Université de Montréal

Introduction à l'apprentissage automatique en pharmacométrie

Concepts et Applications

Par

Paul-Antoine Leboeuf

Axe Pharmacométrie & Pharmacothérapie, Faculté de Pharmacie

Mémoire présenté en vue de l'obtention du grade de maître ès sciences (M.Sc)

en Sciences Pharmaceutiques, option Pharmacologie

Décembre 2020

© Paul-Antoine Leboeuf, 2020

Université de Montréal

Axe Pharmacométrie & Pharmacothérapie, Faculté de Pharmacie

Ce mémoire intitulé

Introduction à l'apprentissage automatique en pharmacométrie

Concepts et Applications

Présenté par

Paul-Antoine Leboeuf

A été évalué(e) par un jury composé des personnes suivantes

Denis DeBlois
Président-rapporteur

Fahima Nekka
Directrice de recherche

Evelyne Lutton
Membre du jury

Résumé

L'apprentissage automatique propose des outils pour faire face aux problématiques d'aujourd'hui et de demain. Les récentes percées en sciences computationnelles et l'émergence du phénomène des mégadonnées ont permis à l'apprentissage automatique d'être mis à l'avant plan tant dans le monde académique que dans la société. Les récentes réalisations de l'apprentissage automatique dans le domaine du langage naturel, de la vision et en médecine parlent d'eux-mêmes. La liste des sciences et domaines qui bénéficient des techniques de l'apprentissage automatique est longue.

Cependant, les tentatives de coopération avec la pharmacométrie et les sciences connexes sont timides et peu nombreuses. L'objectif de ce projet de maitrise est d'explorer le potentiel de l'apprentissage automatique en sciences pharmaceutiques. Cela a été réalisé par l'application de techniques et des méthodes d'apprentissage automatique à des situations de pharmacologie clinique et de pharmacométrie. Le projet a été divisé en trois parties. La première partie propose un algorithme pour renforcer la fiabilité de l'étape de présélection des covariables d'un modèle de pharmacocinétique de population. Une forêt aléatoire et l'XGBoost ont été utilisés pour soutenir la présélection des covariables. Les indicateurs d'importance relative des variables pour la forêt aléatoire et pour l'XGBoost ont bien identifié l'importance de toutes les covariables qui avaient un effet sur les différents paramètres du modèle PK de référence. La seconde partie confirme qu'il est possible d'estimer des concentrations plasmatiques avec des méthodes différentes de celles actuellement utilisés en pharmacocinétique. Les mêmes algorithmes ont été sélectionnés et leur ajustement pour la tâche était appréciable. La troisième partie confirme la possibilité de faire usage des méthodes d'apprentissage automatique pour la prédiction de relations complexes et typiques à la pharmacologie clinique. Encore une fois, la forêt aléatoire et l'XGBoost ont donné lieu à un ajustement appréciable.

Mots-clés : Apprentissage automatique, Méthodes ensemblistes, Pharmacométrie, Sciences pharmaceutiques, Forêts aléatoires, eXtreme Gradient Boosting

Abstract

Machine learning offers tools to deal with current problematics. Recent breakthroughs in computational sciences and the emergence of the big data phenomenon have brought machine learning to the forefront in both academia and society. The recent achievements of machine learning in natural language, computational vision and medicine speak for themselves. The list of sciences and fields that benefit from machine learning techniques is long.

However, attempts to cooperate with pharmacometrics and related sciences are timid and limited. The aim of this Master thesis is to explore the potential of machine learning in pharmaceutical sciences. This has been done through the application of machine learning techniques and methods to situations of clinical pharmacology and pharmacometrics. The project was divided into three parts. The first part proposes an algorithm to enhance the reliability of the covariate pre-selection step of a population pharmacokinetic model. Random forest and XGBoost were used to support the screening of covariates. The indicators of the relative importance of the variables for the random forest and for XGBoost recognized the importance of all the covariates that influenced the various parameters of the PK model of reference. The second part exemplifies the estimation of plasma concentrations using machine learning methods. The same algorithms were selected and their fit for the task was appreciable. The third part confirms the possibility to apply machine learning methods in the prediction of complex relationships, as some typical clinical pharmacology relationships. Again, random forest and XGBoost got a nice adjustment.

Keywords: Machine learning, Ensemble Methods, Pharmacometrics, Pharmaceutical sciences, Random Forest, eXtreme Gradient Boosting

Table des matières

Liste des tableaux

Liste des figures

Liste des sigles et abréviations

ACoP : Conférence Américaine de pharmacométrie (American Conference of Pharmacometrics)

CART : Classification and Regression Trees

CV : Validation croisée (Cross Validation)

COV : Covariables

GBM : Gradient Boosting Machine

MAE : Erreur absolue moyenne

ML : Machine Learning, Apprentissage Automatique

ME : Erreur moyenne

PK : Pharmacocinétique

PopPK : Population pharmacokinetics

RMSE : Racine de l'erreur quadratique moyenne

R^2 : R carré

SRC : Somme résiduelle des carrés

XGBoost : eXtrem Gradient boosting

À Papa et Maman,

vous qui m'avez soutenu tout au long de cette aventure

Le premier mot est toujours le plus difficile à mettre sur papier,

mais une fois que c'est chose faite, moins que la moitié reste à faire.

Remerciements

Je commencerais par remercier ma directrice de mémoire Fahima Nekka pour son soutien, ses conseils et son écoute tout au long du projet. Je la remercie d'avoir pris le temps de lire et de corriger ce mémoire, de façon itérative et constructive, que même le confinement n'a pas altérée. Mon style scientifique en est sorti vraiment bonifié. Je suis aussi reconnaissant pour l'apport financier qui m'a été octroyé à travers la Chaire CRSNG-Industrie-Pfizer-inVentiv Health-Novartis en Pharmacometrie, détenue par Dr. Nekka.

J'aimerais prendre le temps de remercier Professeure Evelyne Lutton et Professeur Denis Deblois d'avoir accepté d'évaluer mon travail.

Merci à Steven et Guillaume qui ont été pour moi des références. Merci de m'avoir fourni conseils et guidance. Cassandre, Imad, Sara, Anaëlle et Florence, vous m'avez donné le goût d'aller travailler chaque jour. J'ai apprécié tant nos discussions à nature scientifique que celles plus frivoles. J'aurai récolté de bien beaux souvenirs au cours des 20 mois passés à vos côtés.

Mon aventure aurait été bien différente si je ne vous avais pas eu à mes côtés tous les jours. Je parle bien évidemment de vous, les gars : Jeffery, Abdullah et Augusto. Nos fous rires et votre présence me manquent énormément. J'ai particulièrement aimé vous avoir comme compagnons de bureau.

Je terminerai en remerciant du plus profond de mon cœur mes parents. Vos encouragements m'ont été essentiels pour l'achèvement de cet énorme projet. Votre support quotidien n'a pas passé inaperçu. Tout ça n'aura pas été possible sans vous. Je vous aime.

Chapitre 1 – Fondement de l'apprentissage statistique

Modélisation

Lorsque l'on souhaite concrétiser la connaissance qu'on a sur un phénomène, de vérifier sa véracité et de quantifier l'information, l'une des manières d'y arriver est de modéliser le système d'intérêt. La modélisation suppose une représentation simplifiée du système et de son fonctionnement. Deux objectifs motivent la modélisation. Le premier est de permettre la prédiction du comportement du système sous certaines conditions. Habituellement, on souhaite prédire le comportement futur du système pour être informé sur son devenir dans différentes conditions. Le second objectif est d'étudier le fonctionnement du système pour en tirer des conclusions, habituellement sur les facteurs qui l'affectent. On pourrait vouloir savoir quel facteur a le plus d'influence sur la réponse à un traitement, ou encore, savoir à quel effet s'attendre lorsqu'un facteur double en quantité.

Lorsque vient le temps de modéliser un phénomène, la question suivante peut surgir: Quelle méthode de modélisation choisir, quel modèle caractérisera le mieux le système?(1) Il existe un nombre considérable de méthodes proposées, chacune reposant sur une idée intéressante, mais aucune ne répondant complètement au défi de saisir toutes les infinies nuances qui composent le monde réel.(2)

Il existe différentes philosophies de modélisation. L'une d'entre elles propose de développer un modèle en se basant sur des connaissances préalablement acquises. Par exemple, si l'on veut modéliser la trajectoire d'un projectile, on bâtirait le modèle en utilisant les lois de la mécanique classique. Dans ce cas, on considère connaitre les mécanismes sous-jacents au système. On parle alors d'un modèle mécanistique. Ce type de modèles exprime généralement ces mécanismes sous la forme d'équations mathématiques. Ce type de modélisation a pour avantage d'être clairement formulé et facile d'interprétation, étant donné que l'analyse du système se fait dans le cadre défini par le mécanisme explicatif choisi. Seulement, il est légitime de se demander à quel point notre compréhension des lois complexes de la nature est adéquate. (1)

Une autre philosophie propose une approche complètement différente. Il s'agit d'une méthode basée sur l'expérience. On récolte les données reflétant nos observations et on modélise les données collectées. Si on poursuit avec l'exemple de la trajectoire d'un projectile, la force fournie au projectile, l'angle de projection et sa portée pourraient être rapportés comme observations et utilisés dans la modélisation. La méthode décrite ici fait référence à la modélisation dite empirique. Plusieurs modèles statistiques sont des représentants de cette forme de modélisation.(3)

Modélisation statistique

La modélisation statistique est un sous ensemble de la modélisation mathématique. Elle suppose une distribution de probabilité sous-jacente expliquant le comportement de la variable réponse. Une scission existe parmi les modèles statistiques. Il y a la modélisation classique et la modélisation algorithmique. En modélisation statistique, on représente le système par un schéma semblable à une boite noire qui représente la réalité, le monde naturel, qui est nourrie de variables d'entrée et qui produit une variable de sortie. Ce qui est dans la boite, la vraie relation, est hors de portée.

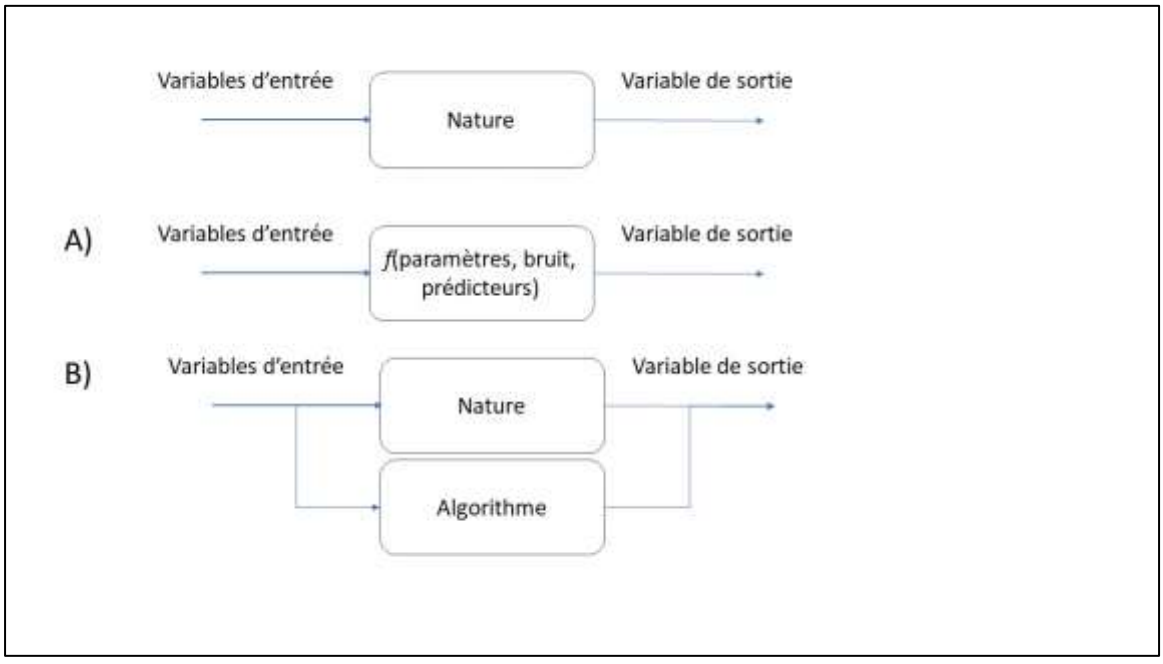

Figure 1. – Représentation des deux cultures en modélisation statistique.*(3)*

La modélisation statistique classique suppose que l'intérieur de la boite noire peut être représenté par une fonction. Elle propose une distribution pour la variable réponse qui peut être caractérisée via des paramètres. Ainsi, les données sont générées d'une pige indépendante à partir de la distribution de la variable réponse. Ce type de modèles a pour force de rendre accessible l'interprétation du phénomène étudié. La situation est schématisée dans l'exemple A de la figure ci-dessus.

