level` et à la mauvaise qualité/quantité de sommeil. Ils sont aussi associés, mais plus faiblement, à l'IMC et à la consommation de caféine. Les autres variables (activité physique, tabac, alcool, occupation) ne montrent pas de relation monotone significative. Donc on peut prendre en considération comme variables exogènes celles présentant une corrélation significative.

2 Fonction de perte

Pour chaque modèle, nous avons utilisé la fonction de perte CCELoss (Categorical Cross-Entropy Loss) qui est adaptée pour les problèmes de classification multi-classes. Elle est définie par la formule suivante :

$$\text{CCELoss}(P) = -\langle y, \log(P) \rangle = -\sum_{i=0}^{C-1} y_i \log(p_i)$$

où C est le nombre de classes, y est un vecteur contenant les variables binaires indiquant si la classe i est la vraie classe (1 si oui, 0 sinon) et P est le vecteur des probabilités prédites par le modèle.

L'utilisation de cette fonction de perte dans le contexte d'apprentissage a été explorée par [1]. Elle compare les probabilités prédites par le modèle et les vraies distributions des classes. L'objectif de l'étude des différents modèles est de minimiser cette perte pour améliorer leur précision en prédiction.

Dans cette section, nous détaillerons l'optimisation de la *CCELoss* pour le modèle bayésien pondéré.

De plus, au vue de la différence de nature des classes (de 0 à 3) et leur fréquence, nous avons pondéré les classes pour donner plus d'importance aux classes sous-représentées (comme la classe 3). À noter que dans la suite du rapport, nous avons décalé les classes de 0.. 3 à 1..4 pour que le poids de la classe 0 ne soit pas nul.

Soit ω_i le poids de la classe i (avec $i \in \{0, 1, 2, 3\}$) et N_i le nombre d'échantillons de la classe i. Nous avons défini les poids comme suit :

$$\omega_i = \frac{1/N_i}{\sum_{j=0}^{3} 1/N_j}$$

Avant toute modélisation, nous avons divisé notre dataset en un ensemble d'entraînement, de validation et de test pour évaluer les performances prédictives des modèles. Nous avons aussi normalisé les variables quantitatives pour améliorer la convergence des modèles linéaires et indexé les variables qualitatives en ajoutant 1 aux nouvelles valeurs possibles.

3 Modèle de Régression Logistique

Pour résumer les performances des modèles linéaires, nous présentons une régression logistique multi-classes (ici 4).

Après avoir entraîné ce modèle sur les données d'entraînement et de tests, nous affichons la matrice de confusion du modèle sur le jeu de données complet pour visualiser sa qualité de classification globale (figure **??**). Comme prévu sur des données à priori non linéaires, la régression logistique n'arrive pas à bien classifier les données, en particulier la classe 3 (Severe).

4 Réseaux de Neurones

Ici, nous étudions un Feed Forward Neural Network à plusieurs couches cachées.

L'optimiseur Adam (Adaptive Moment Estimation) a été privilégié pour éviter au plus possible les effets du *vanishing gradient*[1].

1. L'optimisation du gradient sur chaque couche se fait habituellement par descente de

4.1 Optimisation des hyperparamètres avec Optuna

Pour optimiser les performances du réseau de neurones, nous avons utilisé la bibliothèque Optuna qui effectue une recherche bayésienne des hyperparamètres optimaux. Cette approche permet d'explorer efficacement l'espace des hyperparamètres en se concentrant sur les régions prometteuses.

