Estimation de variables en biologie médicale par différentes méthodes statistiques

Océane Frayssinet

L'institut Agro Rennes Angers
☐ Site d'Angers ☒ Site de Rennes

Bio Logbook

Année universitaire : 2022-2023 Spécialité : Science des données	**Mémoire de fin d'études** ☐ d'ingénieur de l'Institut Agro Rennes Angers (Institut national d'enseignementsupérieur pour l'agriculture, l'alimentation et l'environnement) ☒ de master de l'Institut Agro Rennes Angers (Institut national d'enseignement supérieur pour l'agriculture, l'alimentation et l'environnement) ☐ de l'Institut Agro Montpellier (étudiant arrivé en M2) ☐ d'un autre établissement (étudiant arrivé en M2)

ESTIMATION DE VARIABLES EN BIOLOGIE MEDICALE PAR DIFFERENTES METHODES STATISTIQUES

Par : Océane FRAYSSINET

Soutenu à Rennes le 05/09/2024

Devant le jury composé de :

Maître de stage :
Jakez ROLLAND

Membres du jury :
Laetitia CHAPEL
Jean-Louis MARCHAND

Remerciements :

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de remercier l'équipe de Bio Logbook pour l'accueil qu'elle m'a réservé, ainsi que le temps et les ressources qui ont été mis à ma disposition.

Je souhaiterais tout d'abord remercier Jakez ROLLAND, mon maître de stage qui m'a formée et accompagnée tout au long du projet avec beaucoup de patience et de pédagogie.

J'adresse aussi mes remerciements à Ronan BOUTIN, le créateur de Bio Logbook pour l'accueil dans son entreprise. Je tiens également à remercier les personnes que j'ai eu la chance de côtoyer pendant ces 6 mois : Arnaud LECLEVE, Clément BÉZIER, Leïla ÉQUINET, Marie CODET, Alexandre HOMO et Yann MELLET.

Enfin, j'adresse mes remerciements aux équipes pédagogiques de la spécialité science des données de l'Institut agro Rennes-Angers, pour leur disponibilité et leur pédagogie tout au long de l'année.

Lexique :

— **BCT** : Transformation Box-Cox

— **IP** : Intervalle de Prédiction

— **k-NN** : *k-nearest neighbors* ou k-plus-proches voisins

— **MICE** : *Multiple Imputation by Chained Equations*

— **NN** : *Neural Network* ou réseaux de neurones

— **RF** : *Random Forest* ou forêt aléatoires

— **SVM** : *Support Vector Machine*

— **XGBoost** : *Extreme Gradient Boosting*

Table des matières

1 Introduction

Dans le domaine de la médecine et de la biologie, les analyses de laboratoire, telles que les prises de sang, sont essentielles pour diagnostiquer et surveiller l'état de santé des patients. Cependant, malgré leur importance, ces tests ne mesurent qu'une partie des nombreux paramètres biologiques pertinents. Cette limitation peut empêcher les professionnels de santé d'obtenir une image complète de l'état de santé du patient, les privant ainsi d'informations cruciales pour orienter leur diagnostic.

Grâce aux avancées dans le domaine de l'intelligence artificielle (IA) et du *machine learning*, il est désormais envisageable de combler ces lacunes. Les tests de laboratoire fournissent souvent leurs résultats sous forme de valeurs associées à chaque paramètre, mais une analyse conjointe de ces résultats peut révéler des informations supplémentaires en permettant d'exploiter des interactions entre différents paramètres biologiques. En utilisant des modèles de *machine learning* pour analyser les résultats de laboratoire existants, il est possible de prédire avec précision des paramètres biologiques non mesurés initialement. Par exemple, Luo et al. (2016) ont développé une méthode permettant de prédire le taux de ferritine à partir d'une analyse de sang ne la mesurant pas tandis que Tamune et al. (2020), ont travaillé sur la prédiction de la déficience en vitamine B à l'aide d'analyse de laboratoire de routine. Ces prédictions fournissent aux cliniciens des informations supplémentaires, leur permettant d'effectuer des diagnostics plus complets et de personnaliser davantage les traitements.

Ce projet s'inscrit dans l'un des axes de recherche de la start-up nantaise Bio Logbook, qui vise à enrichir les informations disponibles pour les médecins et à développer des solutions innovantes en médecine prédictive. Le but final étant de fournir les estimations de ces paramètres non mesurés via un logiciel. La figure 1 illustre la visualisation souhaitée pour les polynucléaires éosinophiles : à gauche, les valeurs déjà mesurées sont présentées, tandis qu'à droite, l'estimation de la valeur pour ce même paramètre est affichée, accompagnée d'une courbe représentant la probabilité associée à chaque valeur. Toutefois, pour certains paramètres, l'intervalle de prédiction couvre une très grande majorité des valeurs possibles, ce qui limite la quantité d'informations supplémentaires apportées.

FIGURE 1 – Projection souhaité des estimations dans le logiciel par Bio Logbook.

Actuellement, pour chaque observation (comme les prises de sang ou les prélèvements urinaire par exemple) des modèles adaptatifs sont utilisés car chaque observation possède ses propres paramètres mesurés et non mesurés. Bio Logbook applique un modèle linéaire issu de la régression pas à pas (la méthode sera détaillée ultérieurement). Nous pouvons chercher à améliorer la précision de ces modèles ainsi que de diminuer les intervalles de prédictions.

L'objectif de ce projet est donc d'améliorer les modèles d'estimation existants pour tirer pleinement parti des données de laboratoire grâce aux techniques avancées de *machine learning*. En particulier, nous visons à développer des modèles capables de prédire des paramètres non mesurés avec le plus de précision possible, tout en étant adaptables à toutes les observations possibles. De plus, ces modèles doivent être en mesure de fournir des intervalles de prédiction fiables pour les valeurs prédites, ce qui est essentiel pour garantir la confiance des professionnels de santé dans les résultats obtenus.

2 Présentation de la démarche : Prétraitement des Données, Sélection et Évaluation des Modèles

Le *machine learning*, ou apprentissage machine, est une discipline de l'intelligence artificielle visant à permettre à une machine d'apprendre à partir de données à l'aide de méthodes mathématiques et statistiques. Ce processus se divise en deux étapes principales. La première est la phase d'apprentissage, où le modèle est entraîné en utilisant des exemples où la valeur de la variable à prédire est connue pour construire le modèle. La seconde est la phase de test, où le modèle ainsi formé est utilisé pour prédire une valeur $\hat{Y}$ pour une nouvelle donnée $X = \{X_1, \ldots, X_n\}$.

L'apprentissage machine comprend plusieurs étapes importantes :

1. Comprendre le problème
2. Acquérir les données
3. Traiter et nettoyer les données
4. Extraire des variables prédictives pertinentes
5. Choisir des algorithmes adaptés au problème
6. Optimiser les algorithmes (trouver les hyper-paramètres optimaux)
7. Tester (faire des prédictions sur un jeu de données indépendant pour évaluer l'efficacité de l'algorithme)
8. Déployer l'algorithme

2.1 Présentation du jeu de données

Pour atteindre l'objectif visé, je dispose d'un jeu de données comprenant 2 063 722 observations biologiques et 234 paramètres mesurés, provenant d'un laboratoire de biologie. Ces paramètres incluent à la fois des données numériques et catégorielles, comme le sexe et l'âge. Chaque observation correspond à un résultat biologique d'un patient. Cependant, aucun patient n'a été testé pour l'ensemble des paramètres disponibles, ce qui conduit à un taux de 91% de données manquantes et à ce que 100% des observations soient incomplètes.

2.2 Préparation et transformation des Données

2.2.1 Standardisation des données

Afin de préparer les données pour l'apprentissage machine, nous avons appliqué une normalisation min-max à toutes les variables. Cette technique consiste à transformer les valeurs de chaque caractéristique pour qu'elles soient comprises dans une échelle commune de 0 à 1. La formule utilisée pour cette transformation est la suivante :

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

où X représente la valeur d'origine, $X_{\min}$ et $X_{\max}$ sont respectivement les valeurs minimale et maximale de la caractéristique X.

Cette normalisation est surtout utile pour les algorithmes basés sur la distance que nous verrons dans la section suivante, car elle garantit que toutes les caractéristiques contribuent de manière égale au calcul des distances, tout en conservant la distribution originale des données. En effet, sans normalisation, les variables avec des échelles plus larges pourraient dominer la mesure de distance et biaiser les résultats du modèle.

2.2.2 Transformation des données

Il est souvent bénéfique de transformer la variable à prédire Y, afin d'améliorer la distribution et la normalité des données, ce qui est essentiel pour de nombreux algorithmes de modélisation. Les variables cibles testées au cours du stage présentaient rarement une distribution normale, rendant ces transformations particulièrement utiles. Nous allons tester la transformation logarithmique qui va compresser les valeurs extrêmes, réduisant ainsi l'impact des valeurs aberrantes et rendant la relation entre les variables plus linéaire. De même, la transformation de Box-Cox (Box et al. (1964)) permet d'ajuster la distribution des données en optimisant un paramètre lambda, ce qui maximise la normalité et l'homoscédasticité des résidus du modèle : des hypothèses nécessaires pour appliquer un modèle linéaire par exemple.

Pour toute valeur de x positive, on définit la transformée de Box-Cox de la manière suivante :

$$B(x, \lambda) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{si } \lambda \neq 0 \\ \log(x) & \text{si } \lambda = 0 \end{cases}$$

Nous avons ensuite inversé ces transformations sur les valeurs prédites afin de calculer les valeurs prédites dans des unités non transformées. La figure 2 présente des QQ plots (Quantile-Quantile plots) qui comparent les quantiles d'une distribution normale théorique avec ceux de la distribution observée des données, appliquées à différentes transformations. Dans cet exemple, nous utilisons l'albumine comme variable d'intérêt pour illustrer ces transformations. Les QQ plots nous permettent d'évaluer graphiquement la normalité des données : plus les données suivent une distribution normale plus elles se rapprochent de la ligne rouge. Dans ce cas précis, bien que plusieurs transformations aient été testées pour ajuster la distribution de l'albumine, aucune ne semble aligner les données de manière satisfaisante avec une distribution normale car les points s'écartent de la ligne droite représentant la normalité attendue. Cependant, l'absence de normalité parfaite après transformation ne signifie pas nécessairement que les transformations sont inutiles. Ainsi, malgré le fait que les transformations ne produisent pas une normalité complète, nous allons tester les différentes méthodes sur les algorithmes de prédiction pour évaluer leur impact sur la performance des modèles. En effet, certains algorithmes peuvent béné-

ficier de ces transformations en réduisant l'impact des valeurs ou en stabilisant la variance des données. Cette approche expérimentale permettra de déterminer si les transformations appliquées conduisent à une amélioration dans le cadre des modèles prédictifs que nous explorons.