La modélisation algorithmique cherche un algorithme qui a la capacité de prédire la réponse, et ce, sans chercher à expliquer et comprendre ce qu'il y a dans la boite noire représentant la réalité, le monde naturel. Ce type de situations est illustré dans l'exemple B de la figure ci-dessus. Ainsi, le modélisateur aspire à ce que son algorithme soit un approximateur universel de fonctions capables d'assimiler les relations et les interactions présentes dans les données. Évidemment, ce genre de modèles ne facilite pas la compréhension du système, mais concentre ses efforts sur la prédiction de la réponse. Cette philosophie s'est enracinée bien plus récemment dans la pratique des statistiques, où plusieurs techniques d'apprentissage statistiques et d'apprentissage automatique en font partie.

La décision à prendre lorsqu'on choisit le type de modélisation statistique à adopter reflète le parallèle entre les deux cultures de modélisation statistique. Soit la simplification est juste, soit-on n'essaye pas de comprendre l'intérieur de la « boite noire » de la réalité, et alors, on utilise un algorithme qui prédit directement la réponse sans tenter d'expliquer les relations complexes sous-jacentes.

Évidemment, l'objectif motivera le choix du type de modélisation. Il faut se demander s'il est souhaitable de privilégier l'interprétation du modèle ou de pouvoir prédire avec le plus de précision possible? Est-on dans un contexte d'inférence ou de prédiction? Heureusement, les modèles statistiques ne se retrouvent pas tous à ces extrêmes. Les concepts abordés ne se traduisent pas en catégories exclusives; certains types de modèles permettent d'atteindre un bon compromis.

Cependant, il faut garder en tête que l'objectif premier est d'avoir de l'information juste et précise et qu'il est contreproductif d'interpréter un modèle pas assez précis. Un modèle très

complexe et précis, qui ne peut pas être adéquatement interprété, est plus précieux qu'un simple modèle linéaire sans qualité prédictive mais que nous pouvons interpréter parfaitement. Si les deux modèles de types différents fournissent une information juste, alors le modèle le plus interprétable sera préféré. (3)

Modèles non-linéaire à effets mixtes

Les modèles non-linéaires à effets mixtes sont dits semi-mécanistiques. La partie mécanistique correspond à leur structure. Celle-ci représente la relation non-linéaire qui existe entre les paramètres du modèle et la variable réponse. En pharmacocinétique, cette structure est composée d'équations différentielles représentant les échanges entre les différents *compartiments* du corps. En plus de la structure mécanistique, il permet de modéliser une variable aléatoire, généralement une concentration plasmatique. (4)

La paramétrisation du modèle se fait de manière mixte comme son nom l'indique avec des effets fixes et des effets aléatoires. Les effets fixes représentent les variables contrôlées durant l'expérimentation comme la dose, la fréquence d'échantillonnage, les covariables collectées durant la durée de l'étude, telles que le poids, l'âge et le sexe ainsi que les paramètres pharmacocinétiques typiques de la population. Les effets aléatoires correspondent aux paramètres qui varient à chaque observation et c'est leur variance qu'on cherche à estimer.(4) Ce type de modèle est facilement interprétable et permet de simuler différents cas de figures. Il permet de répondre à plusieurs interrogations concernant la réponse d'un individu (ou d'un groupe spécifique d'individus) au traitement, advenant que son état de santé change. Plus particulièrement, il permet d'explorer les possibilités de traitement chez les groupes de patients spécifiques i.e. les enfants, les personnes âgées, les femmes enceintes, les obèses, les insuffisants rénaux, etc. Ce type de réponses est essentiel pour les compagnies pharmaceutiques développant un médicament, ou chez les cliniciens voulant proposer un soin individualisé à leurs patients.(5)

L’apprentissage automatique

En 1959, Arthur Samuel, un pionnier du domaine de l’apprentissage automatique, a fait l’énoncé suivant : « L’apprentissage automatique est le domaine d'étude qui fournit aux ordinateurs la capacité d'apprendre sans être explicitement programmés pour le faire ». On dit qu'un programme est en train d'apprendre lorsque sa performance pour une tâche, mesurée par un certain critère, s'améliore avec l'expérience.(2, 6, 7)

Pour mieux comprendre ce concept, il faut revisiter les techniques de programmation traditionnelles. Prenons l’exemple d’un programme dont l’objectif est d’indiquer si une personne possède une certaine maladie. La première étape serait d’observer le phénomène et d’identifier les caractéristiques exclusives aux victimes de cette maladie, comme le poids, l’âge et la présence de maladies concomitantes par exemple. En second lieu, il faudrait rédiger un programme de détection en lui indiquant de manière explicite les caractéristiques déterminantes au diagnostic de la maladie. Il est question ici de transmettre nos hypothèses sur la nature de la relation étudiée. Et finalement, il faudrait tester l’efficacité de l’algorithme lors d’une étude clinique avant de la mettre en application. Cela dit, il est probablement difficile d’identifier d’avance les caractéristiques pertinentes, de même qu’il est peu probable de parvenir à toutes les identifier. Cette situation pourrait se présenter lorsque des centaines de variables sont impliquées, comme lors de l’étude des effets de facteurs génétiques par exemple. En revanche, un algorithme d’apprentissage automatique détermine de manière autonome les caractéristiques prédictives à la présence de la maladie, sans avoir besoin qu’une hypothèse lui soit transmise. De manière générale, les algorithmes d’apprentissage automatique sont particulièrement attrayants pour les tâches jugées trop complexes pour les approches traditionnelles, où la solution serait inexistante, approximative ou pressentie sous-optimale. L’apprentissage machine est particulièrement apprécié pour ses applications en reconnaissance d’image, en traitement du langage naturel, en finance, en bio-informatique, en médecine, etc. (8)

L’apprentissage statistique

L’apprentissage automatique tire sa source des sciences computationnelles et de la théorie de l’apprentissage statistique.(7, 8) L’apprentissage automatique s’intéresse à trois types d’apprentissage en particulier : l’apprentissage supervisé, l’apprentissage non supervisé et l’apprentissage par renforcement. Les deux premières catégories se classifient dans le domaine de l’apprentissage statistique.(9), qui fait référence à un vaste ensemble d'outils permettant de « comprendre » les données. L’apprentissage supervisé consiste à apprendre d'un système où les variables indépendantes et dépendantes sont disponibles. L’apprentissage non supervisé se fait avec des ensembles de données pour lesquelles il n’y a pas de variable de type réponse (dépendantes). Néanmoins cela n’empêche pas d’apprendre sur la structure et les relations internes de l’ensemble de données.(9) Le tous est résumé dans la figure 2.

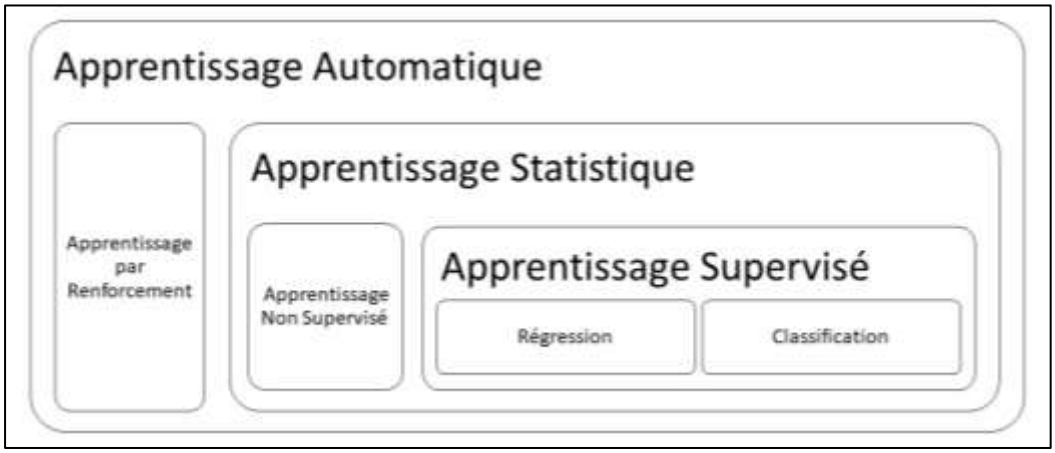

Figure 2. – Typologie de l'apprentissage automatique

Nomenclature

L’apprentissage statistique et l’apprentissage automatique partagent plusieurs concepts similaires mais utilisent des termes différents. Ainsi, le concept de *variables indépendantes* peut être interchangé avec les termes *prédicteurs*, *caractéristiques* ou *variables d’entrée*, voir figure 3. Ceci s’applique aussi au concept de *variable dépendante,* qui est synonyme de *variable de sortie* ou de *réponse*. Les variables de type qualitative peuvent être identifiées comme variables discrètes ou variables catégorielles. Les valeurs d’une variable de sortie de type qualitative sont appelées facteurs ou classes. Une tâche ayant pour objectif de prédire une réponse de type

catégorielle est référée comme un problème de classification. À l'inverse, lorsque l'objectif est de prédire une réponse de type continu, on parle d'un problème de régression. Nous porterons une attention particulière à la régression dans ce travail.

L'apprentissage supervisé

L'apprentissage supervisé suppose une situation où une certaine réponse, dénotée Y, est expliquée par un vecteur de prédicteurs, dénotés X = (x_1, x_2, x_3, …, x_p). La relation peut être représentée de manière générale avec l'équation suivante :

$$Y = f(X)\ (1)$$

L'estimation de la fonction *f* se fait à l'aide d'un algorithme capable d'implémenter et entrainer différents modèles *f*(X,a), où *a* représente les paramètres de la fonction *f*. L'objectif de l'apprentissage supervisé est d'estimer la fonction qui se rapproche le plus de la réponse théorique.(7) L'estimation de Y, Ŷ, est possible grâce à un ensemble de données d'entrainement, composé d'observations, fourni à l'algorithme. Cela permet d'identifier un modèle $\hat{f}(X)$ étant le plus proche possible de $f(X)$, ε étant la différence existante entre l'estimation et la vraie valeur.

$$\hat{Y} = \hat{f}(X) + \varepsilon\ (2)$$

Chaque observation de l'ensemble de données, composée d'un couple (X_i, Y_i), est supposée indépendamment et identiquement distribuée (i.i.d.) d'une population d'observations, signifiant que les données d'entrainement sont représentatives des données auxquelles le modèle sera confronté lorsqu'il sera mis en production.

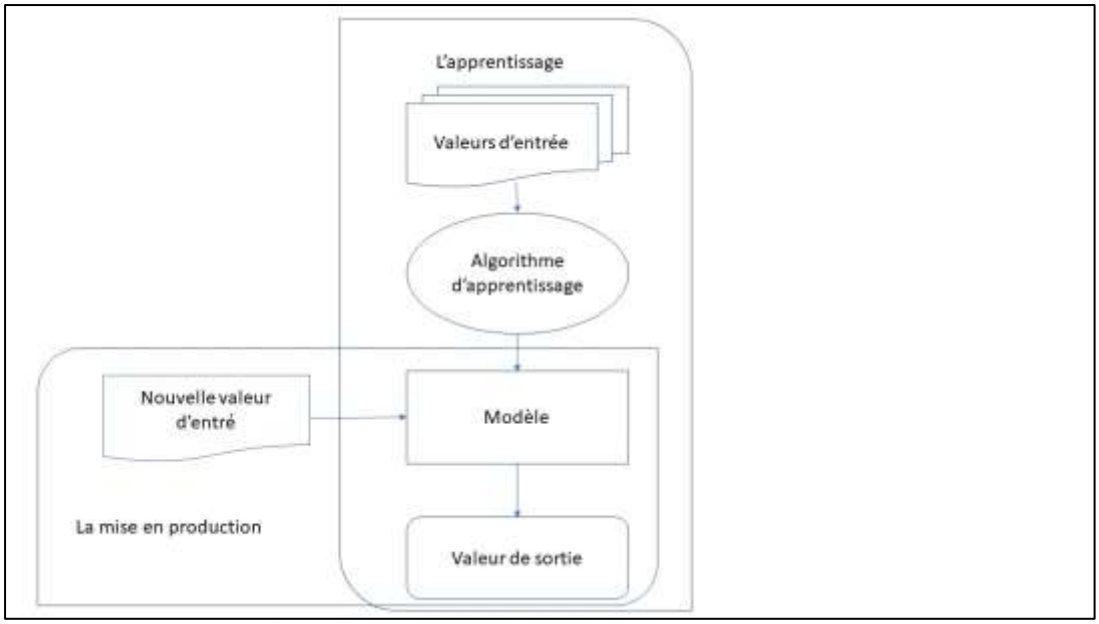

Figure 3. – Étapes de vie d'un modèle: apprentissage et production

Méthodes paramétriques et non paramétriques

L'estimation de la fonction *f* peut se faire avec deux types de méthodes d'apprentissage statistique. Les méthodes dites paramétriques, associées aux statistiques traditionnelles, et les méthodes dites non paramétriques, généralement mises à l'avant plan lorsqu'il est question d'apprentissage automatique.