4.1.1 Hyperparamètres recherchés

Les hyperparamètres suivants ont été optimisés sur 50 essais avec un budget temps de 1 heure :

- **Nombre de couches cachées** (`n_layers`) : entre 2 et 4 couches
- **Nombre de neurones par couche** (`n_units_l`i) : entre 32 et 256 neurones (par pas de 32)
- **Taux de dropout** (`dropout_rate`) : entre 0.0 et 0.5
- **Taux d'apprentissage** (`learning_rate`) : entre 10^{-4} et 10^{-2} (échelle logarithmique)
- **Taille de batch** (`batch_size`) : parmi {16, 32, 64, 128}

4.1.2 Stratégie d'optimisation

L'optimisation a été réalisée avec les caractéristiques suivantes :

- **Métrique d'optimisation** : maximisation de l'accuracy sur l'ensemble de validation
- **Pruning** : utilisation du *MedianPruner* pour arrêter prématurément les essais non prometteurs (5 essais de démarrage, 10 étapes de préchauffage)
- **Early stopping** : arrêt de l'entraînement après 10 epochs sans amélioration
- **Nombre maximum d'epochs** : 50 par essai (pour accélérer la recherche)

4.1.3 Résultats de l'optimisation

Les meilleurs hyperparamètres obtenus sont présentés dans le tableau 2 :

gradient. Adam lui utilise la moyenne des gradients passés et la moyenne des carrés des gradients présents pour stabiliser l'optimisation, minimiser le bruit et donc éviter d'avoir un gradient trop petit qui ne peut pas converger efficacement.

Hyperparamètre	Valeur optimale
Nombre de couches cachées	3
Neurones couche 1	256
Neurones couche 2	64
Neurones couche 3	32
Architecture	[256, 64, 32]
Taux de dropout	0.0310
Taux d'apprentissage	9.21×10^{-4}
Taille de batch	32

TABLE 2 – Hyperparamètres optimaux obtenus par Optuna

Avec un tel modèle, nous obtenons les graphiques suivants :

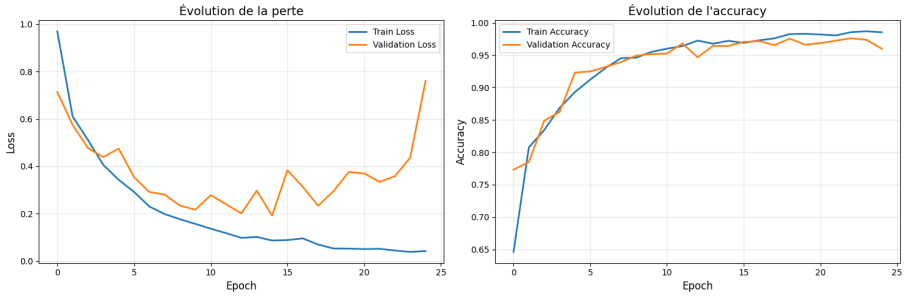

FIGURE 5 – Courbe de perte et d'accuracy sur les données d'entraînement et de validation

Nous remarquons que l'overfitting apparait apres environ 25 époques : moment ou la perte de validation augmente.

La validation accuracy est proche de la training accuracy. Cela signifie que le modèle généralise correctement et qu'il n'est pas en train de sur-apprendre ni de sous-apprendre. Nous avons donc :

- **Stabilité du modèle :** le fait que les deux courbes soient proches montre que le modèle est stable et qu'il n'y a pas de perte d'information entre train et validation.
- **Choix du nombre d'époques :** on arrête l'entraînement quand les deux courbes se stabilisent, ce qui évite le sur-apprentissage.
- **Solidité du modèle :** la cohérence entre train et validation accuracy montre la robustesse du modèle.

4.2 Analyse des résultats

La courbe ROC (Receiver Operating Characteristic) permet d'évaluer les performances du modèle en traçant le taux de vrais positifs (TPR) en fonction du taux de faux positifs (FPR) pour différentes valeurs seuils.