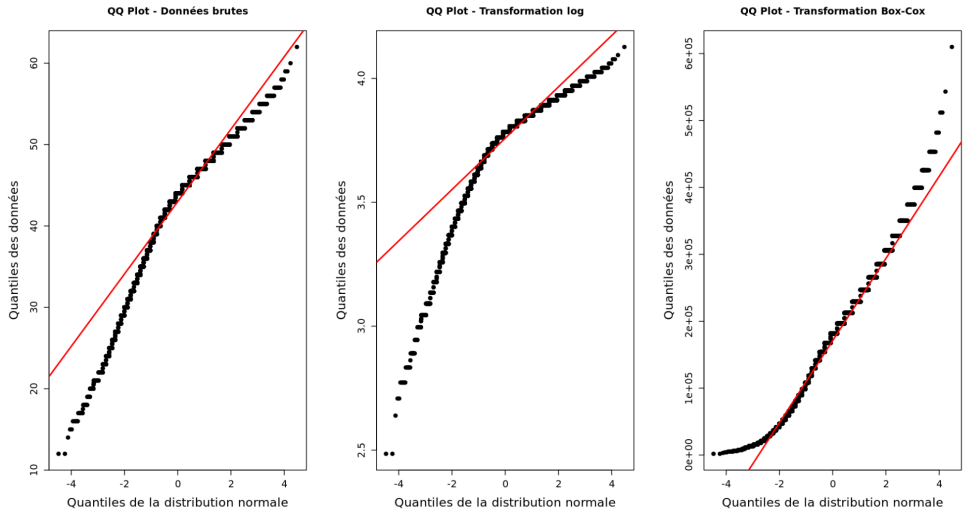


FIGURE 2 – QQ Plots des données brutes et après transformations log et Box-Cox.

2.3 Construction des modèles

Dans le cadre de notre étude, nous avons testé trois types d'algorithmes pour réaliser des prédictions : la régression linéaire multiple (actuellement utilisée par Bio Logbook), les algorithmes basés sur les arbres de décision, et ceux basés sur la distance entre deux observations, afin de diversifier les modèles. Bien que la possibilité d'utiliser des réseaux de neurones ait été envisagée, elle n'a pas été retenue en raison du temps considérable nécessaire pour optimiser leur architecture.

2.3.1 Modèles linéaires

La régression linéaire est un modèle d'apprentissage supervisé qui consiste à trouver par ajustement affine une droite pouvant expliquer la variable à prédire Y à partir de la variable X. On parle de régression linéaire multiple quand on a plusieurs variables explicatives $\{X_1, \ldots, X_n\}$. La variable expliquée s'exprime alors comme une fonction linéaire des variables explicatives :

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \varepsilon$$

Avec β_n un coefficient à estimer et ε représentant l'écart entre la valeur observée Y et la valeur exprimée par la régression linéaire.

Les coefficients Beta peuvent être calculés de différentes manières mais la méthode la plus utilisée est celle des moindres carrées qui cherche à choisir les coefficients Beta qui minimisent les erreurs de prédiction epsilon.

C'est cette méthode qui est utilisé actuellement dans l'entreprise Bio Logbook . Cependant, les relations linéaires ne sont adaptés que rarement aux paramètres biologiques, c'est pourquoi nous allons tester d'autres méthodes.

2.3.2 Modèles basés sur les arbres de décisions

Le Random Forest et XGBoost sont particulièrement adaptés pour leur capacité à gérer un grand nombre de variables tout en capturant des interactions complexes entre celles-ci.

L'algorithme Random Forest, ou forêt aléatoire, est un algorithme de régression et de classification proposé par Breiman (2001). Il utilise le bagging (bootstrap aggregation), une méthode ensembliste qui consiste à générer plusieurs jeux de données d'apprentissage différents par tirage aléatoire avec remplacement. Les variables sont également sélectionnées de manière aléatoire, une technique appelée feature sampling. Les forêts aléatoires reposent sur le principe des arbres de décision. Un arbre de décision est un outil qui modélise une hiérarchie de tests pour prédire un résultat. Chaque feuille de l'arbre correspond à une décision, tandis que les branches représentent les choix logiques menant à une autre branche ou à une feuille. Les arbres de décision sont construits par des séparations successives de leurs nœuds selon différents critères de séparation. Dans l'implémentation utilisée (ranger Wright et al. (2023)), nous employons le critère de Gini, qui dans le contexte de la régression, est adapté pour maximiser la pureté des sous-ensembles après chaque division, ce qui revient à réduire la variance au sein de chaque nœud. L'algorithme Random Forest génère un ensemble de n arbres de décision, chacun ayant une vision partielle des données car construit à partir d'un sous-ensemble de données et d'un sous-ensemble de variables. La prédiction finale du modèle est obtenue en prenant la moyenne des prédictions de tous les arbres. Les hyperparamètres optimisés dans notre cas incluent notamment le nombre d'arbres et la profondeur des arbres. En effet, un arbre trop profond peut entraîner un sur-apprentissage (overfitting) car, plus il dispose de feuilles et de branches, plus il devient complexe et moins il est capable de se généraliser à d'autres données. Par conséquent, nous cherchons à construire des arbres aussi petits que possible tout en conservant leur capacité prédictive.

Le gradient boosting, quant à lui est également une méthode d'ensemble qui combine également plusieurs arbres de décision, pour créer un modèle robuste mais contrairement aux forêts aléatoires qui utilisent le bagging, où les arbres sont construits en parallèle, le gradient boosting construit les arbres de manière séquentielle. Chaque nouvel arbre est formé pour corriger les erreurs des prédictions de l'arbre précédent. Cela se fait en ajustant les résidus pour minimiser une fonction de perte (ici l'erreur quadratique moyenne). En ajoutant des arbres qui ciblent spécifiquement les erreurs résiduelles, le modèle devient progressivement plus précis. XGBoost est une implémentation améliorée du gradient boosting qui pénalise des modèles très complexes et optimise la mémoire de calcul. En plus de la diversité et de la profondeur des arbres comme pour les forêts aléatoires, les hyperparamètres à optimiser concerne le taux d'apprentissage (*learning rate)* et les paramètres pénalisant la complexité du modèle.

2.3.3 Modèles basés sur la distance

Le k-NN pour *k-Nearest Neighbors*, ou méthode des k plus-proches-voisins, est un algorithme de *machine learning* basé sur la similarité des données. Il fonctionne en identifiant les k observations les plus proches (ou "voisins") d'une nouvelle observation à prédire. La distance entre les points de données peut être mesuré à l'aide de plusieurs méthode. Ici la distance de Hamming (explication) et la distance euclidienne ont été testées pour finalement conserver la distance euclidienne, celle-ci donnant les meilleurs résultats.

Le SVM pour la régression (pour *Support Machine Vector*), est un algorithme qui cherche une fonction qui s'écarte des valeurs réelles de moins d'une certaine quantité, appelée epsilon, pour toutes les données d'entraînement. En même temps, il essaie de maximiser la marge autour

de cette fonction. Le SVM utilise des noyaux (ceci permet de projeter les données dans un espace de dimension plus élevée) pour transformer les données dans un espace de plus haute dimension, ce qui permet de modéliser des relations non linéaires. Le SVM est puissant pour gérer des données complexes et peut être plus précis que k-NN pour des problèmes de régression non linéaire, mais il est beaucoup plus coûteux en terme de calcul. Pour optimiser un modèle de SVM, on doit notamment optimiser le paramètre C qui contrôle le compromis entre la marge et les erreurs de prédiction et epsilon qui définit la tolérance des erreurs autour de la fonction de régression.

Pour les algorithmes qui le nécessitaient, les variables catégorielles ont été transformées en variables dummy variable grâce au one-hot encoding pour permettre leur utilisation avec des algorithmes qui ne gèrent pas directement les données catégorielles. Les hyperparamètres des modèles ont été choisis par validation croisée afin d'optimiser leur performance sur les données disponibles.

2.4 Évaluation des modèles

La qualité de l'estimation sera évaluée en utilisant deux métriques principales :

— **Le RMSE (Root Mean Square Error)** : Le RMSE mesure l'écart type des résidus (erreurs de prédiction). Il a l'avantage d'être facilement optimisable, notamment pour les algorithmes basés sur le gradient tel que le XGBoost, et pénalise davantage les grandes erreurs que les petites, offrant ainsi une mesure plus sensible aux erreurs importantes. La formule du RMSE est donnée par :

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

où y_i représente la valeur observée, $\hat{y}_i$ représente la valeur prédite, et n est le nombre total d'observations.

— **Le MAPE (Mean Absolute Percentage Error)** : Le MAPE exprime les erreurs en pourcentage, ce qui facilite la comparaison entre différents paramètres ayant des unités différentes. Il est particulièrement utile pour évaluer les performances de prédiction dans des contextes où les erreurs relatives sont plus significatives que les erreurs absolues. La formule du MAPE est donnée par :

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

où y_i représente la valeur observée, $\hat{y}_i$ représente la valeur prédite, et n est le nombre total d'observations.

Pour évaluer la performance des différents modèles de prédiction que nous avons testés, nous utiliserons une analyse de variance (ANOVA) à un facteur avec mesures répétées, en considérant chaque jeu de données comme un individu et chaque modèle de prédiction comme une répétition de la mesure de prédiction.

3 Etablissement de la stratégie de gestion de données manquantes

Comme écrit ci-dessus, aucune observation n'est complète, ce qui complique l'analyse, car la plupart des algorithmes de *machine learning* ne gèrent pas directement les valeurs manquantes. Il y a 91% de données non mesurées et 100% d'observations incomplètes. Parmi ces données, on trouve des paramètres fréquemment mesurés (comme les paramètres de la numération de formule sanguine telles que le volume plaquettaire, les érythrocytes et leucocytes qui sont mesurés dans 51% des observations), des paramètres moyennement rares (comme l'albumine, mesurée dans 5% des observations) et des paramètres très rares (par exemple les marqueurs tumoraux, présents dans moins de 1 % des observations).