La première catégorie se distingue par ses conditions nécessaires à l'estimation de la fonction *f*. Elle assume une forme particulière pour *f*. Par exemple, une régression linéaire est choisie pour l'estimation de la relation entre Y et X lorsque celle-ci est supposée de nature linéaire. L'avantage de ces techniques réside en la simplicité de l'estimation puisqu'il suffit d'estimer les quelques paramètres impliqués, qui sont faciles à interpréter. Cependant, ces méthodes peuvent fournir des prédictions biaisées si la forme de f choisie est trop loin de la réalité.

Les méthodes non paramétriques ne nécessitent pas d'hypothèse sur la forme de *f,* et par le fait même ont plus de chance de s'adapter à la vraie forme de la relation. C'est pourquoi ces méthodes sont considérées comme étant plus flexibles que les méthodes paramétriques. Cependant, elles requièrent de plus grandes quantités de données d'entrainement pour estimer avec succès les nombreux paramètres impliqués dans leurs fonctions. Malheureusement,

comme toutes les méthodes de ce genre, elles présentent des risques de surajustement (overfitting), qui pourrait se produire lorsque le modèle s'adapte non seulement à la relation étudiée mais aussi aux bruits présents dans l'ensemble de données. Cette situation génère également des prédictions non fiables. Cela dit, le surajustement peut être facile à identifier et peut généralement être contrôlé en ayant recours à certaines méthodes. Le concept est illustré dans la figure suivante. (9)

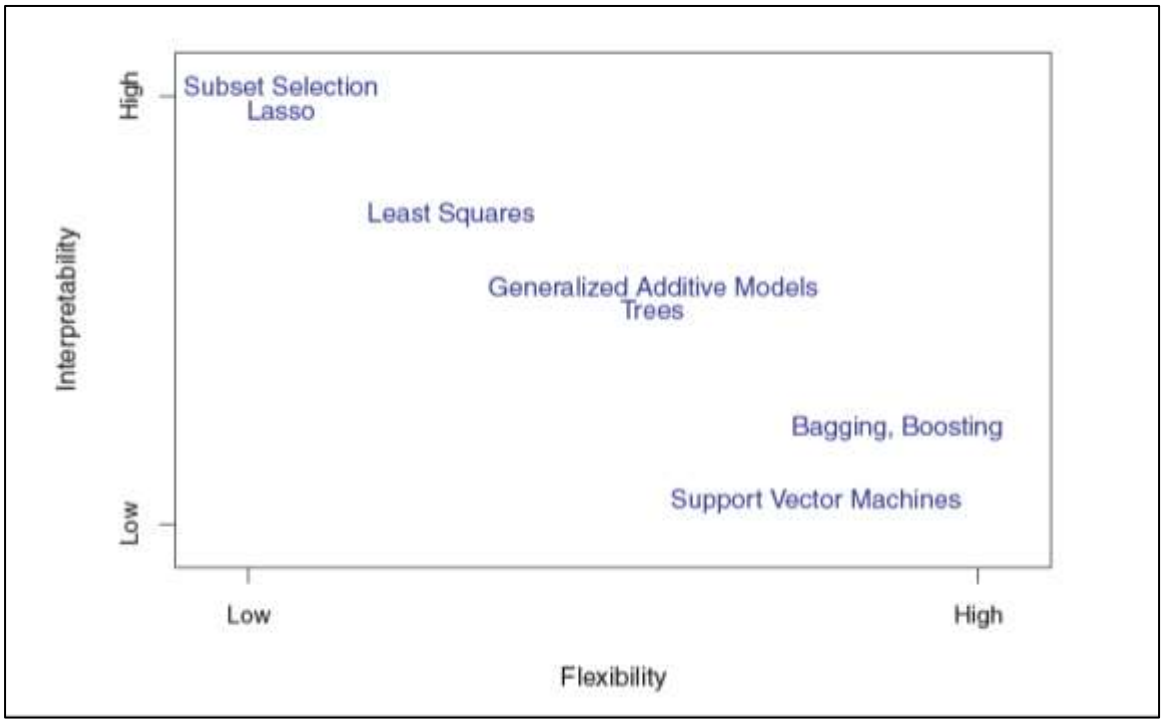

Figure 4. – Relation entre la flexibilité d'une méthode et son interprétabilité.

Équilibre entre la précision des prédictions et l'interprétabilité du modèle

L'utilisation des méthodes d'apprentissage statistique a deux objectifs : l'inférence et la prédiction. Une tâche est motivée par l'inférence lorsque l'intérêt est de comprendre la façon dont la variable dépendante est affectée par les variables indépendantes. Dans ce cas, la préférence est donnée à une méthode qui offre une bonne capacité d'interprétation. Lorsqu'il importe peu de savoir la forme exacte de *f* du moment où l'on obtient des estimations précises de la réponse, les méthodes offrant une bonne précision de prédiction sont préférées. Le plus souvent, une méthode atteint une bonne précision en raison de sa flexibilité. Un modèle est

flexible lorsqu'il comporte peu de restrictions dans la gamme des formes autorisées pour estimer la variable dépendante. La régression linéaire est un exemple de méthodes inflexibles car elle est limitée à des fonctions linéaires pour estimer la variable dépendante. D'autre part, le perceptron ou les réseaux de neurones sont très flexibles mais beaucoup plus difficiles à interpréter. En règle générale, l'interprétabilité d'une méthode diminue au fur et à mesure que la flexibilité augmente. Ce concept est appelé l'équilibre entre la précision des prédictions et l'interprétabilité du modèle.(9)

L'équilibre entre la variance et le biais

L'écart entre les valeurs prédites et les valeurs observées se quantifie par deux mesures : l'erreur réductible et l'erreur irréductible. L'erreur réductible est composée du biais et de la variance du modèle comme l'équation suivante.(9)

$$E\left(y_i - \hat{f}(x_i)\right)^2 = Var\left(\hat{f}(x_i)\right) + \left[Biais\left(\hat{f}(x_i)\right)\right]^2 + Var(\varepsilon) \quad (3)$$

Ainsi, pour réduire l'erreur total, il faut choisir une méthode ayant un faible biais et une faible variance. Un modèle a une grand variance lorsqu'il est sensible à une petite modification des données d'entrainement. Ceci se produit lorsque le modèle s'est trop adapté à ses données d'entrainement, et que lorsqu'il s'entraine avec des données différentes, le modèle estime une relation f largement différente. Il est question ici de surentrainement. Il y a deux solutions pour réduire la variance d'un modèle : restreindre sa flexibilité ou fournir plus de données d'entrainement afin d'augmenter la variété d'exemples rencontrés par le modèle dans son apprentissage. Le biais d'un modèle fait, quand à lui, référence à une mauvaise estimation de la relation f. L'ajout de données d'entrainement n'y changera rien. Si la relation étudiée est non-linéaire, peu importe la taille de l'ensemble d'entrainement, jamais un modèle linéaire saura bien capter la tendance. On parle ici de sous-entrainement. Il faut opter pour un estimateur ayant plus de flexibilité. Finalement, l'erreur totale ne peut descendre en dessous d'une certaine quantité représentée par l'erreur irréductibe ε. Comme m'entionné plus haut, augmenter la flexibilité d'une méthode réduit le biais, mais augmente par le fait même sa variance. La difficulté se trouve donc dans l'identification d'une méthode ayant un bon équilibre. Le phénomêne est illustré dans la figure suivante. (10)

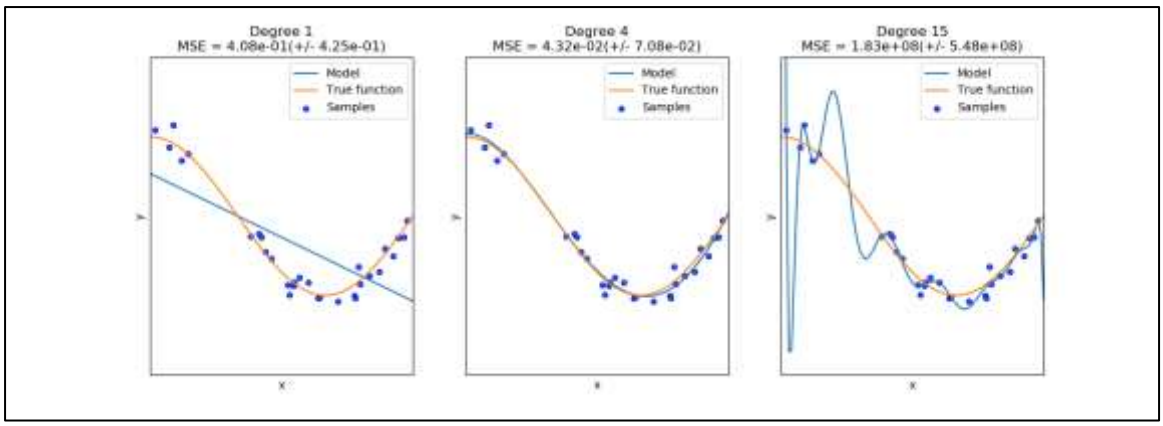

Figure 5. – De gauche à droite : sous-ajustement, situation équilibrée, surajustement

L'évaluation

Pour évaluer le pouvoir prédictif d'un modèle, plusieurs méthodes existent mais aucune ne peut s'appliquer à toutes les situations, avec des avantages et des inconvénients à chacune. Il est donc nécessaire de tester plusieurs méthodes pour en trouver une qui soit satisfaisante. Pour évaluer un modèle, la façon de faire est de mesurer si la prédiction trouvée pour Y_i est proche de la vraie valeur. L'objectif d'une prédiction est de produire des réponses pour des nouvelles observations. Évidemment, il n'est pas nécessaire d'évaluer la performance sur les données d'entrainement, puisque la valeur de Y est déjà connue. Idéalement, l'évaluation des modèles doit se faire sur un ensemble de données qui n'a pas été utilisé durant l'entrainement, qu'on désigne par un ensemble test.

L'évaluation de la performance prédictive des modèles de régression peut être réalisée par des métriques d'erreur telles que l'erreur moyenne (ME), la racine de l'erreur quadratique moyenne (RMSE), l'erreur absolue moyenne (MAE) et le R carré (r^2).

$$ME = \frac{1}{N}\sum_{i=1}^{n}(\hat{y}_i - y_i) \quad (4)$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (5)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (6)$$

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (7)$$

La mesure, ME, évalue le biais des prédictions du modèle. Le biais illustre la tendance à constamment sous-estimer ou surestimer les observations. Les autres mesures évaluent la précision du modèle. Une mesure de précision permet de calculer à quel degré les prédictions correspondent aux observations. Les directives en matière de pharmacométrie suggèrent de choisir la RMSE avant la MAE.(11, 12) La RMSE souligne les erreurs des grandes valeurs, comme on le souhaite dans une situation clinique, alors que la MAE exprime les erreurs sur une échelle linéaire.

Bonnes pratiques d'apprentissage

Prétraitement des données

Le prétraitement des données est un ensemble d'étapes préparant les données pour la phase d'apprentissage. Il s'agit de transformer des données brutes en un ensemble de données structurées, prêtes à être traitées par un modèle. La première étape consiste à effectuer une analyse exploratoire des données. Celle-ci peut être réalisée à l'aide d'outils de visualisation et de statistiques descriptives. Elle permet d'obtenir une bonne idée de ce que contient l'ensemble de données. Cette étape préparatoire comprend le traitement des valeurs manquantes et des valeurs aberrantes. Certaines méthodes, comme les arbres de décision, peuvent prendre en charge les valeurs manquantes, mais en général, il est préférable de les considérer dans l'analyse. Chaque analyse doit avoir un plan préétabli pour gérer les valeurs manquantes. Bien que le remplacement des valeurs manquantes par une mesure de tendance centrale ainsi que la suppression complète de l'observation soient des solutions fréquemment choisies, il n'y a pas de méthode universelle. Les données aberrantes sont, quant à elles, des données qui ne semblent pas correspondre à la tendance générale. Parfois, il peut être bon de supprimer les valeurs aberrantes car elles peuvent avoir glissé dans l'ensemble de données par

erreur. Les inclure dans l'analyse contribue à l'erreur irréductible. Parfois, il faut les conserver dans l'analyse parce qu'elles peuvent être le témoin de tendances inconnues ou imprévues dans les données. L'analyse exploratoire des données n'est pas spécifique au domaine de l'apprentissage automatique. Cela dit, la solution au problème n'est pas un sujet d'étude de l'apprentissage automatique. Les problèmes reliés aux valeurs manquantes et aux valeurs aberrantes se retrouvent dans tous les types d'analyses de données. Les stratégies d'échantillonnage limitées et les valeurs manquantes lors d'essais cliniques de phase III sont de bons exemples. D'une manière ou d'une autre, il faut avoir des raisons valables et basées sur la logique lorsqu'une décision est prise sur la façon de gérer ce type de données.