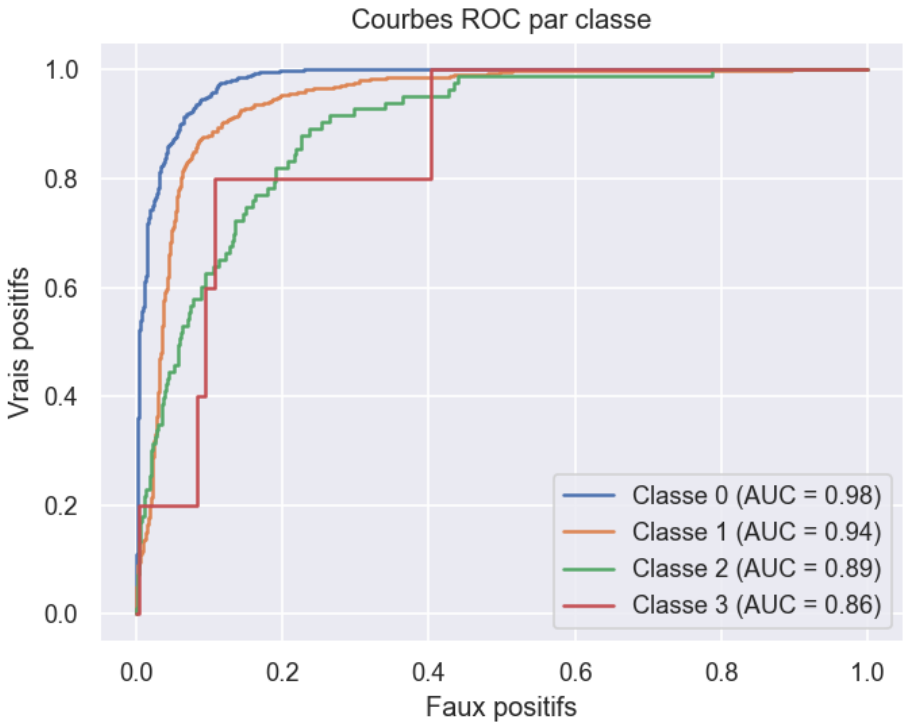

FIGURE 6 – Évaluation des performances du réseau de neurones à action directe

Cette matrice de confusion nous confirme bien que les deux premières classes sont très dominantes sur les autres. Cela a des conséquences sur la performance de la dernière classe où le bruit peut être important et donc réduire la précision des prédictions.

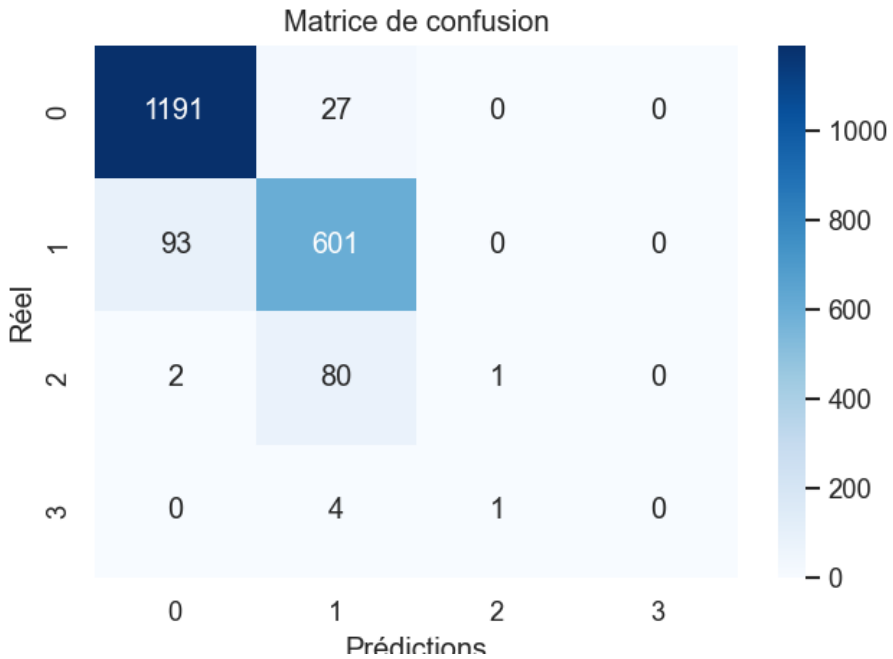

FIGURE 7 – Matrice de confusion de toutes les classes prédites par FFNN

5 SVM Non linéaire

Par des SVM à noyaux non-linéaires :

— polynomiaux : $(\gamma\langle x, x'\rangle + r)^d$, où d est spécifié par le paramètre `degree`, r par `coef0`.