Bien que nous parlions de données manquantes, il s'agit en réalité de données non mesurées. Les données manquantes peuvent être classées en deux catégories principales :

1. **Données Manquantes Aléatoirement (MAR)** : La probabilité que ces données soient manquantes est liée à une ou plusieurs variables observées, mais pas aux valeurs des données manquantes elles-mêmes. Autrement dit, le fait que certaines données soient manquantes peut être expliqué par d'autres variables que nous avons mesurées.
2. **Données Manquantes Non Aléatoirement (NMAR)** : La probabilité que ces données soient manquantes dépend directement des valeurs de la variable manquante elle-même. En d'autres termes, l'absence de ces données est liée aux valeurs qui auraient été mesurées.

Dans notre contexte, les données non mesurées sont souvent dues à des décisions basées sur les valeurs des autres variables observées. Par exemple, un médecin pourrait choisir de mesurer certaines variables en fonction de la suspicion qu'elles pourraient présenter des valeurs anormales. Si la probabilité de non-mesure est donc associée à ces variables observées, on peut considérer que les données non mesurées sont manquantes aléatoirement (MAR). Cette hypothèse permet d'appliquer des techniques d'imputation pour compléter les données manquantes. Toutefois, cette affirmation sera examinée plus en détail dans la section discussion de notre étude.

Le problème est que les algorithmes de *machine learning* mentionnés ne gèrent pas directement ou correctement les données manquantes. Pour surmonter le problème d'observations incomplètes, deux approches principales ont été envisagées en plus de celle adoptée par Bio Logbook avant mon arrivée :

1. **Méthode Bio Logbook. Sélection des modèles par régression stepwise** : Cette méthode repose sur l'utilisation de la régression stepwise. L'article publié par Bio Logbook (Boutin et al. (2024)) décrit la mise en œuvre de cette variante de régression step wise. Afin de palier au manque d'observations complètes, à chaque step, lorsqu'un paramètre est ajouté au modèle, la méthode de Bio Logbook reconstruit un jeu de données avec des individus possédant à une date donnée une valeur pour l'ensemble des paramètres du modèle. Il faut au moins 120 individus. La transformation logarithmique est effectuée sur la variable réponse, ce qui semblait améliorer les performances des modèles. Cette approche conduit à la création de milliers de modèles, ce qui s'avère être à la fois coûteux en termes de calculs et de stockage, mais aussi dans la manière de sélectionner un modèle plutôt qu'un autre pour estimer les résultats d'un patient.
2. **Méthode 1. Création de sous-ensembles de données complets** : Cette méthode consiste à regrouper les observations ayant les mêmes paramètres mesurés pour former des ensembles de données complets, bien que plus petits. Cela facilite l'application des

modèles de *machine learning*, même si ces modèles doivent être ajustés avec un nombre restreint de variables pour s'adapter à un maximum d'observations. Cette approche simplifie l'analyse statistique et l'entraînement des modèles, mais réduit la taille de l'ensemble de données et peut introduire un biais puisqu'elle regroupe les observations des patients ayant les mêmes variables mesurées.

3. **Méthode 2. Imputation des paramètres explicatifs** : Cette méthode consiste à regrouper les observations ayant les mêmes paramètres mesurés pour former des ensembles de données complets, bien que plus petits. Cela facilite l'application des modèles de *machine learning*, même si ces modèles doivent être ajustés avec un nombre restreint de variables pour s'adapter à un maximum d'observations. Cette approche simplifie l'analyse statistique et l'entraînement des modèles, mais réduit la taille de l'ensemble de données et peut introduire un biais puisqu'elle regroupe les observations des patients ayant les mêmes variables mesurées.

Pour garantir la praticité des modèles, nous excluons les paramètres qui sont quasiment toujours mesurés avec la variable à prédire (par exemple le sodium et le potassium pour évaluer les déséquilibres acido-basiques) ou au contraire jamais mesurés avec car dans les faits le modèle ne serait pas utilisable.

3.1 Détail méthode 1

Dans cette méthode, il est question de faire des jeux de données complets contenant moins de paramètres mais dont chaque observation est complète. Pour cela, nous sélectionnons les 100 combinaisons de paramètres les plus fréquemment mesurées ensemble, ce qui nous permet de créer 100 ensembles de données distincts pour entraîner nos modèles à plusieurs reprises (une observation peut cependant se retrouver dans plusieurs jeux de données). Ces combinaisons fréquentes regroupent entre 40 et 50 paramètres à chaque fois. Nous éliminons les ensembles de données qui comportent moins de 120 observations car ça nous permet d'avoir au minimum 96 individus dans l'entrainement du modèle (80% des observations en entrainement et 20% en test). En conséquence, nous obtenons des ensembles de données complets correspondant aux configurations de prises de sang les plus couramment prescrites, ce qui devrait être particulièrement pertinent pour notre analyse. Enfin, la dernière étape consiste à de nouveau sélectionner des variables pour que nos 100 modèles crées ayant un nombre de variables réduits puissent s'adapter à la totalité des observations (voir partie 4.3).

3.1.1 Sélection des Variables des modèles (pour sous-jeux de données complets)

La création de sous-ensembles de données complets présente un défi majeur : il devient nécessaire de développer plusieurs modèles distincts pour prédire chaque nouvelle observation, en fonction des variables mesurées. Actuellement, Bio Logbook s'appuie sur des modèles de régression linéaire développés à l'aide de la méthode de régression pas à pas (stepwise regression). Cette méthode permet de sélectionner automatiquement les variables les plus pertinentes pour le modèle en les ajoutant ou en les supprimant de manière séquentielle. Concrètement, la régression pas à pas procède en évaluant, à chaque étape, l'impact de l'ajout ou du retrait d'une variable sur la performance globale du modèle. Les variables sont ajoutées ou supprimées une par une en fonction de leur significativité statistique et de leur capacité à améliorer le modèle, mesurée par des critères comme le Critère d'Information Bayésien (BIC).
Cependant, une approche alternative consiste à utiliser des modèles où seules les variables les

plus importantes sont sélectionnées. Par exemple, un modèle XGBoost peut être entraîné sur l'ensemble des variables, puis les variables les plus importantes peuvent être extraites pour construire un modèle plus efficace. Cette méthode garantit que seules les variables les plus informatives sont utilisées, ce qui peut améliorer la performance du modèle en évitant le surapprentissage et en réduisant la complexité. L'intérêt c'est qu'on peut choisir le nombre de variables à garder, le choix de garder 5 variables explicatives dans le modèle nous permet de couvrir plus de 99% des observations mais cette méthode s'applique uniquement aux algorithmes Random Forest et XGBoost car cela nécessite de pouvoir classer les variables par importance.

3.2 Détail méthode 2

Pour optimiser cette méthode qui consiste à imputer des données non mesurées, il est nécessaire de sélectionner préalablement les paramètres afin d'éviter d'avoir trop de valeurs manquantes et des paramètres trop corrélés. Un premier filtre consiste à éliminer les paramètres qui sont rarement mesurés en même temps que la variable cible. Ensuite, un second filtre est appliqué pour supprimer les variables peu pertinentes à l'aide d'une sélection pas à pas (stepwise), comme mentionné précédemment. Bien que certaines valeurs ne soient pas techniquement manquantes, elles seront traitées comme telles en utilisant des méthodes d'imputation, telles que la méthode MICE (Multiple Imputation by Chained Equations) et l'imputation par la moyenne. Le principe de MICE repose sur l'idée que chaque variable contenant des valeurs manquantes peut être prédite par les autres variables du jeu de données. L'algorithme procède par itérations : pour chaque variable, il crée un modèle de régression utilisant les autres variables comme prédicteurs pour estimer les valeurs manquantes. Une fois les valeurs imputées pour une variable, l'algorithme passe à la suivante, répétant ce processus pour toutes les variables concernées. Lors du test d'une nouvelle observation, les données incomplètes sont complétées par l'imputation k-NN ou la moyenne, permettant ainsi d'obtenir une observation complète et d'appliquer le modèle choisi. Le choix de l'imputation k-NN est justifié par sa capacité à traiter les données sans nécessiter un processus itératif complexe, rendant ainsi le traitement plus rapide et adaptable.

Dans la figure 3, nous avons un aperçu des données qui sont conservées pour la prédiction de l'albumine grâce à cette méthode. Certaines variables montrent une grande proportion de données manquantes, ce qui indique la variabilité de la proportion de données manquantes entre les variables. La représentation des variables les plus corrélées avec l'albumine permet d'identifier rapidement celles qui pourraient avoir le plus grand impact dans la prédiction, par exemple la CRP (C-Reactive Protein) est la plus corrélée négativement avec l'albumine. Cette figure illustre les défis posés par la nature incomplète des données et sert de base pour les étapes ultérieures de sélection de variables et d'ajustement des modèles.

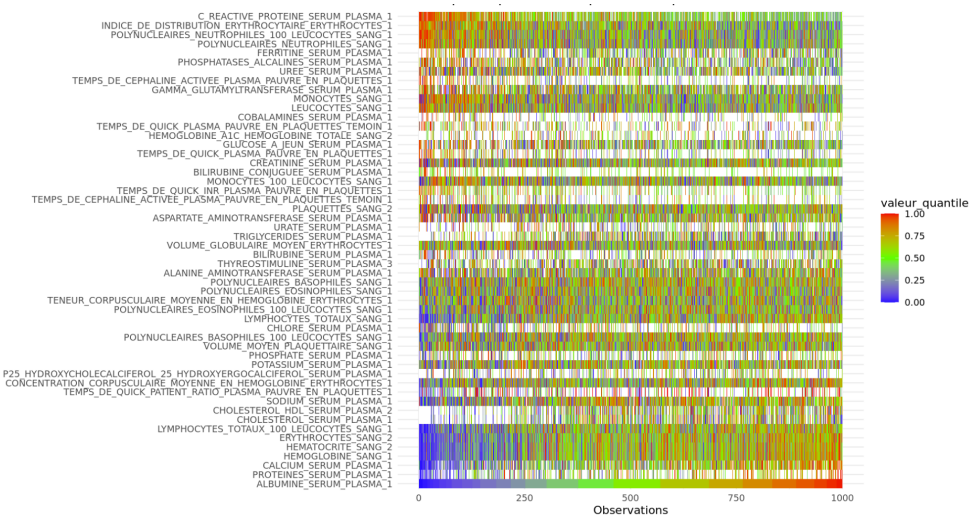


FIGURE 3 – Vue d'ensemble des données et corrélations avec le paramètre albumine avant l'imputation de données. Dans ce heatmap, les lignes représentent les variables conservées dans la prédiction de l'albumine et les colonnes représentent les cas de patients. Les couleurs des cellules sont basées sur les quantiles des résultats (par exemple, le 75e percentile = 0,75), et les résultats manquants sont blancs. Les paramètres sont ordonnés par corrélation croissante avec l'albumine, et les observations sont ordonnées de l'albumine la plus faible à la plus élevée. Les variables qualitatives ne sont pas représentées, seules les variables les plus fréquemment mesurées ont été conservées. 1000 cas ont été représentés.