La deuxième étape du prétraitement des données est la mise à l'échelle des variables; elle sert à assurer la transformation des prédicteurs pour qu'ils soient sur une échelle commune. De nombreuses méthodes d'apprentissage automatique sont sensibles à la manière dont les variables leur sont présentées. Lorsqu'une méthode fonctionne avec des calculs de distance euclidienne, toutes les variables doivent être à la même échelle pour garantir des résultats reproductibles.(13) Sinon, des valeurs numériques différentes d'une même observation (i.e. 1000g, 1kg), résultent en différentes importances attribuées pendant la phase d'entraînement du modèle. Qu'une variable soit initialement soumise en grammes ou en kilogrammes, le modèle devrait toujours atteindre le même résultat. Il y a deux façons de mettre à l'échelle les prédicteurs : la normalisation et la standardisation.(6) La standardisation donne à la variable une moyenne de zéro et un écart-type unitaire.

$$x_i' = \frac{x_i - \bar{X}}{\sigma_X} \quad (8)$$

Avec la normalisation, les valeurs sont rééchelonnées entre 0 et 1. La normalisation a l'avantage de limiter les valeurs à une certaine plage ; cependant, elle rend la variable sensible aux valeurs aberrantes inaperçues.

$$x_i' = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (9)$$

Il fauter noter que les méthodes basées sur les arbres de décision fonctionnent par partitions récursives et ne reposent pas sur des calculs de distance euclidienne. Elles font partie des rares méthodes d'apprentissage automatique qui ne nécessitent pas une étape de mise à l'échelle.

Le réglage fin du modèle

La paramétrisation d'un modèle a deux composantes : les paramètres et les hyperparamètres. L'objectif de la phase d'entrainement est d'estimer les paramètres qui permettent le meilleur ajustement des observations. Les hyperparamètres sont des paramètres qui doivent être proposés au modèle avant l'estimation. Ils contrôlent la vitesse d'apprentissage et la complexité du modèle. Comme il n'y a pas moyen d'identifier dès le départ les bonnes valeurs pour les hyperparamètres d'un modèle, il faut les rechercher dans l'espace des hyperparamètres, voir figure 6. Un modèle doit être formé pour évaluer une combinaison d'hyperparamètres. Si un modèle possède 2 hyperparamètres appelés A et B, avec chacun pouvant prendre trois valeurs différentes, il y aurait un total de neuf modèles différents à tester. Le modèle le plus performant indique les valeurs d'hyperparamètres à choisir. En pratique, les combinaisons potentielles d'hyperparamètres sont représentées dans une grille. C'est ce qui donne le nom à la méthode, *recherche par grille*. Si l'espace de recherche n'est pas trop grand, et par le fait même la grille qui le représente, une recherche exhaustive est effectuée. Parfois, passer à travers la grille, et donc entrainer un modèle pour chaque élément, est trop exigeant pour les ressources de calcul disponibles. Dans ces cas, la recherche se limite à quelques tentatives choisies au hasard dans la grille, et le nombre de tentatives est spécifié pour correspondre aux ressources disponibles.(10) Bien que d'autres algorithmes de recherche puissent être appliqués, comme les algorithme heuristiques, (14) la recherche par grille, qu'elle soit exhaustive ou aléatoire, est la méthode la plus répandue pour optimiser les valeurs des hyperparamètres d'un modèle.

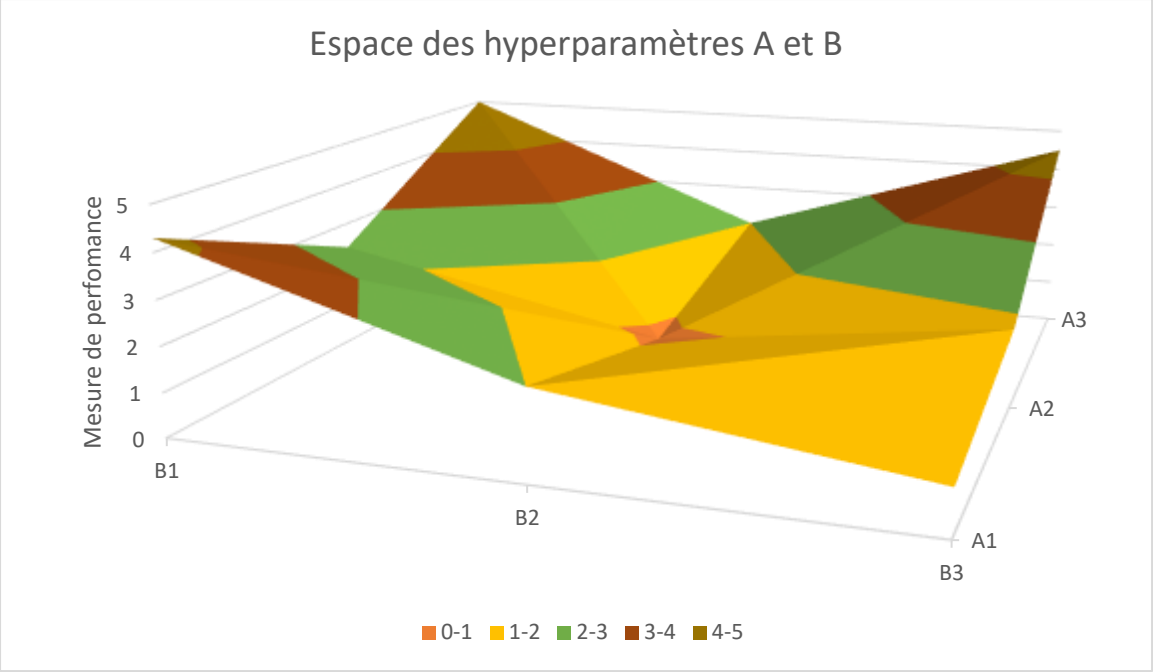

Figure 6. – Espace des prédicteurs pour une situation avec 2 prédicteurs

Les méthodes de rééchantillonnage

Validation croisée

Les deux objectifs d'un processus d'apprentissage sont : 1) estimer les paramètres durant la phase d'entraînement et 2) tester la performance du modèle choisi lors de la phase d'évaluation. Utiliser l'ensemble des données disponibles pour effectuer les deux tâches est une erreur. Tester un modèle avec des données *vues* par le modèle durant son entraînement mène à une estimation trop optimiste de la performance. Avant le début de la phase d'entraînement, une partie des données doit donc être tenue à l'écart afin d'éviter d'avoir une estimation biaisée. Cependant, cela réduit par le fait même la quantité de données disponibles pour l'entraînement. De plus, l'ensemble de données retenues, également appelé ensemble de validation, fournit seulement une unique estimation de l'erreur de ce modèle. L'erreur réelle, ou l'erreur que le modèle va induire lorsqu'il sera mis en production est supposée suivre une distribution normale. Ainsi, une estimation unique peut être trompeuse car elle peut ne pas être *proche* de la moyenne de la distribution de l'erreur et que finalement, le résultat varie lorsqu'une autre part de données est choisie comme ensemble de validation, rendant ainsi le

processus peu fiable. Pour résoudre ces trois problèmes, une solution est que l'ensemble de données soit divisé en n parties. L'entrainement est répété *n* fois pour que chaque partie soit utilisée, une fois à son tour, comme ensemble de validation. En fin de compte, elle produit une estimation moyenne de l'erreur réelle. De cette façon, on a maximisé la quantité de données sur lesquelles le modèle est entraîné et on obtient une estimation plus fiable. Cette méthode de rééchantillonnage est appelée "validation croisée par n parties". Le nombre de parties habituellement choisi est de 5 ou 10. Pour continuer avec l'exemple de recherche par grille mentionné plus haut, chacun des 9 modèles seraient entraînés 5 fois, une fois pour chaque partie. La procédure totale comprendrait donc 45 phases d'entraînement. Enfin, le modèle avec la plus petite erreur de validation croisée est conservé pour la suite des procédures. Comme les valeurs optimales des hyperparamètres sont maintenant connues, le modèle choisi est à nouveau entraîné, mais cette fois sur toutes les données d'entraînement, garantissant une estimation des paramètres du modèle avec les meilleurs conditions rassemblées.(9)

Le modèle final ayant achevé son apprentissage, il est maintenant temps de l'évaluer. Auparavant, le modèle choisi n'était pas entraîné à nouveau sur toutes les données de formation, et l'erreur de validation croisée a été désignée comme l'erreur d'évaluation.(15, 16) Cependant, il ne faut pas sélectionner les mêmes données pour l'estimation des paramètres, l'identification des hyperparamètres et l'évaluation d'un modèle. Cela signifie que de nouvelles données sont nécessaires. Elles sont connues sous le nom de l'ensemble test. Cet ensemble peut fournir une estimation juste. Malheureusement, les problèmes liés à une seule estimation de l'erreur refont surface. Il faut plus d'une estimation pour garantir une estimation fiable de l'erreur réelle. La solution s'appelle la validation croisée emboîtée. Avec cette méthode, l'ensemble des données est divisé en C parties et chaque C-ième partie est utilisée une fois comme ensemble de test. La procédure complète, y compris la validation croisée interne, est répétée pour chaque partie C de la boucle externe.(15) La structure générale est présenté dans la figure suivant inspiré le formule de Caret.(17)

```
Define a set of all the different hyperparameters combination to evaluate
For each iteration of the outer resample technique do
        For each combination of hyperparameters do
                For each iteration of the inner resample technique do
                        Hold-out specific samples
                        Fit the model on the remainder
                        Predict the hold-out samples
                End
                Calculate the average performance across the hold-out prediction
        End
        Determine the optimal combination of hyperparameters
        Fit the final model to all the training data using the optimal combination of hyperparameters
End
Assets the final performance by calculating the average performance across the hold-out prediction
```

Figure 7. – Validation croisée emboîtée

Chapitre 2 – Arbres de décision et méthodes ensemblistes

Arbres de décision

Introduite en 1984, CART (Classification and Regression Trees) est une technique qui a pour but de compartimenter l'espace des prédicteurs en plus petites régions permettant la prédiction d'une réponse pour toute nouvelle observation respectant les caractéristiques d'une région.(9, 18) La compartimentation de l'espace se fait par fractionnements binaires récursifs.

Tableau 1. – Jeu de données factices

A	B	y	$\hat{y}$
7	2	16	14
7	3	10	14
7	7	19	14
7	8	14	14
7	8	12	14
8	1	11	14
8	4	17	14
8	6	13	14
1	2	37	36
2	1	34	36
3	2	35	36
4	2	35	36
5	2	39	36
1	6	62	58

1	8	55	58
2	7	59	58
3	7	60	58
4	6	59	58
5	6	53	58
5	8	58	58

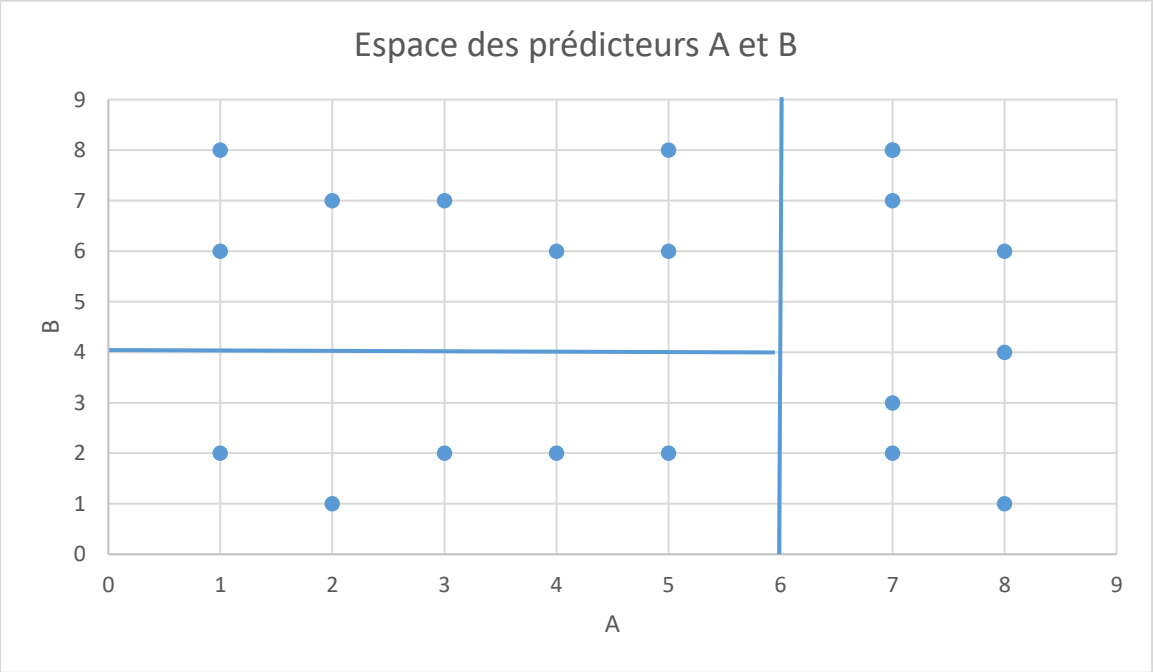

Figure 8. – Espace des prédicteurs du tableau 1

Comme son nom l'indique, CART peut être utilisé comme classifieur ou comme régresseur. Les règles permettant la séparation de l'espace des prédicteurs peuvent être représentées graphiquement sous la forme d'un arbre inversé. Le nœud supérieur est appelé nœud racine et les nœuds terminaux représentent les feuilles. Ces derniers ne comportent pas de critère de séparation comme les nœuds de décision et le nœud racine. Ils représentent les régions définies par l'algorithme.