— rbf : $\exp(-\gamma\|x - x'\|^2)$, où γ est spécifié par le paramètre `gamma`, doit être supérieur à 0.

— sigmoid : $\tanh(\gamma\langle x, x'\rangle + r)$, où r est spécifié par `coef0`.

Nous étudions la non-linéarité de la répartition des classes pour déterminer si elle dépend de fonctions classiques ou non.

5.1 Résultats

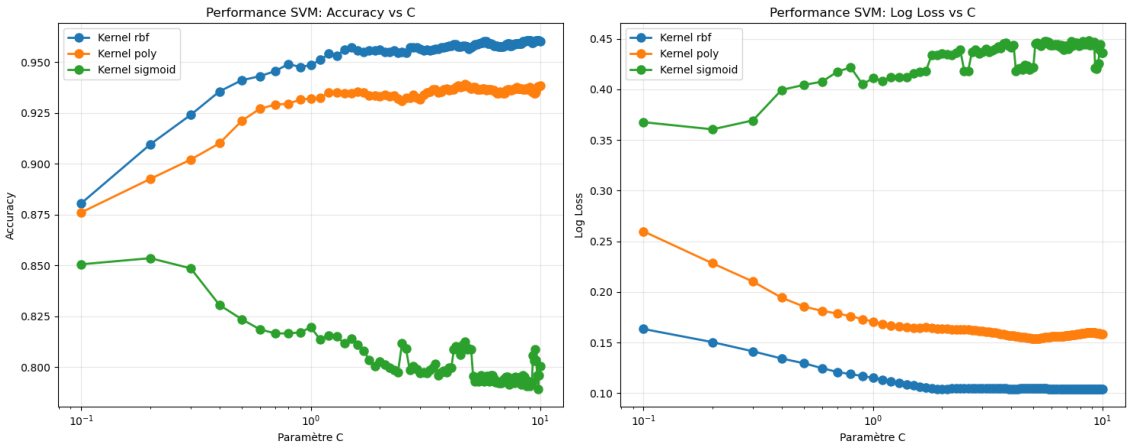

FIGURE 8 – Évolution des performances des modèles SVM non linéaires

Les résultats suivants montrent malheureusement que même le meilleur SVM en termes de perte classifie encore mal les données en modélisation et prédiction. En effet, les précédents modèles ont une LogLoss de presque 0.4.

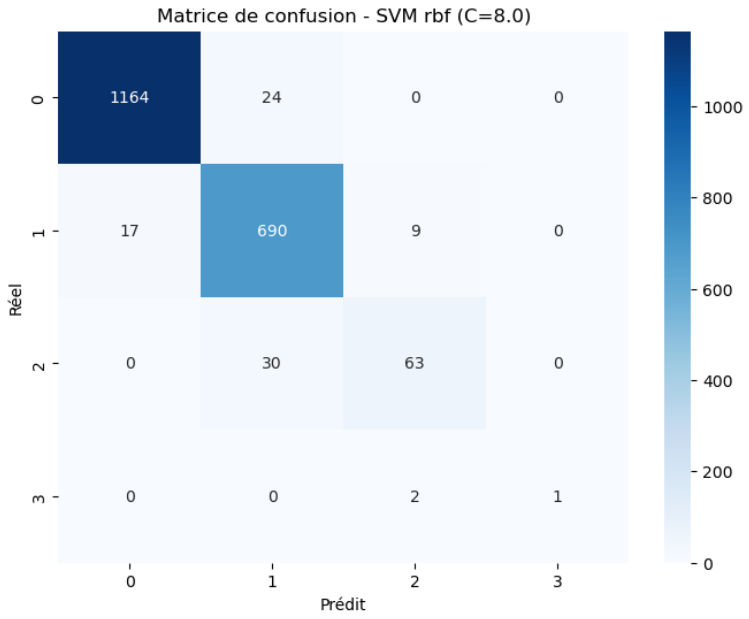

FIGURE 9 – Matrice de confusion pour le meilleur modèle SVM

6 Arbre de Décision

6.0.1 Optimisation de la profondeur

Nous avons testé différentes profondeurs maximales (3, 5, 7, 10, 15, 20, et sans limite) en utilisant une validation croisée. Les hyperparamètres suivants ont été fixés pour éviter le surapprentissage :