Pour comparer ces deux approches, nous utiliserons les mêmes observations des jeux de données entre les deux stratégies bien que les paramètres puissent être différents.

4 Résumé de la démarche expérimentale

La figure 4 ci-dessous illustre les étapes entre les différentes méthodes. Les méthodes de calcul des intervalles de prédiction seront détaillés dans une autre partie.

	Modèle actuel	Modèles testés	
Jeu de données	Sous-jeu de données complet		Jeu imputé
Sélection de paramètre	Stepwise regression	Variables d'importance	Stepwise regression
Transformation de la variable à prédire	Logarithme	Pas de transformation Logarithme Boxcox	
Modèles testés	Linéaire	Random Forest XGboost K-NN SVM pour la régression	
Intervalle de prédiction	Standard deviation du modèle linéaire	Bootstrap non paramétrique	
Métriques de performance	RMSE MAPE		

FIGURE 4 – Résumé de la démarche

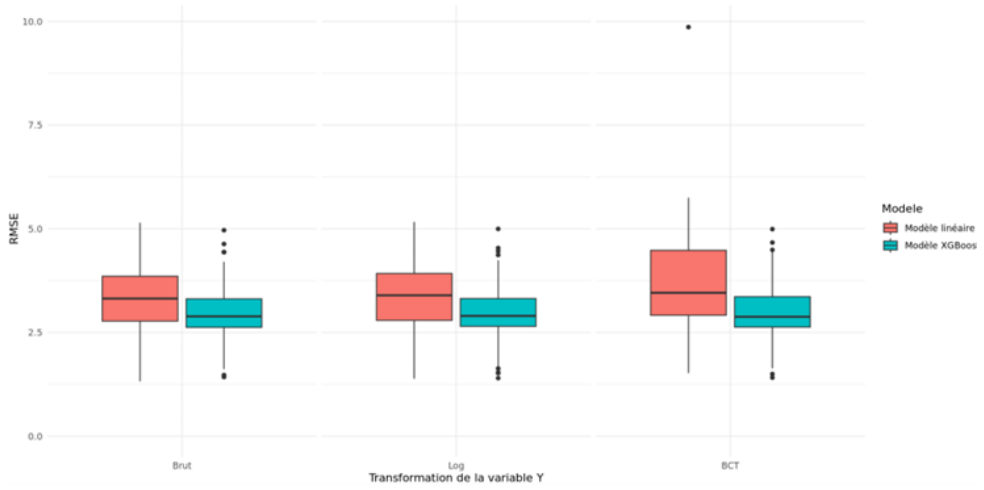
5 Résultats sur plusieurs variables

5.1 Exemple sur l'albumine

Dans cette section, nous allons nous concentrer sur la prédiction de l'albumine, une protéine essentielle produite par le foie, dont les niveaux dans le sang sont souvent utilisés comme indicateur de l'état nutritionnel des patients. L'albumine est particulièrement importante dans le contexte médical car une diminution de sa concentration peut signaler une malnutrition, une inflammation chronique, ou d'autres conditions médicales graves selon la Haute Autorité de Santé. Cependant, malgré son importance clinique, l'albumine n'est mesurée que dans une proportion très limitée des cas, représentant seulement 5% de l'ensemble des mesures disponibles.

5.1.1 Méthode avec sous-jeux de données complets

Dans un premier temps nous allons tester la méthode qui consiste à utiliser des sous-jeux de données. La figure 5 illustre la comparaison des RMSE et des MAPE pour deux modèles : un modèle linéaire (utilisé par Bio Logbook) et un modèle non linéaire (XGBoost). Ces comparaisons ont été effectuées sur 100 jeux de données différents, obtenus via la méthode des combinaisons les plus fréquentes comme expliqué ci-dessus, avec trois approches différentes pour transformer la variable Y : aucune transformation, transformation logarithmique, et transformation de Box-Cox.



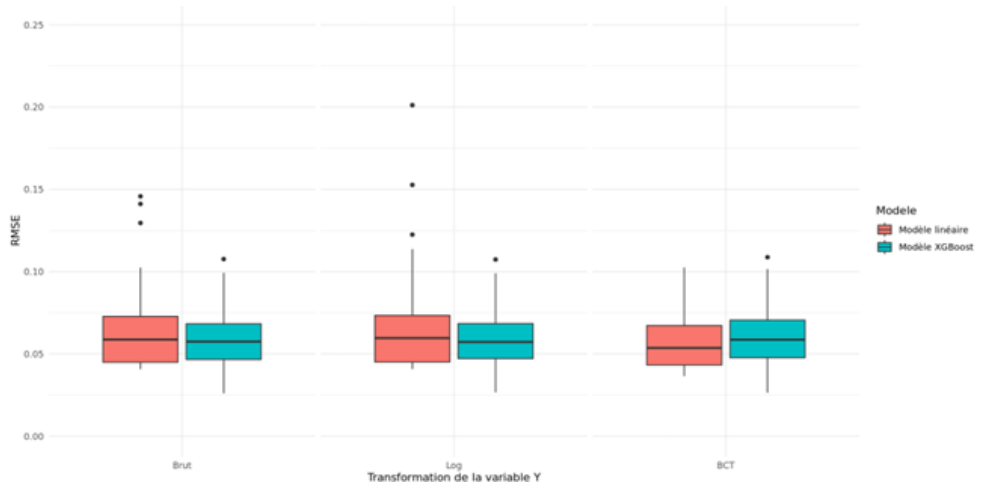


FIGURE 5 – En haut : Comparaison des RMSE pour différentes transformations de la variable Y (Brut, Log, BCT) et deux modèles (linéaire et XGBoost). En bas : Comparaison des MAPE pour les mêmes configurations.

Les résultats des ANOVA montrent une différence significative entre les RMSE du modèle linéaire et de XGBoost (p-value = 0,045), ce qui indique que le modèle XGBoost performe différemment du modèle linéaire. Toutefois, aucune différence significative n'a été observée entre les différentes transformations de la variable Y (p-value = 0,093). Par conséquent, il est plus simple de conserver la transformation qui demande le moins de modification, c'est-à-dire sans transformation.
Cependant on note que, lorsque l'on compare uniquement les modèles linéaires et XGBoost sans appliquer de transformation, aucune différence significative n'a été observée par la comparaison par paire (p-value = 0.25). Les résultats similaires obtenus avec le MAPE confirment ces conclusions même si on voit graphiquement que la transformation Box-Cox diminue le MAPE moyen sur le modèle linéaire.
Étant donné qu'il n'y a pas de différence significative entre les différentes transformations de la variable, nous n'allons pas faire de transformations de variables pour comparer les autres modèles. Nous avons les résultats dans la figure 6 ci-dessous.

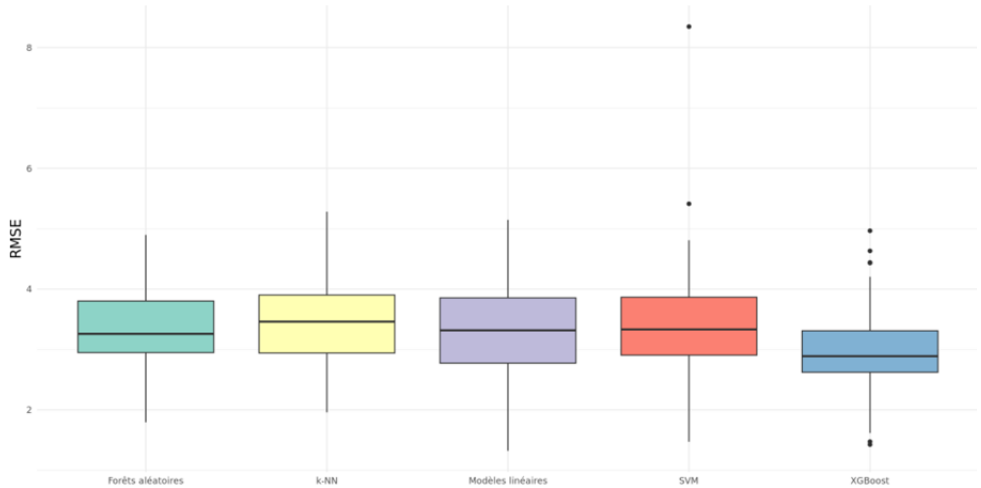


FIGURE 6 – Comparaison du RMSE entre différents algorithmes pour la prédiction de l'albumine.

Il n'y a pas de différences significatives entre les modèles, mais le modèle XGBoost présente la moyenne de RMSE la plus faible et une variabilité réduite. De plus, son coût computationnel

est raisonnable, ce qui en fait une option à conserver. Contrairement au modèle linéaire, qui a tendance à extrapoler lorsqu'il rencontre des données inédites, le modèle XGBoost ne génère pas de nouvelles données, car basés sur un arbre de décisions, ils ne vont pas continuer la tendance au-delà de la plage des données d'entraînement.

5.1.2 Méthode avec imputation

Dans cette partie, nous allons étudier plus en détail la méthode d'imputation et tout d'abord en comparant les méthodes d'imputation. D'un côté nous avons une imputation basique qui consiste à imputer par la moyenne, et toute nouvelle observation incomplète est complétée par la moyenne également. De l'autre, nous avons une imputation MICE et les nouvelles observations sont complétées par k-nn (pour k=5).