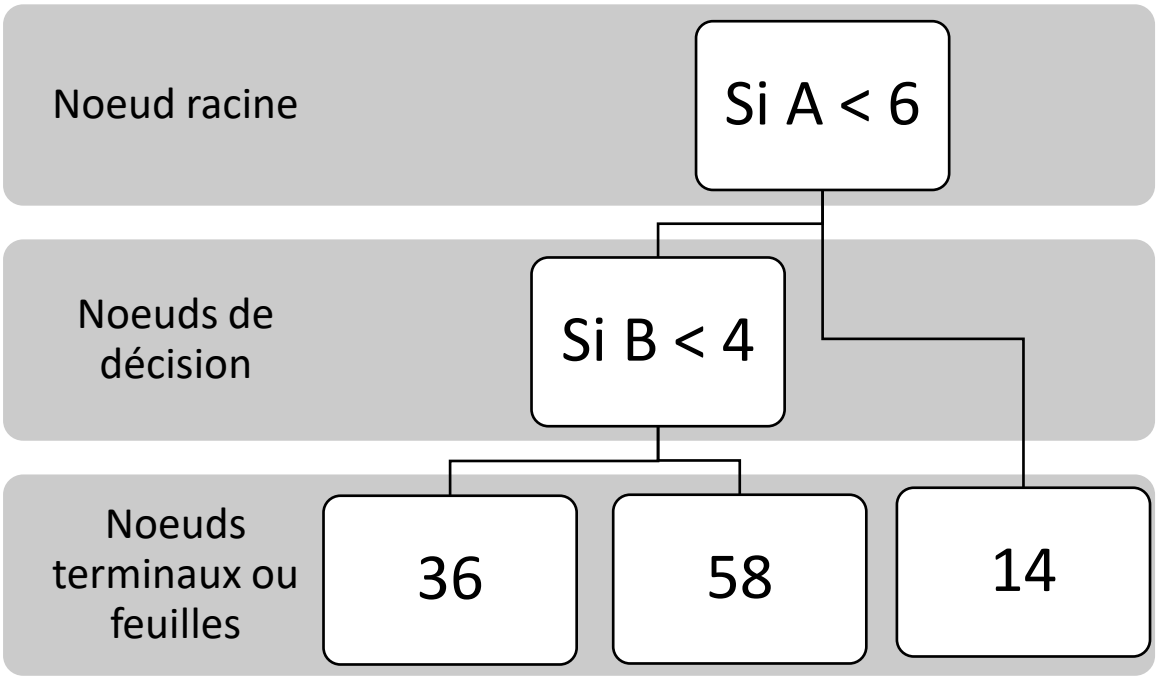

Figure 9. – Arbre de décision lié aux données du tableau 1

Pour faire une prédiction, une observation sera évaluée selon un ensemble de conditions qui permettront d'attribuer cette observation à une région en particulier de l'espace des prédicteurs. Le processus d'attribution débute avec le nœud racine. L'observation est assignée à la branche gauche du nœud lorsqu'elle respecte les conditions de scission. Selon l'exemple du tableau 1 et des figure 8-9, si l'observation à une valeur de A inférieure au seuil de séparation 6, elle continuera sa descente à gauche. L'observation descendra encore et encore et sera assignée à l'une ou à l'autre des branches de chaque nœud rencontré jusqu'à ce qu'un nœud terminal soit atteint. Enfin, la valeur de ce nœud sera utilisée comme prédiction. Cette dernière correspond à la moyenne des valeurs de sortie des observations se trouvant dans la région en question. Ainsi selon la figure 8, une observation $X_{(A,B)}$ = (2, 1) recevrait une estimation $\hat{Y}$ de 36 considérant que la branche de gauche correspond au respect des seuils de séparation pour cet arbre de décision.

La construction de l'arbre, également appelée phase d'entraînement, débute avec le choix de la composition du nœud racine. Deux items sont à déterminer : le prédicteur à utiliser ainsi que la valeur du seuil de séparation. La recherche se fait de manière exhaustive. Pour les prédicteurs

de type continu, comme la variable A possède 7 éléments différents, il y a 6 seuils possibles de séparation, soient : 1,5; 2,5; 3,5; 4,5; 6; 7,5. Pour les variables discrètes, il existe un nombre de séparations égal à la moitié du nombre de combinaisons différentes. Par exemple, pour une variable discrète de 3 catégories, il y a 7 séparations possibles à faire comme le démontre le tableau2.

Tableau 2. – Séparations possibles à 4 catégories

1	234
2	134
3	124
4	123
1,2	3,4
1,3	2,4
1,4	2,3

Pourquoi le nœud racine a hérité de la variable A et pourquoi la séparation se fait à 6 en particulier? Cette combinaison a été retenue parmi toutes les combinaisons car elle correspond à la combinaison provoquant la plus faible impureté. L'impureté d'un nœud indique dans quelle mesure la prédiction de ce nœud– $\hat{y}$– correspond aux observations – Y– se retrouvant dans la région représentée par celui-ci (2, 9). Comme mentionné plus haut, la valeur d'un nœud terminal d'un arbre de régression correspond à la moyenne des valeurs le constituant. La prédiction d'un nœud héritier d'un bon critère de séparation, dit très peu impur, correspondra fortement aux valeurs observées et *vis-versa*.

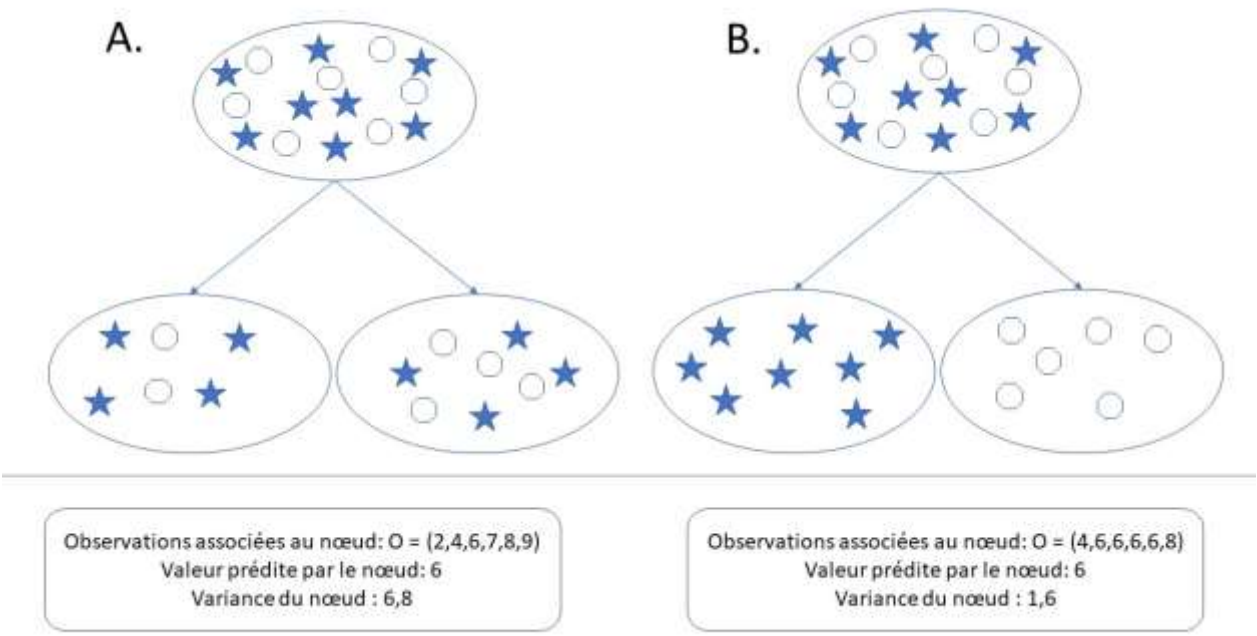

Figure 10. – Concept d'impureté

Avec les arbres de régression, l'impureté vise à minimiser la différence entre les valeurs observées et la prédiction du nœud en utilisant la *somme résiduelle des carrés* (SRC) qui se retrouve à être la variance du nœud comme le démontre l'équation ci-dessous et la figure 10.

$$I = \sum_{i=1}^{N} \frac{(y_i - \bar{y})^2}{N_i} \quad (10)$$

Comme mentionné plus haut, un CART cherche à compartimenter l'espace des prédicteurs. Considérons la situation initiale où il n'y a pas encore eu de séparation. L'entièreté de l'ensemble des prédicteurs est considérée comme une seule région. Celle-ci peut être représentée comme étant un nœud terminal fournissant la moyenne comme prédiction aux observations. Si on prend l'exemple de l'arbre représenté dans la figure 8, cette région contient l'ensemble des observations, soit 20 observations. La moyenne des y est de 34,9 et la variance de ce nœud est de 368,79. L'objectif est donc de trouver la meilleure façon de séparer l'espace des prédicteurs. Ceci est défini comme étant la séparation qui entraînera la plus grande réduction de variance. Comme m'entonné plus haut, la recherche est exhaustive. La séparation

peut se faire sur l'axe du prédicteur A ou sur l'axe du prédicteur B. Aux choix pour le prédicteur A mentionnés précédemment, s'ajoutent les choix pour le prédicteur B : 1,5; 4; 6,5; 7,5. En regardant la figure 8, on observe que le seuil A < 6 est celui qui a été reconnu comme étant celui provoquant la plus grande diminution de variance, et il a donc été choisi pour composer le premier nœud. Le calcul de la diminution de la variance se trouve à être la différence entre l'impureté du nœud parent I_p et la moyenne pondérée des nœuds fils gauche I_G et droit I_D.

$$\Delta I = I_p - \left[\frac{n_G}{N} I_G + \frac{n_D}{N} I_D\right] \quad (11)$$

Ainsi, on peut voir que plus l'impureté des nœuds fils est petite, plus la réduction est grande. On peut noter que le premier terme, celui de l'impureté du nœud parent, est fixe peu importe le choix de la séparation, et donc facultatif dans le calcul.

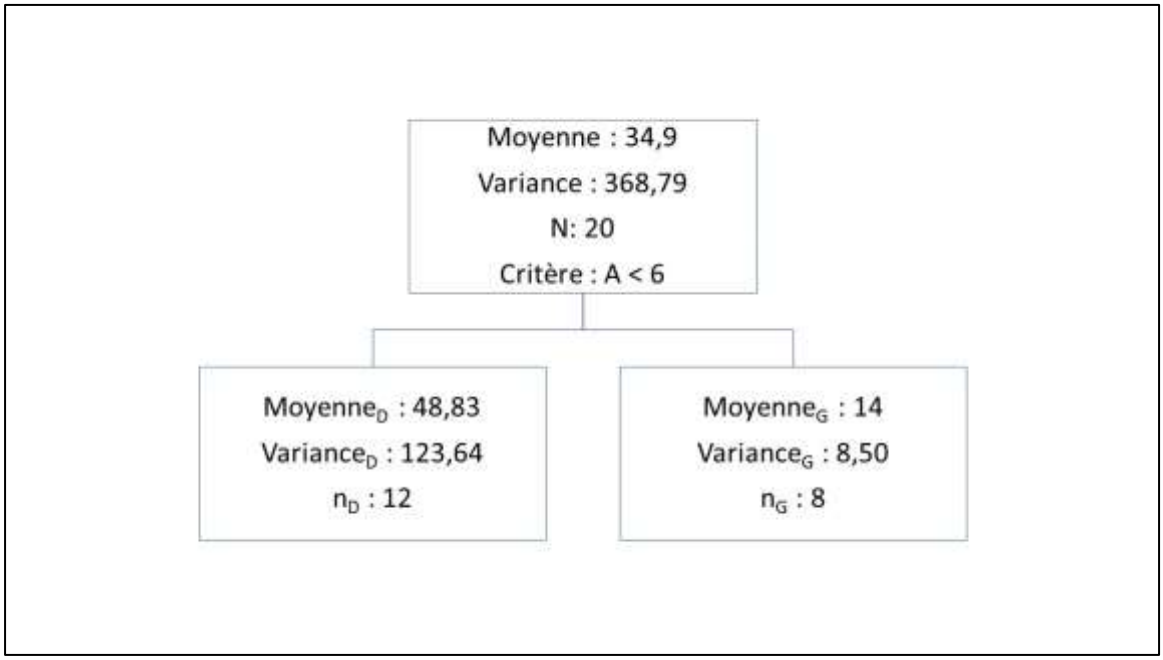

Figure 11. – Exemple de calcul d'impurté basé sur la variance du premier nœud de la fig. 9

À ce stade, l'arbre n'est composé que d'un niveau, c'est-à-dire que les deux nœuds fils du nœud racine sont des nœuds terminaux. Autrement dit, l'espace des prédicteurs est seulement séparé en deux. La suite des procédures cherchera à diminuer la variance présente dans ces deux régions en séparant à nouveau ces régions après avoir identifié le meilleur seuil de séparation pour chacune des régions. La phase de construction continue à ajouter des nœuds dans l'arbre

jusqu'à ce que les nœuds terminaux et les régions qu'ils représentent contiennent une certaine quantité minimale d'observations.