- `min_samples_split` = 10 : nombre minimum d'échantillons requis pour diviser un nœud
- `min_samples_leaf` = 5 : nombre minimum d'échantillons dans une feuille
- `class_weight` = 'balanced' : pondération automatique des classes

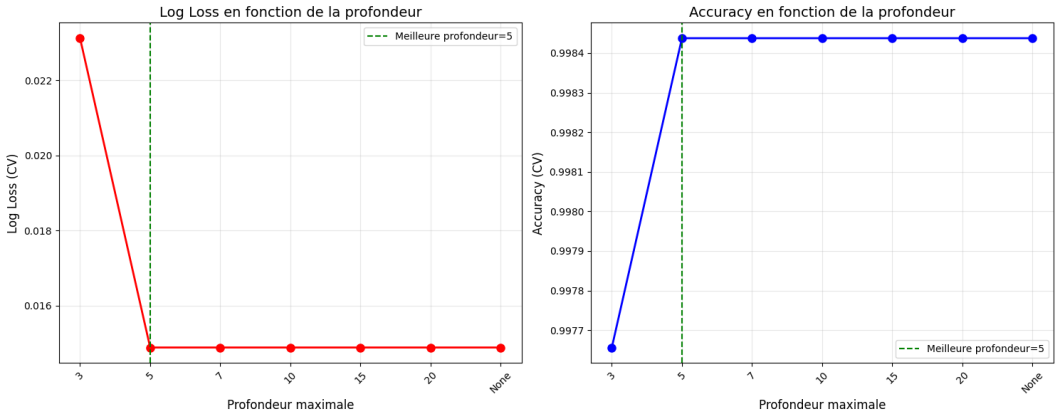

FIGURE 10 – Performance de l'arbre de décision en fonction de la profondeur maximale

Les résultats de la validation croisée montrent que la meilleure profondeur maximale est de [5] avec une Log Loss moyenne de 0.0149 ± 0.0146 et une accuracy de 0.9984 ± 0.0007.

6.0.2 Importance des variables

L'analyse de l'importance des variables révèle les facteurs les plus discriminants pour la prédiction des problèmes de santé :

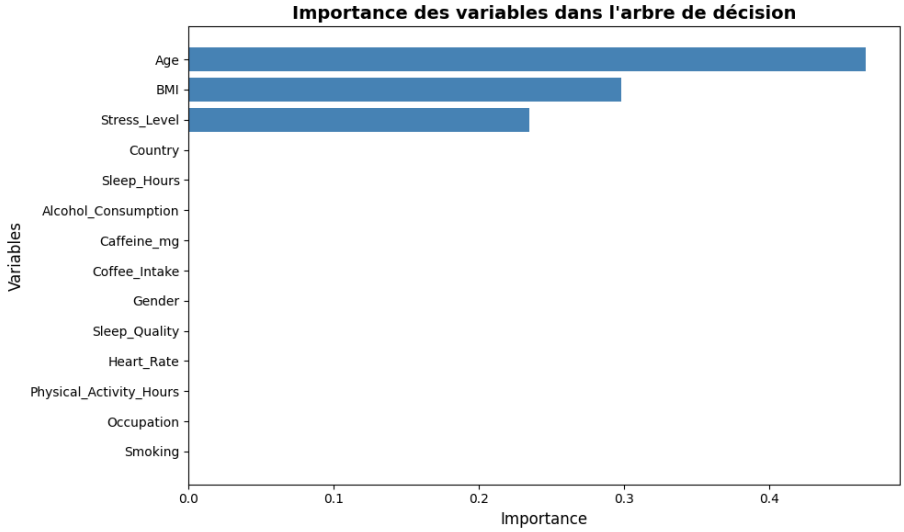

FIGURE 11 – Importance des variables dans l'arbre de décision optimal

Les variables `Stress_Level`, `Age` et `BMI` apparaissent comme les plus importantes, confirmant les observations faites lors de l'analyse exploratoire. Cela indique que ces facteurs sont les principaux déterminants des problèmes de santé dans notre dataset.