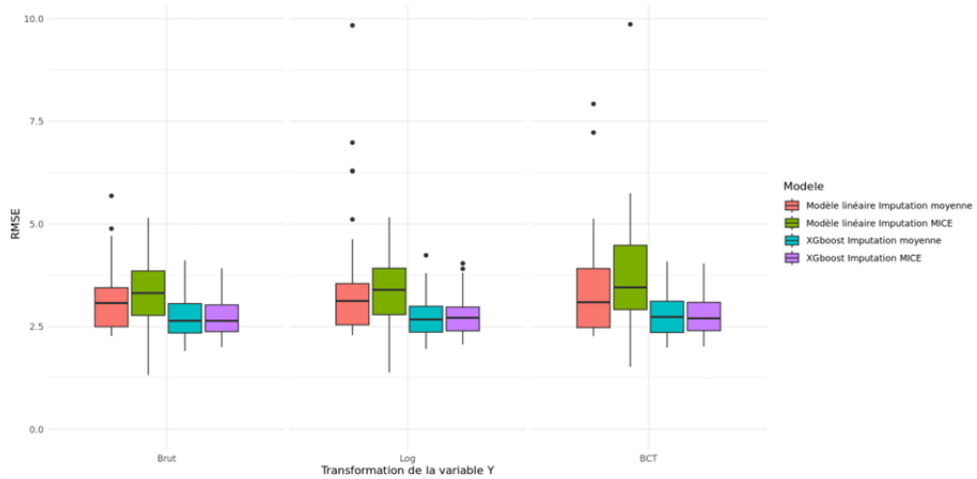


FIGURE 7 – Comparaison du RMSE entre différentes méthodes d'imputation, algorithmes de prédiction et transformation de la variable Y.

Les résultats de la figure 7 montrent qu'il n'y a pas de différence significative entre l'imputation par la moyenne et l'imputation par la méthode MICE pour chaque modèle. Cependant, l'imputation par la moyenne est beaucoup plus simple à mettre en œuvre. Cela indique que, dans le cas présent, la méthode MICE n'apporte pas d'amélioration notable. De plus, la recherche du plus proche voisin pour chaque valeur manquante peut être extrêmement longue, surtout pour des ensembles de données volumineux. Pour surmonter les limitations de temps de calcul, il aurait été nécessaire d'utiliser des serveurs plus puissants. Cependant, en raison de la nature sensible et confidentielle des données médicales, il n'est pas toujours possible de transférer ces données telles quelles. Pour contourner le problème de la confidentialité tout en utilisant la puissance de calcul externe, une solution consiste à créer des jeux de données synthétiques. Cet aspect a été abordé, notamment par l'utilisation de la méthode Avatar (Guillaudeux et al. (2023)) durant le stage mais n'a pas abouti.

Pour résumer nous allons faire un graphique qui compare : la prédiction si on remplaçait tout par la moyenne, la prédiction comme Bio Logbook le faisait, la prédiction selon la méthode des modèles complets gardés (XGBoost sans transformation de la variable Y) et la prédiction selon la méthode des modèles imputés gardés (imputation avec la moyenne).

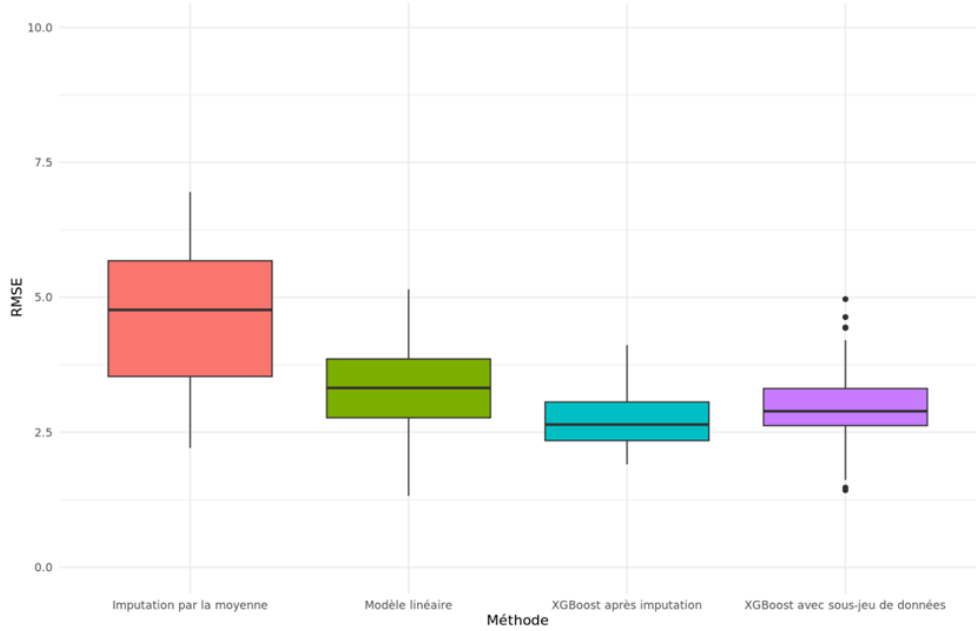


FIGURE 8 – Comparaison des RMSE entre différents méthodes et modèles.La première consiste à remplacer toutes les valeurs par la moyenne, la deuxième correspond à la méthode utilisée par Bio Logbook avec un modèle linéaire, la troisième corrrespond à la méthode des modèles complets gardés (XGBoost sans transformation de la variable Y), et la dernière correspond à la méthode des modèles imputés gardés, où l'imputation est effectuée en remplaçant par la moyenne.

On observe une différence significative entre l'imputation par la moyenne et les trois autres modèles, d'après les comparaisons par paires. Cependant, l'ANOVA n'indique pas de différences significatives entre les trois autres modèles. Néanmoins, on constate que le modèle XGBoost, après l'imputation des données, donne les meilleurs résultats.

5.2 Modèle de stacking

La méthode de stacking en *machine learning* est une technique d'apprentissage ensembliste qui vise à améliorer la précision des modèles prédictifs en combinant plusieurs algorithmes. Elle se déroule en deux étapes principales. D'abord, plusieurs modèles de base (appelés base learners) sont entraînés indépendamment sur les mêmes données. Chaque modèle génère ses propres prédictions, qui sont ensuite utilisées comme nouvelles caractéristiques pour un second modèle, appelé méta-apprenant. Ce modèle méta est ensuite entraîné pour apprendre à combiner les prédictions des modèles de base de manière optimale, en tirant parti de leurs points forts tout en corrigeant leurs erreurs individuelles. Cette approche permet de créer un modèle global censé être plus robuste et performant que chacun des modèles pris séparément, c'est ce que Rahman et al. (2023) a montré dans la prédiction du risque de mortalité au Covid-19. Dans notre cas, nous avons utilisé les modèles mentionnés ci-dessus en tant que base learners et utilisé soit le Random Forest soit le modèle linéaire en tant que méta-apprenant, les modèles les plus simples étant privilégiés.

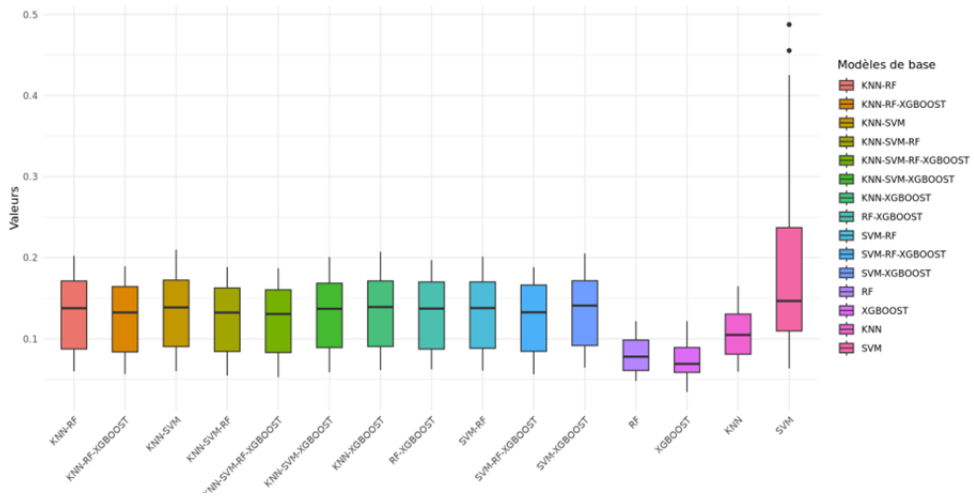


FIGURE 9 – Résultats du stacking (base learners : k-NN, Random Forest, SVM, XGBoost et meta-learners : Random Forest)

Le stacking n'a pas produit les résultats escomptés comme on le voit sur la figure 9, probablement en raison du surajustement des modèles de base, qui capturent souvent des relations similaires dans les données. Cette absence de diversité parmi les modèles de base a limité l'efficacité du modèle de stacking, ne permettant pas d'améliorer les performances globales. De plus, le stacking s'avère très gourmand en ressources computationnelles, ce qui complique son utilisation, notamment pour des ensembles de données volumineux. Compte tenu de ces limitations, il est préférable de se concentrer sur des modèles plus simples et mieux adaptés, qui offrent une performance fiable tout en étant plus efficaces en termes de calcul.

5.3 Exemple sur la troponine

Pour évaluer la généralisation de notre approche appliquée à l'albumine, nous allons la tester sur un autre paramètre, comme la troponine. La troponine est une protéine produite par le muscle cardiaque en temps normal. Cependant, en cas de lésion du muscle cardiaque, comme lors d'un infarctus du myocarde de type I, la troponine est libérée dans le sang, ce qui en fait un biomarqueur essentiel pour le diagnostic. Contrairement à l'albumine, la distribution des valeurs se rapproche d'une distribution exponentielle (figure 10). La majorité des patients ayant une concentration sanguine proche de 0ng/L mais certains ont une valeur à plus de 1000ng/L.