L'un des grands avantages des CART est leur interprétabilité. en effet, le modèle peut facilement être représenté par un schéma. De plus il est facile d'obtenir l'importance d'un prédicteur, qui est définie comme étant la somme de l'apport à la réduction de la variance de chaque nœud relatif au prédicteur d'intérêt, pondérée par la probabilité pour une observation d'accéder au nœud(19)

$$Imp(x) = \sum_{x \leftrightarrow y} \frac{N_t}{N} \Delta I_{(t,s_y)} \quad (12)$$

où N_t est le nombre observations atteignant le nœud t, N le nombre total de nœud et $\Delta I_{(t,s_y)}$ la diminution de l'impureté attribuée à la séparation s du nœud t relative à la variable y et notée s_y.

Théoriquement, il serait possible de faire grandir un arbre de décision jusqu'à ce que chaque observation ait son propre nœud terminal. À ce moment, tous les derniers nœuds de décision auraient une impureté et une valeur 0 de SRC, étant donné que la valeur du nœud terminal serait identique à la valeur de sortie de l'observation. Cette situation correspond à un cas parfait de surentraînement. Ainsi la croissance est arrêtée lorsque les nœuds terminaux contiennent une certaine quantité minimale d'observations.

En plus, il y a généralement une étape d'élagage pour vérifier le rapport coût-bénéfice de chaque nœud en termes de performance de l'arbre. Plus l'arbre est grand, plus il a de chances de capter la relation f, ce qui promet un faible biais. Cependant plus l'arbre est grand, plus il risque de trop s'adapter aux données d'entraînement et d'avoir une variance plus grande. Malgré des restrictions et des techniques d'élagage, les arbres de décision ont tendance à souffrir de surentraînement et à présenter un compromis biais-variance déséquilibré lorsque mis en production.

Bootstrap aggregation (bagging)

Pour résoudre les problèmes de variance des CART, des techniques ensemblistes telles que le *bagging* ont été proposées.(20) Le bagging est généralement appliqué aux arbres de décision, mais peut théoriquement être utilisé avec d'autres types de méthodes. Ce type de solution est inefficace pour les méthodes ayant un faible biais. Étant donné un ensemble de n observations : $z_1, z_2, \ldots, z_n$, ayant chacun une variance σ^2, la variance de la moyenne $\bar{Z}$ diminue avec l'augmentation de la taille selon σ^2/n. Ainsi, prendre la moyenne de plusieurs modèles, i.e. plusieurs arbres de décision, diminue la variance du résultat. Habituellement, seul un ensemble d'entrainement est disponible et un seul arbre de décision est généré. C'est ici qu'intervient le *boostrap*.

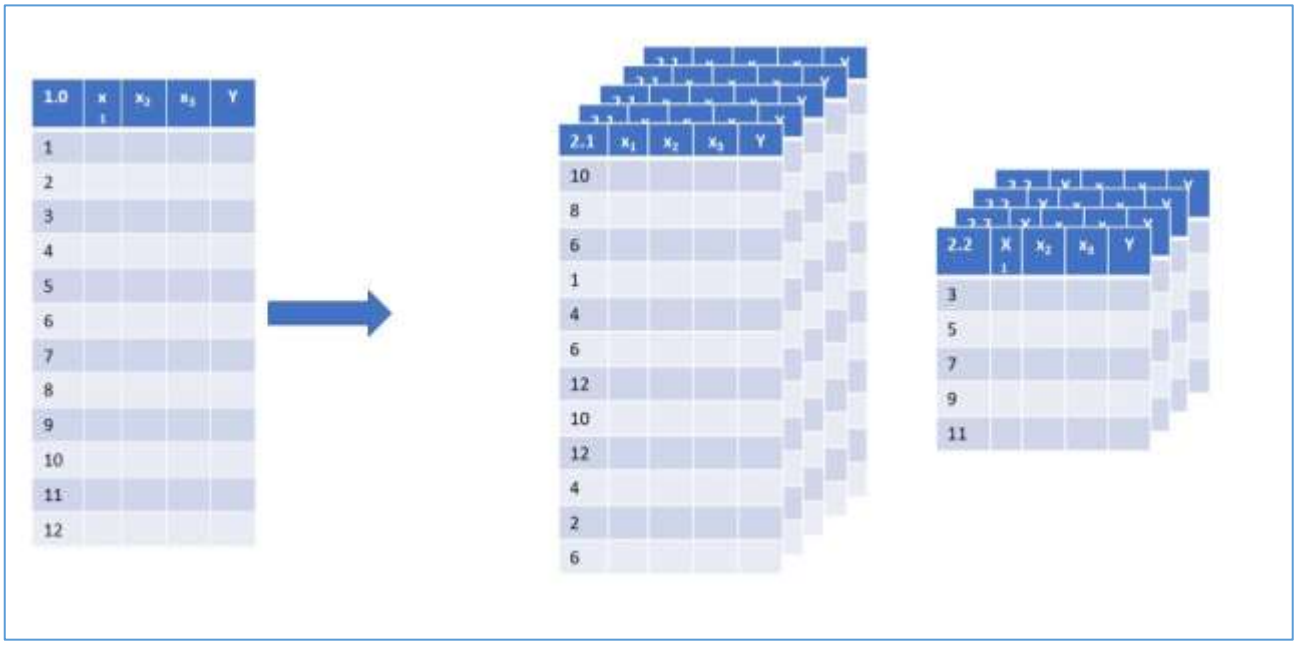

Figure 12. – Procédure de bagging avec à gauche le jeu de données original, les différents jeu de données échantillonés avec remise et leur ensemble out of bag.

Avec une procédure de *bagging*, illustré dans la figure 12 et 13, *B* sous-ensembles de données provenant de l'ensemble original sont générés. L'échantillonnage de l'ensemble se fait avec remise et arrête lorsque la taille originelle est atteinte. Les B sous-ensembles permettent

l'entrainement parallèle de *B* arbres de décision. Quelques centaines d'arbres diminuent habituellement la variance de manière appréciable. La moyenne des prédictions de l'ensemble des arbres a normalement une variance plus faible que la prédiction d'un seul arbre. Le *bagging* bénéficie de la précision d'arbres pleinement développés, ayant un faible biais, et réduit leur variance en utilisant la moyenne de leur prédiction. De plus, la procédure de *bagging* a pour avantage de fournir un ensemble test intégré, rendant inutile le besoin de faire une validation croisée. Puisque le rééchantillonnage est avec remise, certaines observations se retrouvent plusieurs fois sélectionnées dans un sous-ensemble. La performance est donc calculée sur les observations non sélectionnées (observations hors du sac) et ce, pour chaque sous-ensemble, permettant d'avoir une distribution empirique de la performance i.e. la moyenne et l'écart-type.

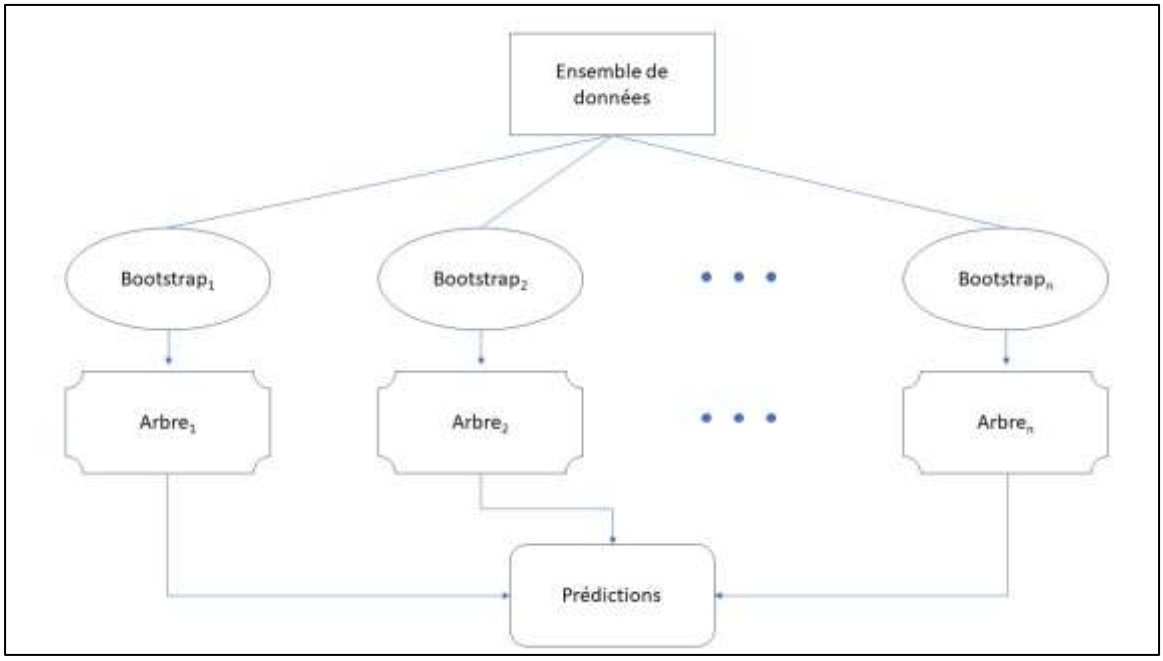

Figure 13. – Bagging de CARTs pour former une forêt aléatoire

Forêt aléatoire

L'idée d'introduire de la variabilité entre les arbres en les entrainant sur des sous-ensembles différents est essentielle pour le concept de *bagging*. Répéter l'entrainement *B* fois sur le même arbre CART donnerait *B* clones du même arbre de décision et la variance de la forêt serait

semblable à celle d'un seul arbre. En utilisant un rééchantillonnage, on s'assure que les arbres sont différents. Cela dit, il reste que les arbres sont largement corrélés entre eux. La réduction de la variance est moins importante lorsque les modèles sont corrélés entre eux que lorsqu'ils ne le sont pas. Malgré des ensembles différents, si un prédicteur est très important, il sera toujours choisi en haut de chaque arbre et la logique inverse s'applique pour les prédicteurs de faible importance. Les arbres ne sont pas des clones mais ils restent semblables. La méthode des forêts aléatoires apporte une solution sous la forme d'une étape supplémentaire au *bagging*.(21) Elle permet de décorréler les arbres en limitant le nombre de prédicteurs disponibles lors du choix de chaque nœud.