6.0.3 Visualisation de l'arbre

Pour faciliter l'interprétation, nous présentons deux visualisations de l'arbre :

Un arbre complet (profondeur maximale = 3) (14) et un arbre complet (profondeur maximale = 5) (15)

Sur l'ensemble de test, l'arbre de décision optimal obtient une Log Loss de 0.061%.

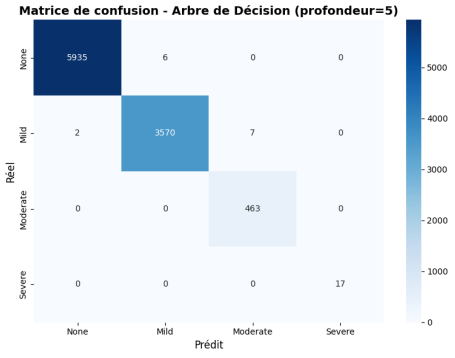

FIGURE 12 – Matrice de confusion de l'arbre de décision

La matrice de confusion révèle que le modèle performe bien sur les classes 1 et 2, mais rencontre des difficultés avec les classes minoritaires 3 et 4, malgré l'utilisation de poids de classes équilibrés. Cela s'explique par le déséquilibre important des données observé dans la matrice de confusion.

7 Conclusion

Ces études révèlent à quel point il y a de fortes disparités entre les classes et les trois algorithmes de prédiction confirment ce point. Cela est passé par l'évaluation des métriques par courbes AUC/ROC, F1-score, l'accuracy, mais aussi des hyperparamètres afin d'optimiser au mieux le nombre de couches cachées entre les neurones afin de pouvoir mieux transmettre l'information, rétropropager et minimiser les erreurs de prédiction.

La dernière classe est très minoritaire elle indique l'état sévère du patient suite à la consommation de café ce qui démontre que le café n'est pas néfaste pour la santé d'autrui, à l'inverse les deux premières classes suggérant des effets très relatifs de la caféine sur les patients montre bien au contraire qu'elle est bénéfique pour la santé.

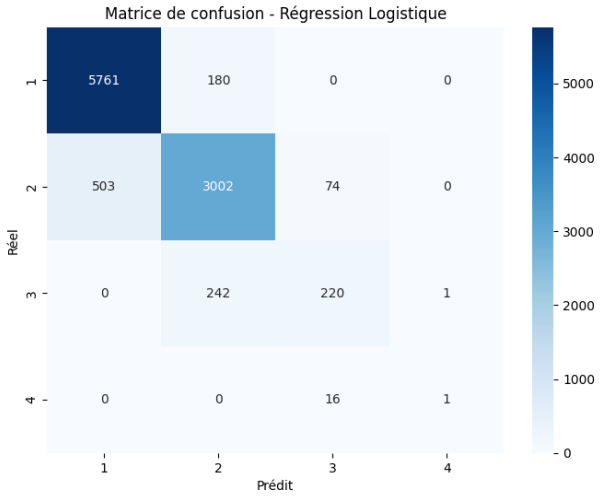

FIGURE 13 – Matrice de confusion d'une régression logistique

Même un cas très simple de régression logistique révèle la diificulté à équilibrer les classes par rapport à leur dominance et leur probabilité de prédiction. Dans ce type de contexte, il est donc assez courant d'avoir de telles disparités entre les classes de `Health_Issues`. Cela reste indispensable

pour les patients de savoir que la caféine ne va pas influencer la santé générale des patients bien au contraire.