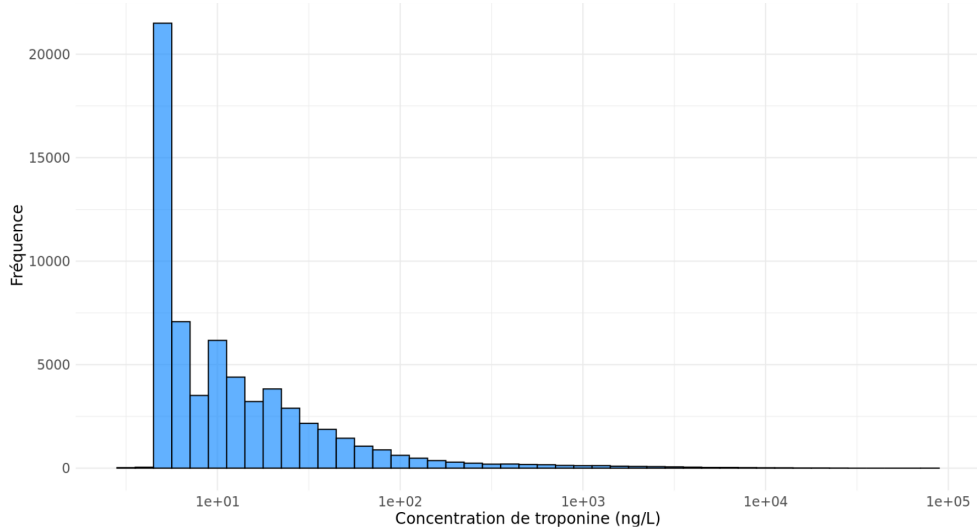


FIGURE 10 – Distribution de la troponine cardiaque

5.3.1 Régression pour prédire la troponine cardiaque

En appliquant les mêmes méthodes que précédemment à la prédiction de l'albumine, les résultats obtenus sont insatisfaisants. Par exemple, nous avons essayé d'utiliser un sous-ensemble de données complets avec le modèle XGBoost. Bien que ce modèle produise un graphique de prédiction, les valeurs élevées sont systématiquement sous-estimées comme nous le voyons dans la figure 11 qui comparent les valeurs prédites aux valeurs observées.

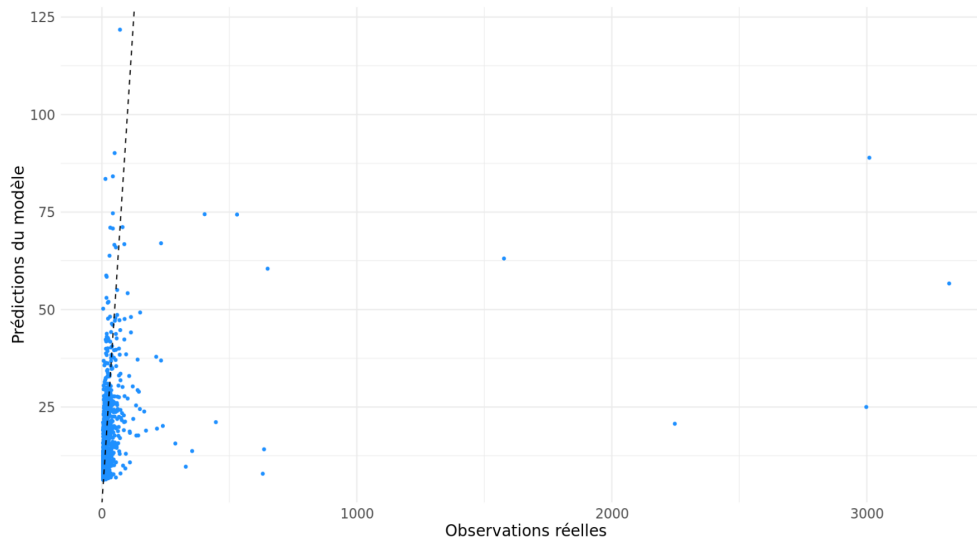


FIGURE 11 – Comparaison des observations et prédictions pour la troponine

5.3.2 Classification de la troponine

Dans ce cadre-là, il apparait plus facile de faire de la classification plutôt que de la régression, malgré le fait que la visualisation sur le logiciel ne soit pas comme imaginé puisqu'il ne s'agit pas de la prédiction d'une valeur. Puisque la haute autorité de santé considère une concentration anormale de troponine à partir de 14ng/L, on peut labeliser les patients ayant un taux inférieur à ce seuil comme « normal » et si ce n'est pas le cas comme « anormal ». L'important étant que l'algorithme soit capable de détecter un taux anormal. Nous avons tracé des courbes ROC (figure

12) pour quelques modèles, le but étant de voir si c'est réalisable avec des algorithmes basiques.

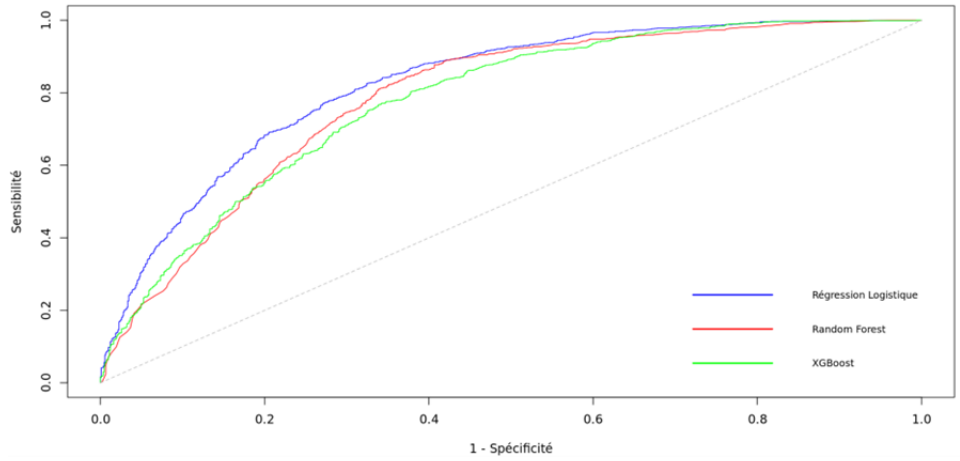


FIGURE 12 – Courbe ROC des modèles

Les courbes ROC montre qu'on peut classifier avec une assez bonne qualité, et notamment la régression logistique semble la meilleure. On peut voir que c'est une alternative plus réaliste pour des paramètres avec une distribution particulière qu'un algorithme de régression.

6 Intervalle de prédictions

Le but final étant de donner une valeur de la prédiction sur un logiciel il faut un intervalle de prédiction pour pouvoir donner un ordre d'idée de l'exactitude de la valeur. Pour donner une mesure de la fiabilité de la valeur prédite, il est nécessaire de fournir un intervalle de prédiction. Contrairement à un intervalle de confiance, qui donne un intervalle autour d'un paramètre comme la moyenne, l'intervalle de prédiction vise à prédire où une nouvelle observation individuelle est susceptible de se situer. C'est pourquoi l'intervalle de prédiction est généralement plus large que l'intervalle de confiance, car il doit tenir compte non seulement de l'incertitude autour de la moyenne mais aussi de la variabilité de l'erreur. Dans le cas du modèle linéaire utilisé par Bio Logbook, l'intervalle de prédiction peut être calculé à l'aide d'une formule de la forme :

L'intervalle de confiance pour une estimation Y_i est donné par :

$$Y_i \pm z_{1-\frac{\alpha}{2}} \times \sigma$$

où σ est l'écart-type de l'erreur et $z_{1-\frac{\alpha}{2}}$ est la valeur critique correspondant au niveau de confiance souhaité. L'écart-type de l'erreur σ peut être calculé à partir de la somme des carrés

des erreurs (SSE) comme suit :

$$\sigma^2 = \frac{\text{SSE}}{n-2} = \frac{1}{n-2}\sum_{i=1}^{n}(t_i - y_i)^2$$

où *SSE* est la somme des carrés des erreurs, n est le nombre total d'observations et k est le nombre de paramètres estimés dans le modèle (Shrestha et al. (2006)).

Cependant, cette méthode suppose que les données et les erreurs de prédiction suivent une distribution gaussienne avec une moyenne nulle et un écart-type , ce qui n'est pas le cas de nos données.

Dans la littérature, plusieurs autres méthodes ont été développées pour construire des intervalles de prédiction théoriques (IP) basés sur l'hypothèse mentionnée concernant les erreurs de prévision.

La méthode delta, proposée par Hwang et al. (1997), utilise et linéarise un modèle de réseau de neurones (NN) autour d'un ensemble de paramètres obtenus par minimisation de la somme des carrés des erreurs. Ensuite, la théorie asymptotique est appliquée au modèle linéarisé pour construire les IP. Cependant, cette méthode suppose que le bruit est homogène et normalement distribué, ce qui n'est pas toujours le cas dans les situations réelles. De plus, cette méthode est coûteuse en termes de calcul.

La technique bayésienne (Kothari et al. (1993), MacKay (1992)) a également été développée pour la construction d'IP basés sur les NN. Une fonction de coût régularisée est appliquée pour entraîner le modèle NN, permettant une meilleure généralisation que d'autres réseaux. Comme pour la technique delta, cette méthode est exigeante en termes de calcul car elle nécessite le calcul de la matrice hessienne pour la construction des IP.

La méthode bootstrap (Heskes (1996)) est l'une des techniques les plus fréquemment utilisées dans la littérature pour la construction d'intervalles de confiance et d'IP. La méthode bootstrap peut également être divisée en bootstrap paramétrique et non paramétrique (Efron et al. (1994)). Le bootstrap paramétrique suppose que les données originales sont normalement

distribuées. L'avantage de la méthode bootstrap non paramétrique est sa simplicité et sa facilité de mise en œuvre. En raison du fait que nous puissions nous affranchir de la distribution normale des données et de la relative simplicité à mettre en place, elle a été choisie dans ce travail pour calculer les intervalles de prédictions (méthode décrite dans Li et al. (2018)) et les comparer à la méthode actuellement utilisée par Bio Logbook.

Pour commencer, on peut diviser la variance totale par la variance du modèle et du bruit et nous obtenons ceci :

$$\sigma_i^2 = \sigma_{\hat{y}_i}^2 + \sigma_{\varepsilon_i}^2$$

Le processus de sélection de B échantillons dans la première étape est appelé le processus Bootstrap, comme illustré dans la Fig. 2. En remplaçant les quantiles de l'approximation de *Student* $t_{1-\alpha/2}$ par les quantiles issus de la distribution *bootstrap* du test de *Student* $t_{df}^{1-\alpha/2}$ (Efron et Tibshirani, 1993), l'intervalle de prédiction est construit comme suit :

$$\hat{y}_i \pm t_{df}^{1-\alpha/2} \sqrt{\frac{\sigma_{\hat{y}_i}^2 + \sigma_{\varepsilon_i}^2}{n}}$$

$t_{df}^{1-\alpha/2}$ est le quantile $1-\alpha/2$ de la fonction de distribution t avec des degrés de liberté df. Généralement, df est égal au nombre d'échantillons bootstrap. La méthode détaillée est décrite comme suit.