La technique des forêts aléatoires est très appréciée par la communauté et est reconnue pour sa performance. Cela peut s'expliquer par le fait que le réglage fin du modèle soit assez simple et contienne peu de paramètres à gérer. Les inventeurs ont proposé des valeurs pour le nombre maximal de prédicteurs admissibles lors du choix de chaque nœud, qui est habituellement le ⅓ du nombre total des prédicteurs pour un arbre de régression, et la quantité minimale d'observations dans les nœuds terminaux est de 5. Ces recommandations peuvent être vues comme des valeurs par défaut, mais ces paramètres peuvent quand même être inclus dans une grille de recherche lors du réglage fin. (2)

1. Pour b = 1 jusqu'à B arbre
    a. Collecter un échantillon e_b de taille N de l'ensemble de données E.
    b. Construire un arbre A_b avec e_b.
        i. Sélectionner une fraction *m* de l'ensemble des prédicteurs *p*
        ii. Identifier le meilleur choix de nœud parmi les *m* prédicteurs
        iii. Maintenir la croissance de A_b jusqu'à atteindre le critère d'atteinte de la quantité minimale d'observations
        iv. Faire l'élagage de A_b
2. Rassembler l'ensemble des A_b pour la prédiction
    a. $\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} A_b\,(x)$

Figure 14. – Pseudo code d'une forêt aléatoire

Le *boosting* est une solution ensembliste, plus précisément un modèle additif progressif, proposé pour réduire le biais des modèles. (22) Le *boosting* a également l'avantage de réduire la variance. Il tire sa source de la théorie de *la force des modèles d'apprentissage faibles*. (23) Il a été prouvé que les modèles d'apprentissage faibles (*weak learners*) pouvaient aussi bien, ou sinon, mieux performer qu'un modèle d'apprentissage fort (*strong learners*) lorsque combinés.(2) Cette alternative ne possède pas le fardeau computationnel et le risque de surapprentissage qui viennent avec l'entrainement d'une modèle d'apprentissage fort. (23) Un modèle d'apprentissage faible est défini comme un modèle étant légèrement supérieur à un estimateur aléatoire. Un exemple commun de modèle d'apprentissage faible est une *souche décisionnelle*. i.e. un arbre décisionnel ayant seulement un niveau. Un modèle d'apprentissage fort est défini comme un modèle ayant de bonnes prédictions sur la grande majorité de ses observations. La plupart des méthodes en apprentissage automatique peuvent être considérés comme des modèles d'apprentissage forts : machine à vecteurs de support, réseau de neurones, arbre décisionnel *mature, etc.*

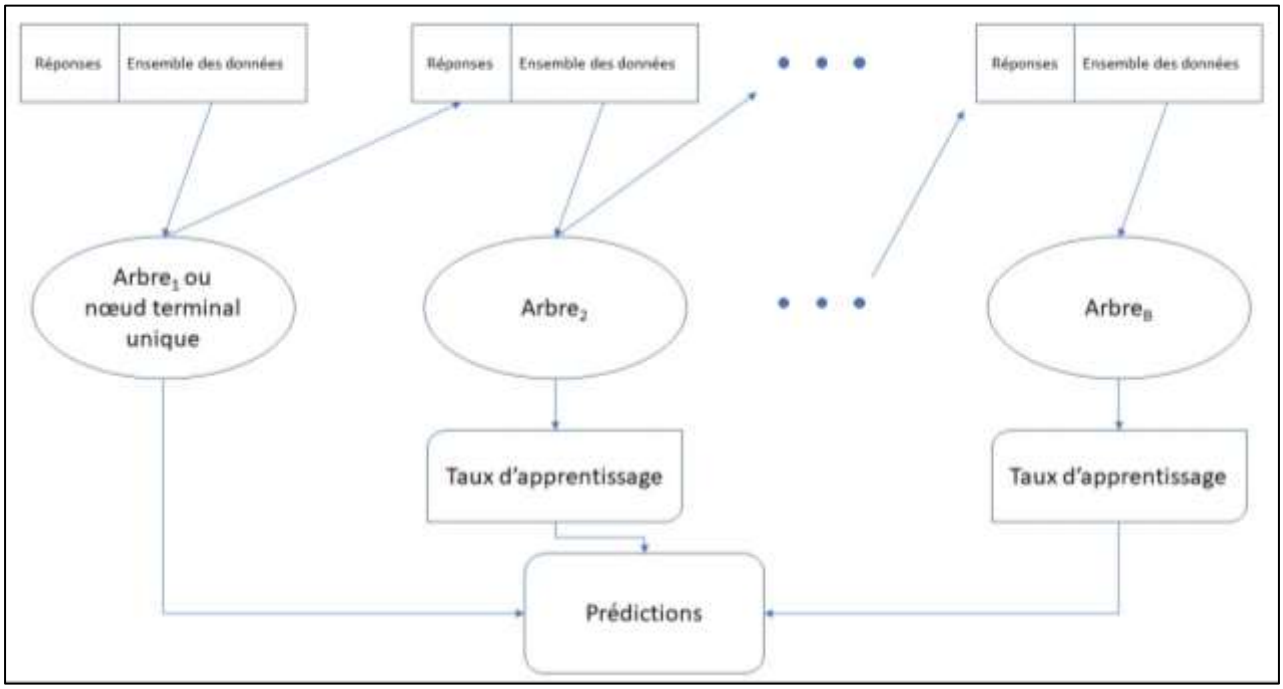

Figure 15. – Boosting d'arbres CART

Le concept boosting consiste à empiler de nombreux petits modèles, voir figure 15. Le *boosting* est aussi une méthode ensembliste. Bien qu'avec le *bagging* les arbres soient générés en parallèle de manière indépendante, ce n'est pas le cas du *boosting*. Avec ce dernier, les arbres sont construits de manière séquentielle et chaque arbre est dépendent de l'arbre qui le précède dans la mesure où chaque arbre se concentre sur les faiblesses du précédent. Pour augmenter l'importance des réponses – y – mal prédite par l'arbre précédent, l'arbre actuel s'entraine sur une version de l'ensemble des données modifié de la réponse y. Cela peut prendre la forme d'une surreprésentation des observations mal prédites où un arbre peut s'entrainer sur les résidus de l'arbre précédant. Ainsi, le modèle $\hat{f}$ s'améliore en réduisant le biais à mesure que les arbres s'ajoutent. La prédiction de nouvelles observations se fait avec l'ensemble des arbres. Chaque arbre contribue à la prédiction selon un paramètre qui contrôle l'apport de chaque arbre au vote final. Ce paramètre est le taux d'apprentissage. Dans certaines versions du boosting, ce paramètre est fixe, alors que pour d'autres versions, ce paramètre dépend de la qualité de la prédiction de l'arbre. Un taux d'apprentissage faible ralentira la procédure d'apprentissage et rendra nécessaire l'utilisation d'un grand nombre d'arbres pour atteindre l'objectif. Le nombre d'arbres exact n'a pas de réelle importance avec du *bagging* du moment qu'il est suffisant à la réduction de la variance. Autrement dit, on ne peut pas en avoir trop. En avoir beaucoup ne mène pas à du surajustement, mais seulement à un temps de calcul plus élevé. C'est une autre histoire avec du *boosting* qui, par son fonctionnement, génère des arbres indépendants les uns des autres. Avec le *boosting*, à chaque itération, chaque arbre tente d'apprivoiser les faiblesses de $\hat{f}$, mais *ne* génère pas des arbres indépendants. Ainsi, comme il a été expliqué précédemment, mettre trop d'effort pour s'adapter aux données d'entrainement mène à du surapprentissage. Le nombre d'arbres est un paramètre à surveiller avec le boosting.

Pour s'assurer d'avoir des modèles d'apprentissage faible, la profondeur des arbres est limitée selon un paramètre spécifique. Pour certaines versions, la profondeur des arbres utilisés correspond carrément à des souches décisionnelles, alors que pour d'autres versions, la profondeur est plus importante, sans être pour autant celle d'un arbre *mature*. (2)

1. Établir la situation initiale $\hat{f}(x) = 0$ et $r_i = y_i$ pour les i observations
2. Pour b = 1 jusqu'à B arbre
    a. Construire un arbre A_b.
        i. Identifier le meilleur choix de nœud parmi les prédicteurs
        ii. Maintenir la croissance de A_b jusqu'à atteindre le critère taille d.
        iii. Faire l'élagage de A_b
    b. Mettre à jour $\hat{f}(x)$
        i. $\hat{f}(x) = \hat{f}(x) + \epsilon A_b(x)$
    c. Mettre à jour r_i pour chaque observation *i*
        i. $r_i = r_i - A_b(x_i)$
3. Rassembler l'ensemble des A_b pour la prédiction
    a. $\hat{f}(x) = \sum_{b=1}^{B} \varepsilon A_b(x)$

Figure 16. – Pseudo code d'un algorithme de boosting

Gradient boosting machine (GBM)

L'apprentissage supervisé se résume à l'approximation d'une fonction f. L'approximation se fait en deux étapes : la détermination d'une fonction de perte *L,* comme l'équation 13 et sa minimisation.

$$\mathrm{L}(f) = \sum_{i=1}^{N} L\big(y, f(x_i)\big) \quad (13)$$

Les techniques d'approximations par additions progressives comme le *boosting* sont des techniques qui n'optimisent pas tous leurs paramètres d'un seul coup comme le font les réseaux de neurones, les expansions de fonction de bases, ou les modèles additifs généralisés (GAM). À la place, ils tentent d'optimiser les paramètres de l'actuel modèle en gardant les modèles précédents (et leurs paramètres) fixes. Cela ralentit le processus d'apprentissage mais ralentit également l'apparition du surapprentissage. Les techniques d'approximation pas additions

progressives peuvent être considérées de par leur fonctionnement comme un type de problèmes d'optimisation numérique, avec la particularité que les paramètres à optimiser sont des fonctions, comme on peut le voir avec l'équation 14.

$$\hat{f} = \underset{f}{arg\,min}\ L\,(f)\ (14)$$

Ce problème peut se résoudre par une technique d'optimisation itérative comme une descente du gradient. L'objectif est de trouver le meilleur arbre, celui qui nous rapproche du minimum de la fonction de perte, et cela se fait en créant un arbre qui s'entraine sur la solution négative du gradient de la fonction de perte en respect de la dernière itération. Cette solution se trouve à être équivalente aux résidus de l'ancien arbre, sachant que la fonction de perte la plus souvent choisie est une variation du carré de l'erreur et que son gradient équivaut à la définition des résidus.(2)

La fonction de perte:

$$L = \frac{1}{2}[y_i - f(x_i)]^2\ (15)$$

Son gradient à la position de l'arbre précédent noté f_{m-1} :

$$g_{im} = \left[\frac{\partial L(y_i, f(x_i)}{\partial f(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)} \quad (16)$$

Le négatif du gradient équivaut exactement aux résidus:

$$-g_{im} = y_i - f(x_i)\ (17)$$

Le GBM est donc une variation du *boosting* générique ou chaque arbre s'entraine sur les faiblesses de l'arbre précédent. Cette méthode itérative travaille à réduire la somme des résidus en empilant les modèles d'apprentissage faibles qui minimisent la fonction de perte à chaque étape. L'arbre initial est une souche; ainsi le premier arbre prédit pour l'ensemble des $\hat{y}_i$ la valeur de la moyenne des y_i. Les arbres subséquents sont plus grands mais restent tout de même des modèles d'apprentissage faible. Pour donner d'autre exemples, Adaboost, la première implémentation du boosting, est un type de boosting qui tente d'optimiser une

fonction de perte exponentielle.(2, 23) Le gradient résulte en une version pondérée des y_i au lieu des résidus. Cela dit, Adaboost peut être vu comme une version du GBM. Le GBM est mathématiquement défini pour supporter n'importe quelle fonction de perte.(2)

Par la suite chaque arbre tente de faire un pas dans la bonne direction pour que la somme des différences entre $\hat{y}_i$ et y_i tende vers 0. Cependant, pour éviter le surapprentissage et pour donner la chance à plusieurs arbres de traiter ces dites-faiblesses, l'apport de chaque arbre est réduit par la valeur du taux d'apprentissage.

Plus récemment, en s'inspirant des propriétés de *bagging*, le GBM incorpore de l'échantillonnage durant l'entrainement en optimisant par descente de gradient stochastique.(2) Cela permet de réduire la variance et le temps de calcul. À chaque itération, seule une fraction des données d'entrainement est utilisée, habituellement la moitié de ces données.

eXtrem Gradient *boosting* (XGBoost)

XGboost est une amélioration de GBM.(24) Il prend en compte la complexité des arbres du modèle et tente de les limiter dans le but de réduire les risques de surapprentissage. La fonction objective à considérer est composée d'un terme mesurant la précision du modèle, une fonction de perte, et d'un terme habituellement absent des autres techniques et qui pénalise la complexité.

$$obj(\theta) = L(\theta) + \Omega(\theta) \quad (18)$$

Cette expression, ou θ représente les paramètre d'une modèle, n'est pas sans rappeler la définition de l'erreur d'apprentissage, l'équation 3. La fonction de perte représente le besoin d'un modèle efficace ayant un petit biais. Le concept de pénalité de la complexité est un concept de régularisation qui tente de simplifier le modèle pour empêcher qu'il devienne trop complexe et par le fait même trop adapté aux données. L'expression de la fonction objective du XGBoost a pour but de satisfaire le compromis biais-variance. Concrètement, XGBoost possède trois grandes différences avec le GBM :

1. Pour faciliter l'optimisation de la fonction objective, XGBoost utilise une approximation de fonction, l'approximation de Taylor de second degré. Cela permet d'obtenir de meilleurs résultats que de simplement prendre le gradient.(24)
2. La construction d'un arbre avec XGBoost est différente de la construction des CART. Le critère de séparation est semblable à la minimisation de l'impureté mais possède des termes de régularisation λ.

$$Gain = \frac{1}{2}\left[\frac{G_L}{H_L - \lambda} + \frac{G_R}{H_R - \lambda} - \frac{G_L + G_R}{H_L + G_R - \lambda}\right] - \gamma \quad (19)$$

   Cette équation a été développée pour supporter une large gamme de fonctions de perte. Les deux premiers termes représentent la qualité (impureté) des nœuds fils, le troisième terme représente la qualité du nœud parent et le quatrième terme, identifié γ, contrôle l'élagage. L'équation possède la somme des gradients et des hessiennes de la fonction de perte nommé G et H. Ces termes proviennent du polynôme de Taylor. En situation de régression, si γ vaut 0, G et H équivalent simplement à la variance. Le paramètre γ facilite l'élagage des arbres et provient du terme Ω de la fonction objective; il sert à la régularisation du modèle.
3. Finalement, XGBoost est différent du GBM par sa rapidité de calcul. Les calculs ont été simplifiés à plusieurs endroits lors de la construction des arbres. De plus le code source de l'algorithme permet, par sa gestion des données, le calcul en parallèle.