8 Annexes

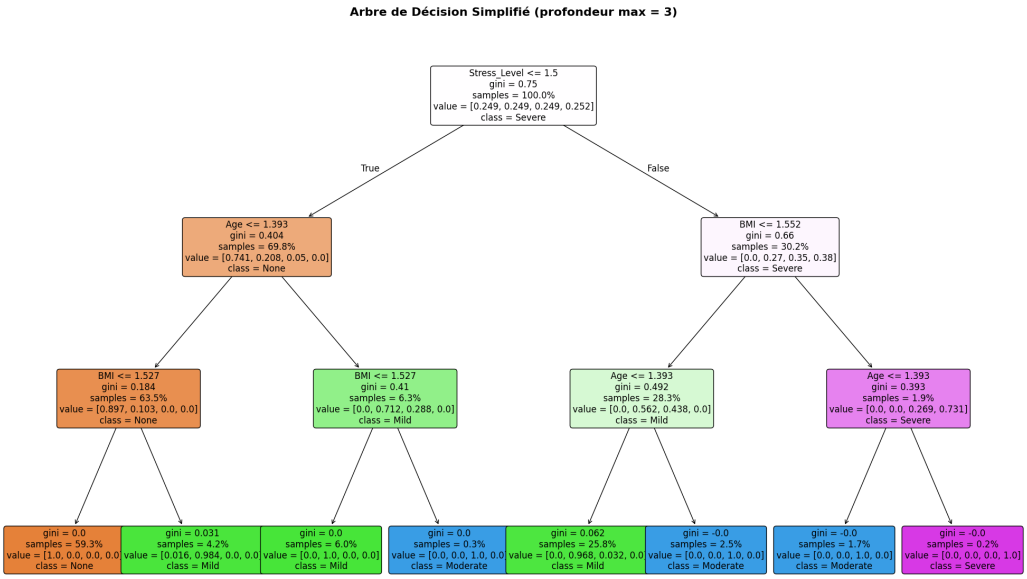

FIGURE 14 – Arbre de décision simplifié (profondeur maximale = 3)

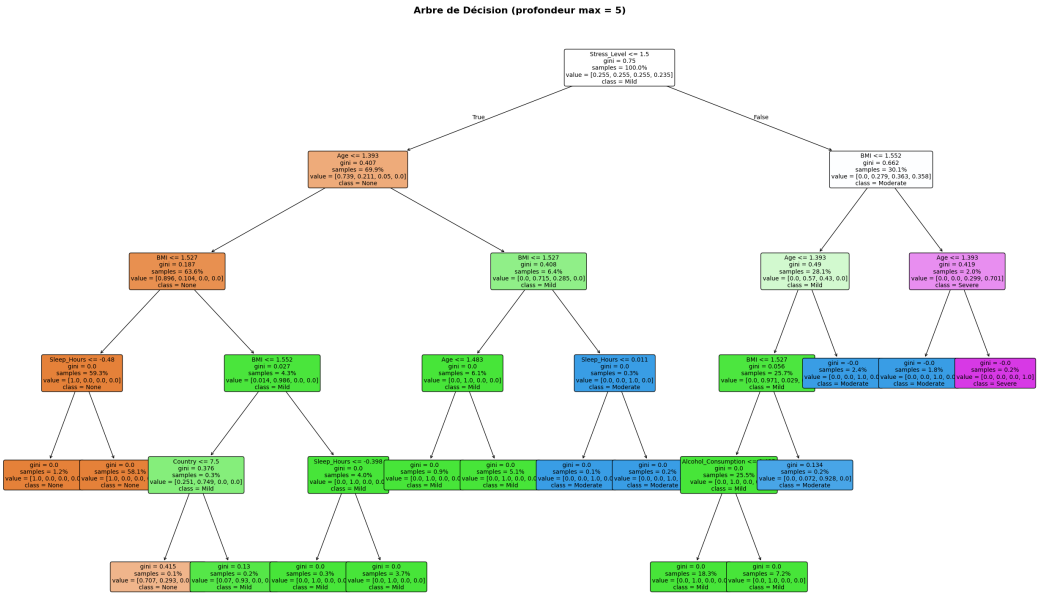

FIGURE 15 – Arbre de décision simplifié (profondeur maximale = 3)

Références

[1] Erwan Scornet. *Vidéos du cours de Machine Learning*. Sorbonne Université, 2025–2026.