La variance du modèle $\sigma_{\hat{y}_i}^2$ et la moyenne de la régression sont estimées dans la première étape. La partie bruit $\sigma_{\varepsilon_i}^2$ est ensuite estimée dans l'étape suivante.

La première partie de la méthode est donc d'obtenir une estimation de $\hat{y}_i^*$ et de σ_{yi}. Cela peut se faire justement grâce à la méthode Bootstrap.

Pour estimer la variance et la moyenne de la régression d'un modèle, on suit les étapes suivantes :

1. **Génération d'échantillons bootstrap** : Créer B échantillons *bootstrap* en sélectionnant n données aléatoires avec remise à partir du jeu de données d'entraînement D1. La taille n peut être égale ou inférieure à celle de D1 pour réduire le calcul si D1 est trop grand.
2. **Modèles de prédiction** : Développer un modèle de prédiction pour chaque échantillon *bootstrap*.
3. **Estimation de la moyenne de la régression** : La moyenne estimée est la moyenne des prédictions de l'ensemble des B modèles de régression.
4. **Estimation de la variance** : La variance *bootstrap* des modèles de régression est calculée à partir de la dispersion des prédictions autour de la moyenne estimée.

La deuxième partie consiste à :

1. **Estimation du bruit de prévision** : On calcule la variance de l'erreur de prévision (σ_ε^2) en utilisant la formule $\sigma_\varepsilon^2 \simeq \mathbb{E}[(t-\hat{y})^2] - \sigma_f^2$ où t est la vraie valeur et $\hat{y}$ la valeur prédite. Cela revient à éliminer l'effet de la variance du modèle pour obtenir l'erreur de prévision.
2. **Calcul de l'erreur de prévision** : Avec un nouveau jeu de données D2, on calcule la moyenne de régression ($\hat{y}_i$), et la variance du modèle (σ_f^2) en utilisant la méthode *bootstrap*. Ensuite, on déduit l'erreur de prévision $t_i - \hat{y}_i$.
3. **Formation d'un nouveau jeu de données** : En utilisant la formule $\max((t_i-\hat{y}_i)^2 - \sigma_f^2, 0)$, on calcule la composante du bruit (r_i^2) pour chaque observation. Ce bruit devient la

nouvelle variable réponse dans le nouveau jeu de données $D_2^{'}$, où les entrées sont les mêmes que D2, mais les sorties sont remplacées par r_i^2.

4. **Modélisation de l'erreur** : Un modèle est ensuite développé pour prédire r_i^2 à partir du jeu de données $D_2^{'}$. L'erreur de prévision σ_ε^2 est alors estimée en appliquant une nouvelle fois la méthode *bootstrap*.

Pour évaluer la qualité de l'intervalle de prédiction, nous allons utiliser deux métriques adaptées :

— **Le NMPW (Normalized Mean Prediction Width)** mesure la largeur moyenne normalisée de l'intervalle de prédiction, qui se définit par :

$$MPIW = \frac{1}{n}\sum_{i=1}^{n}(U(X_i) - L(X_i)) \quad \text{et} \quad NMPW = \frac{MPIW}{R}$$

où $U(X_i)$ est la borne supérieure, $L(X_i)$ est la borne inférieure de y et R est la valeur maximale du paramètre moins la valeur minimale.

— **Le PICP (Prediction Interval Coverage Probability)**, quant à lui, évalue la proportion de valeurs observées qui tombent à l'intérieur de l'intervalle de prédiction. Idéalement, le PICP doit être le plus proche possible de 1, ce qui indique que l'intervalle capture correctement la majorité des observations.

$$PICP = \frac{1}{n}\sum_{i=1}^{n} c_i$$

où $c_i = 1$ si $y_i \in [L(X_i); U(X_i)]$, 0 sinon.

L'objectif est donc de minimiser la largeur de l'intervalle tout en maximisant la couverture des valeurs observées.
Ici nous avons testé 3 méthodes différentes pour évaluer la qualité des intervalles de prédictions. La première méthode correspond au modèle issu de la méthode linéaire, tandis que les deux autres méthodes correspondent au bootstrap non paramétrique mais avec deux méthodes différentes pour calculer le modèle de l'erreur : le XGBoost et les réseaux de neurones. Dix jeux de données différents ont été choisis pour essayer de dégager une tendance. Les résultats selon les deux métriques décrites plus haut sous illustrées ci-dessous.

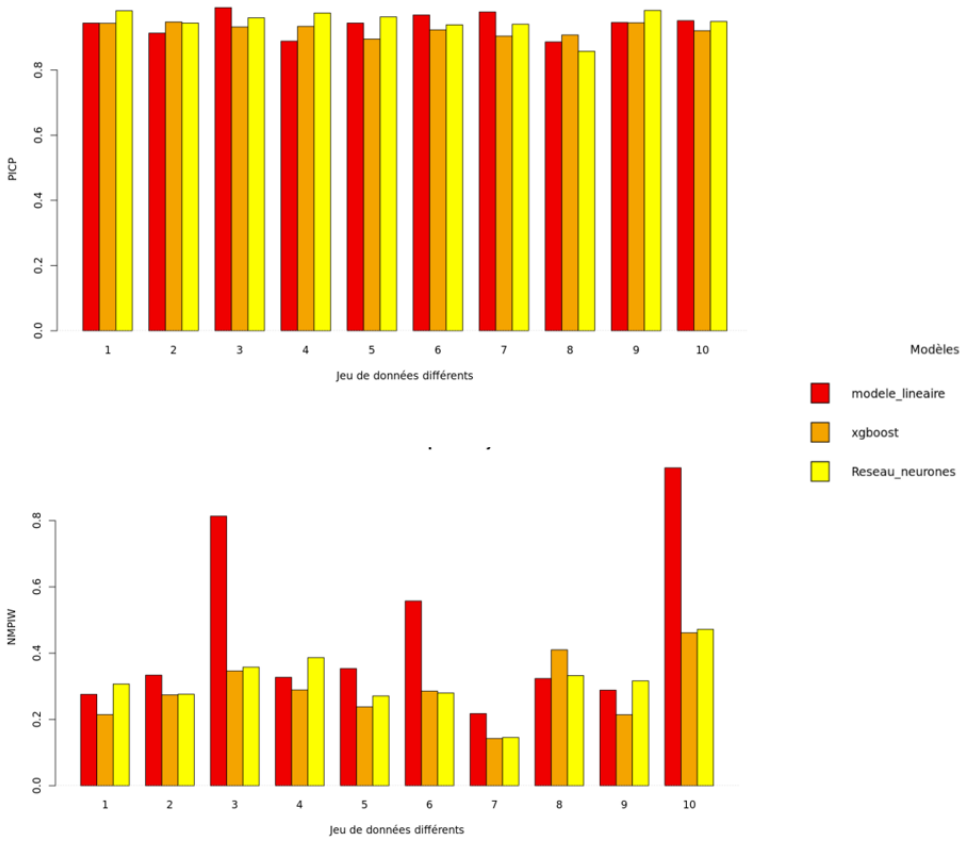


FIGURE 13 – Résultats de PICP (en haut) et NMPIW (en bas) pour dix jeux de données différents.

Dans la figure 13 nous pouvons voir les résultats : pour le PICP, les variations semblent minimes entre les différentes méthodes mais pour le NMPIW les résultats sont plus hétérogènes. Pour le jeu de données 3, 6 et 10 par exemple, la méthode utilisée par Bio Logbook aboutit à des intervalles beaucoup plus larges que les autres méthodes mais ce n'est pas toujours le cas. Il ne semble pas avoir une méthode meilleure qu'une autre car les résultats varient beaucoup selon le jeu de données. Il faudrait effectuer ces tests sur beaucoup plus de jeux de données, mais il faut aussi prendre en compte la très coûteuse demande computationnelle de la méthode de Bootstrap. On peut conclure que la méthode n'est pas améliorée.

Il existe d'autres approches non théoriques proposées récemment. Une méthode appelée Estimation des Bornes Inférieure et Supérieure (LUBE) (Khosravi et al. (2011)) est proposée sans besoin d'hypothèses sur les erreurs de prévision. Les bornes inférieure et supérieure de l'intervalle de prédiction sont directement obtenues via un modèle de réseaux de neurones. La fonction de coût de ce modèle est remplacée par une fonction objective basée sur les IP proposés, avec une méthode de recuit simulé (SA) pour l'entraînement. Cette méthode est censée surpasser la méthode Delta, la méthode bayésienne et la méthode bootstrap dans différents aspects et elle représente une possibilité à étudier.

7 Discussions

7.1 Gestion des Données Manquantes

L'absence de certaines variables est un problème récurrent dans les données cliniques, et dans notre cas en raison de la nature rétrospective de l'étude. Les données utilisées n'ont pas été initialement collectées pour le développement d'un modèle prédictif, ce qui complique l'analyse en présence de jeux de données incomplets. Cette limitation structurelle impose des défis, notamment en introduisant des biais potentiels.
Lorsqu'on crée des sous-ensembles complets en ne conservant que les patients pour lesquels toutes les variables sont mesurées, on risque d'introduire un biais de sélection. Les patients inclus dans ces sous-ensembles ont souvent reçu des prescriptions similaires, reflétant des soupçons de maladies similaires de la part des cliniciens. Cela signifie que ces sous-ensembles ne représentent pas toute la population de patients, mais plutôt un groupe spécifique influencé par des décisions médicales comparables. Ce biais peut limiter la capacité du modèle prédictif à généraliser aux diverses situations cliniques.

7.2 Imputation des Données Manquantes

L'imputation des données manquantes repose sur l'hypothèse que ces données sont manquantes de manière aléatoire (MAR). Toutefois, cette hypothèse est rarement vérifiée en pratique clinique. Les médecins n'ordonnent des tests que lorsqu'ils anticipent certains résultats, ce qui crée un biais systématique dans les données manquantes en fonction des jugements cliniques antérieurs. Par exemple, l'absence d'un test particulier dans les données peut refléter une absence de suspicion clinique pour une maladie donnée, rendant cette absence non aléatoire. De plus, comme indiqué par NIASS et al. (2015), lorsque le taux de données manquantes est élevé (au-delà de 5%), les techniques d'imputation comme MICE deviennent moins efficaces. Dans notre cas, avec environ 20% de données manquantes, cette méthode ne montre pas une efficacité supérieure à une imputation par la moyenne.

7.3 Exploration de Méthodes Alternatives

Il est important de considérer d'autres méthodes plus sophistiquées, étant donné que les modèles que nous avons utilisés sont relativement basiques mais bien éprouvés. Des techniques avancées, comme les réseaux de neurones, pourraient offrir des performances améliorées, bien que leur application en tant que dispositifs médicaux de prédiction nécessiterait une validation, notamment en ce qui concerne la fixation des poids.

7.4 Perspectives pour les Futures Recherches

Il serait également bénéfique de tester ces approches sur d'autres variables cliniques. Par exemple, la ferritine a été étudiée dans la littérature (Luo et al. (2016)), la prédiction de diverses maladies comme le Covid-19 (Gladding (2021)), et dans cette étude, nous avons exploré l'albumine et la troponine. Cependant, davantage de recherches sont nécessaires pour généraliser ces résultats à un ensemble plus large de variables cliniques.

8 Conclusions

Cette étude visait à développer une approche pour prédire des paramètres non mesurés à partir de données cliniques disponibles. Les résultats suggèrent que, pour des variables comme l'albumine, le modèle XGBoost offre de meilleures performances que les modèles linéaires. De ce fait, la transformation de la variable Y par la méthode logarithmique ou Box-Cox ne montre pas d'avantage significatifs. De plus, nous avons montré qu'il existait deux méthodes pour permettre la prédiction tout en s'adaptant à n'importe quelle observation en entrée sans que l'une ne montre un réel avantage significatif. En revanche, pour d'autres variables comme la troponine, la régression semble moins adaptée, et une approche de classification pourrait être plus pertinente mais cela nécessiterait de revoir la visualisation des résultats dans le logiciel. Bien que nous n'ayons pas encore identifié de méthode surpassant clairement celles existantes pour les intervalles de prédiction, des recherches supplémentaires sont nécessaires pour envisager un déploiement à plus grande échelle, notamment dans des logiciels de prédiction clinique.

Bibliographie

BOUTIN, R. ; ROLLAND, J. ; CODET, M. ; BÉZIER, C. ; MAES, N. ; KOLH, P. ; EQUINET, L. ; THYS, M. ; MOUTSCHEN, M. ; LAMY, P.-J. ; ALBERT, A., 2024. Use of hospital big data to optimize and personalize laboratory test interpretation with an application. *Clinica Chimica Acta*. T. 561, p. 119763. Disp. à l'adr. DOI : 10.1016/j.cca.2024.119763.

BOX, G. E. P. ; COX, D. R., 1964. An analysis of transformations. *Journal of the Royal Statistical Society, Series B*. T. 26, p. 211-252.

BREIMAN, L, 2001. Random forests. *Machine Learning*. T. 45, n° 1, p. 5-32.

EFRON, B ; TIBSHIRANI, R. J., 1994. *An Introduction to the Bootstrap*. 1st. Chapman et Hall/CRC. Disp. à l'adr. DOI : 10.1201/9780429246593.

GLADDING Ayar Z., Smith K., 2021. A Machine Learning Program to Identify COVID-19 and Other Diseases from Hematology Data. *Future Science OA*. T. 7, n° 7, FSO733. Disp. à l'adr. DOI : 10.2144/fsoa-2020-0207.

GUILLAUDEUX, M. ; ROUSSEAU, O. ; PETOT, J. ; AL., et, 2023. Patient-centric synthetic data generation, no reason to risk re-identification in biomedical data analysis. *npj Digital Medicine*. T. 6, p. 37. Disp. à l'adr. DOI : 10.1038/s41746-023-00771-5.

HESKES, T, 1996. Practical Confidence and Prediction Intervals. In : MOZER, M.C. ; JORDAN, M. ; PETSCHE, T. (éd.). *Advances in Neural Information Processing Systems*. MIT Press. T. 9. Aussi disponible à l'adresse : https://proceedings.neurips.cc/paper_files/paper/1996/file/7940ab47468396569a906f75ff3f20ef-Paper.pdf.

HWANG, J. T. G. ; DING, A. A., 1997. Prediction Intervals for Artificial Neural Networks. *Journal of the American Statistical Association*. T. 92, n° 438, p. 748-757. Disp. à l'adr. DOI : 10.1080/01621459.1997.10474027.

KHOSRAVI, A ; NAHAVANDI, S ; CREIGHTON, D ; ATIYA, A. F., 2011. Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals. *IEEE Transactions on Neural Networks*. T. 22, n° 3, p. 337-346. Disp. à l'adr. DOI : 10.1109/TNN.2010.2096824.

KOTHARI, S. C. ; OH, Heekuck, 1993. Neural Networks for Pattern Recognition. In : YOVITS, Marshall C. (éd.). *Advances in Computers*. Elsevier. T. 37, p. 119-166. ISBN 9780120121373. ISSN 0065-2458. Disp. à l'adr. DOI : 10.1016/S0065-2458(08)60404-0. Abstract : Publisher Summary.

LI, K. ; WANG, R. ; LEI, H. ; ZHANG, T. ; LIU, Y. ; ZHENG, X., 2018. Interval prediction of solar power using an Improved Bootstrap method. *Solar Energy*. T. 159, p. 97-112. ISSN 0038-092X. Disp. à l'adr. DOI : https://doi.org/10.1016/j.solener.2017.10.051.

LUO, Y ; SZOLOVITS, P ; DIGHE, AS ; BARON, JM, 2016. Using Machine Learning to Predict Laboratory Test Results. *American Journal of Clinical Pathology*. T. 145, n° 6, p. 778-788. Disp. à l'adr. DOI : 10.1093/ajcp/aqw064. Epub 2016 Jun 21.

MACKAY, D. J. C., 1992. The Evidence Framework Applied to Classification Networks. *Neural Computation*. T. 4, n° 5, p. 720-736. ISSN 0899-7667. Disp. à l'adr. DOI : 10.1162/neco.1992.4.5.720.

NIASS, O. ; DIONGUE, A. ; TOURÉ, Aissatou, 2015. Analysis of missing data in sereo-epidemiologic studies. *African Journal of Applied Statistics*. T. 2, n° 1, p. 29-37. Disp. à l'adr. DOI : 10.16929/ajas/2015.1.29.73.

RAHMAN, T. ; CHOWDHURY, M. E. H. ; KHANDAKAR, A. ; AL., et, 2023. BIO-CXRNET : a robust multimodal stacking machine learning technique for mortality risk prediction of COVID-19 patients using chest X-ray images and clinical data. *Neural Computing and Applications*. T. 35, p. 17461-17483. Disp. à l'adr. DOI : 10.1007/s00521-023-08606-w.

SHRESTHA, D. L. ; SOLOMATINE, D. P., 2006. Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks*. T. 19, n° 2, p. 225-235. Disp. à l'adr. DOI : 10.1016/j.neunet.2006.01.012. Epub 2006 Mar 10.

TAMUNE, H ; UKITA, J ; HAMAMOTO, Y ; TANAKA, H ; NARUSHIMA, K ; YAMAMOTO, N, 2020. Efficient Prediction of Vitamin B Deficiencies via Machine-Learning Using Routine Blood Test Results in Patients With Intense Psychiatric Episode. *Frontiers in Psychiatry*. T. 10, p. 1029. Disp. à l'adr. DOI : 10.3389/fpsyt.2019.01029. Published 2020 Feb 20.

WRIGHT, M. N. ; WAGER, Stefan ; PROBST, Philipp, 2023. *ranger : A Fast Implementation of Random Forests*. Version 0.16.0. Aussi disponible à l'adresse : http://imbs-hl.github.io/ranger/. CRAN Package.

Fiche résumé = quatrième de couverture du mémoire*

L'INSTITUT agro Rennes Angers	Diplôme : Master Spécialité : Science des données pour la Biologie Spécialisation / option : Enseignant référent : Laetitia CHAPEL

Auteur(s) : Océane FRAYSSINET Date de naissance* : 11/09/1998	Organisme d'accueil : Bio Logbook Adresse : 1 rue Julien Videment, 44000 Nantes Maître de stage : Jakez ROLLAND
Nb pages : 25 Annexe(s) : 0	
Année de soutenance : 2024	

Titre français : Estimation de variables en biologie médicale par différentes méthodes statistiques.

Titre anglais : Estimation of Variables in Medical Biology Using Different Statistical Methods.

Résumé (1600 caractères maximum) :

Dans les examens de laboratoire, certains paramètres pertinents pour le diagnostic clinique ne sont pas systématiquement mesurés, ce qui peut limiter la valeur diagnostique des tests. La prédiction des paramètres non mesurés à partir de données de laboratoire de routine pourrait être extrêmement utile pour améliorer le diagnostic de laboratoire. Dans cette étude, nous avons exploré divers algorithmes pour prédire des paramètres non mesurés, en testant différentes approches pour s'adapter à toutes les observations possibles (en créant des jeux de données avec les mêmes paramètres ou en imputant les valeurs des paramètres manquants). Nous avons démontré qu'il est possible de prédire efficacement l'albumine et qu'un modèle non linéaire est plus performant qu'un modèle linéaire. Pour la troponine, une approche de classification est plus appropriée qu'une approche de régression. Les intervalles de prédiction paramétriques et non paramétriques ont été comparés, sans qu'aucune méthode ne soit clairement supérieure. Ces résultats suggèrent des pistes pour le développement de nouveaux outils de soutien à la décision clinique basés sur l'intégration et l'interprétation des données de laboratoire.

Abstract (1600 caractères maximum) :

In laboratory examinations, certain parameters relevant to clinical diagnosis are not systematically measured, which can limit the diagnostic value of the tests. Predicting unmeasured parameters from routine laboratory data could be extremely useful for improving laboratory diagnostics. In this study, we explored various algorithms to predict unmeasured parameters, testing different approaches to accommodate all possible observations (by creating datasets with the same parameters or imputing missing parameter values). We demonstrated that it is possible to effectively predict albumin and that a non-linear model performs better than a linear model. For troponin, a classification approach is more appropriate than a regression approach. Parametric and non-parametric prediction intervals were compared, with no method proving to be clearly superior. These results suggest avenues for the development of new clinical decision support tools based on the integration and interpretation of laboratory data.

Mots-clés : données biologiques, données non mesurées, modèle de prédiction, modèle linéaire et non-linéaire, intervalle de prédiction

Key Words : biological data, unmeasured data, prediction model, linear and non-linear model, prediction interval

* *Elément qui permet d'enregistrer les notices auteurs dans le catalogue des bibliothèques universitaires*