Importance des prédicteurs

Les techniques de *bagging* et de *boosting* permettent d'identifier l'impact de chaque variable sur les prédictions du modèle. Deux types de mesures d'importance des variables sont largement utilisés dans la pratique : la diminution moyenne de l'impureté, applicable aux forêts aléatoires et à l'XGBoost, et une mesure obtenue par permutation, unique aux forêts aléatoires. (2)

La diminution moyenne de l'impureté examine la qualité de chaque nœud associé à la variable d'intérêt. Le calcul est semblable au calcul de la diminution de l'impureté. Cette méthode

somme la diminution de l'impureté causée par les séparations attribuables à la variable, et ce, pour tous les arbres, pour ensuite en extraire la moyenne, le processus est décrit dans l'équation suivante.

$$Imp(x) = \frac{1}{A}\sum_{t=1}^{A}\sum_{x \leftrightarrow y}\frac{N_t}{N}\Delta I_{(t,s_y)} \quad (20)$$

Avec N_t le nombre d'observations atteignant le nœud t, $\Delta I_{(t,s_y)}$ la diminution de l'impureté attribuée à la séparation s relative à la variable y , et A le nombre d'arbres dans la forêt.

Étant donné que cette mesure se base sur le calcul de diminution de l'impureté, elle peut être considérée biaisée. Une variable continue ou à haute cardinalité a plus de chance d'être sélectionnée étant donné que ce type de variables possède plus de séparations possibles qu'une variable binaire.(10) Par exemple, si lors du choix de la séparation, il y a une variable continue de 10 valeurs différentes et une variable binaire, 9 des 10 seuils de séparation sont reliés à la variable continue. De plus, cette mesure de l'importance est mesurée lors de l'entrainement de l'arbre et peut être affectée si l'arbre souffre de surapprentissage. Ces limitations ne rendent pas la mesure de l'importance basée sur la diminution de l'impureté inutile, seulement il est nécessaire de les garder en mémoire lors de l'évaluation de l'importance des variables.

La méthode basée sur la permutation quantifie la différence moyenne de la prédiction *hors sac*, voir figure 17. Après la construction d'un arbre, sa performance est calculée en utilisant les données gardées *hors du sac*, les valeurs non sélectionnées lors de l'échantillonnage *bootstrap*. Pour obtenir l'importance d'une variable, cette étape est réalisée à nouveau. Seulement cette fois, la variable d'intérêt est permutée, brisant le lien structurel entre les différents prédicteurs. Cette mesure compare les prédictions de l'ensemble de données *hors sac* avec la variable randomisée et l'ensemble de données *hors sac* avec la variable intacte. Une variable randomisée qui n'entraîne pas de baisse de performance implique que l'arbre ne lui accorde pas beaucoup d'importance. Cette méthode à l'avantage de ne pas être dérivée de la phase d'entrainement, mais augmente la quantité de calculs requis.

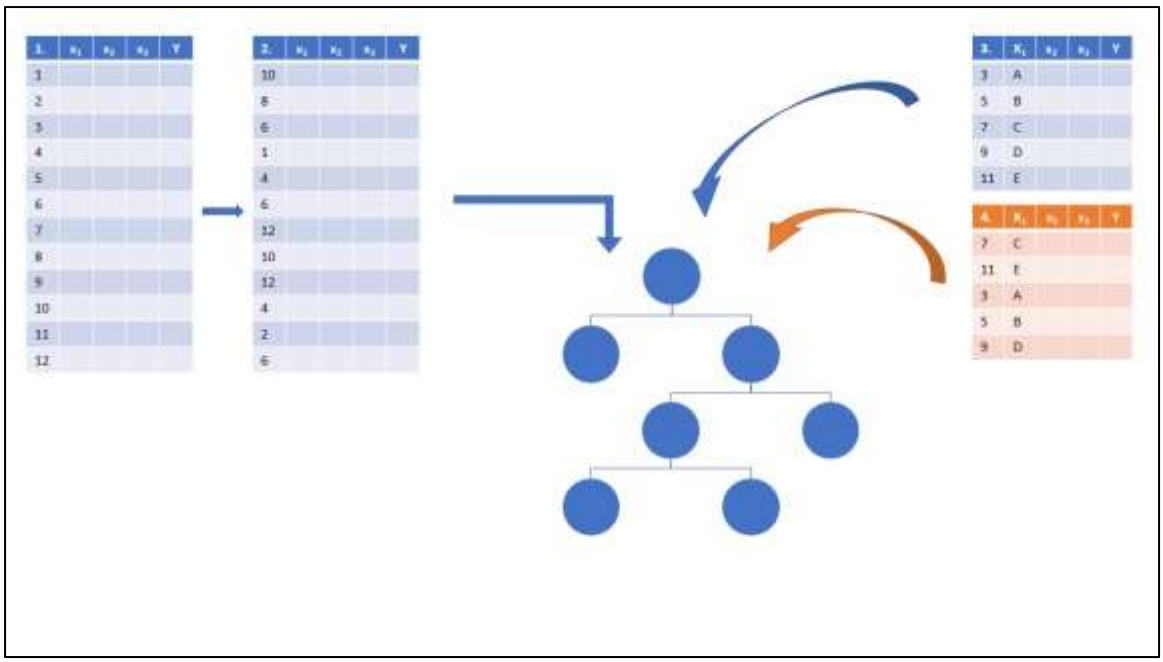

Figure 17. – Importance basée sur la permutation de la variable dans l'ensemble *hors sac*

L'objectif de ce mémoire est d'explorer le potentiel de l'apprentissage automatique dans les sciences pharmaceutiques. Cette exploration, faite à l'aide de techniques et de méthodes d'apprentissage automatique, a été réalisée par l'investigation de plusieurs situations typiques rencontrées en pharmacologie clinique et en pharmacométrie. Le projet est divisé en trois parties distinctes. La première partie propose un algorithme pour améliorer la fiabilité de l'étape de présélection des covariables d'un modèle de pharmacocinétique de population. La deuxième partie confirme la possibilité d'estimer les concentrations plasmatiques avec des méthodes d'apprentissage automatique. La troisième partie démontre l'utilisation des méthodes d'apprentissage automatique pour la prédiction de relations complexes, comme certaines relations pharmacologiques cliniques typiques.

Chapitre 3 – Application of tree-based Ensembles Methods to three pharmacometrics common tasks

En préparation

Paul-Antoine Leboeuf, Fahima Nekka

Faculté de pharmacie, Université de Montréal

Keywords: Machine learning, Ensemble Methods, Pharmacometrics, Pharmaceutical sciences, Random Forest, eXtreme Gradient Boosting.

Abstract

Machine learning offers tools to deal with actual problems. Recent breakthroughs in computational sciences and the emergence of the big data phenomenon have brought machine learning to the forefront in both academia and society. The recent achievements of machine learning in natural language, computational vision and medicine speak for themselves. The list of sciences and fields that benefit from machine learning techniques is long.

However, attempts to cooperate with pharmacometrics and related sciences are timid and limited. The aim of this thesis is to explore the potential of machine learning in pharmaceutical sciences. Specifically, this has been done through the application of machine learning techniques and methods to situations of clinical pharmacology and pharmacometrics. The project was divided into three parts. The first part proposes an algorithm to enhance the reliability of the covariate pre-selection step of a population pharmacokinetic model. The second part confirms that it is possible to estimate plasma concentrations with machine learning methods. The third part confirms the possibility of using machine learning methods for the prediction of complex relationships, as some typical clinical pharmacology relationships.

Introduction

Implementation of machine learning (ML) methods into population pharmacokinetic (PopPK) modeling, biopharmaceutical analyses, and clinical pharmacology have begun to attract attention in the last few decades. These methods have been employed to various tasks and purposes. Examples include the estimation of the plasmatic drug concentration over time, and

optimal regimen optimization and improvement of population modeling procedures. (1-8) ML methods generated a lot of research interest for their unique data driven characteristics that may be considered advantageous over traditional mechanistic-based models. In fact, ML methods are not based on mechanical assumptions such as current bio- or pharmaco-mathematical models. While ML methods may be considered by some to be black-box methods, they have the advantage of simplifying the modelling process since no prior biological knowledge is required and thus reduce the associated human bias.(6) Moreover, ML methods are known for their ability to successfully deal with numerous variables simultaneously and to handle nonlinear and complex relationships with ease, potentially qualifying them by the very fact as adequate methods for studying phenomena related to the biopharmaceutical field. Unfortunately, interpretability, which is an important aspect in drug therapy, seems to be lacking in ML methods, preventing their endorsement from the clinical pharmacologists at large. While ML was introduced in the 2018 American Conference on Pharmacometrics, there is still a long way to go before being fully embraced by the pharmaceutical research community in practice.

Over the past few years, ML has evolved and its methodologies have improved as a result of breakthroughs in the field.(9-11) This is a good time to set to date the good practices of ML for their use at full potential in pharmacometrics, by exploiting their interpretability and taking advantage of their precision methods, while addressing today's increasingly complex problems. This paper has two objectives: first, to introduce the state-of-the-art methodologies and techniques of ML, including fine-tuning, grid search, cross-validation, Random Forest and XGBoost; then to show how to interpret extracted information using the two last methods. Three examples will be used to showcase the potential of these newfangled methodologies. With the first example, we will illustrate how to interpret the variable importance as a metric of Random Forest and XGBoost to strengthen our confidence in the covariate evaluation procedure during a classical population pharmacokinetic (PopPK) analysis. In the second example, we will present another way to estimate plasmatic drug concentration, a task that usually requires a nonlinear and mixed-effect model type. The third example presents a solution

to a type of non-linear relationships often encountered in clinical pharmacology, but different from traditional pharmacokinetic relationships.

Methods

Data preprocessing is constituted of an ensemble of steps to prepare the data for the training phase. It refers to transforming raw data into a structured data set ready to be processed by a model. The first step consists in an exploratory analysis of the data, using visualization tools and descriptive statistics, with the aim to clearly picture the information carried by the data set. This step can involve handling missing values and outliers. Decision trees are among the methods that can take care of missing values. Replacing the missing values with a central tendency measure, or completely deleting the outlier's observations is frequently chosen as solutions. The issue of missing values and outliers is not specific to ML.

The second step of data preprocessing is called variable scaling, which ensures to transform all the predictors to the same scale. Most ML methods are sensitive to the scales of their variables. When a method operates with Euclidian distance calculations or gradient descent optimization, all variables must be on the same scale to ensure reproducible results.(12) Otherwise, different numerical values of the same observation (i.e. 1000g, 1kg) have a different impact during the training phase of the model. Variables can be scaled using standardization or normalization, with the former giving a distribution of mean 0 and a standard deviation of 1, while the latter gives values between 0 and 1. (13) Trees-based methods work by recursive partitions and do not need a scaling step to work.

The purpose of training a model is to estimate the parameters which generate the best fits to the observations. However, hyperparameters are a special type of parameters that must be passed to the model before the estimation. They usually control the speed of learning and the complexity of the model. Since there is no way to identify the correct values for the hyperparameters from the beginning, one must hyperparametrise the space for reasonable values. The set of combinations searched for is represented in a grid. For a model of two hyperparameters A and B, with A taking three different values and B binary values, there is a total of 6 different models with a grid length of six. When the space of research is not too large,

an exhaustive grid search is made. Otherwise, the search is limited to few attempts chosen randomly through the grid. (14) The exhaustive/randomized grid search is the most widespread method to tune a model's hyperparameters.

The two objectives of a learning process are to estimate the parameters and to evaluate the performance of the chosen model, which correspond to training and testing a model, respectively. Prior to the training phase, one part of the data must be held out of the training, preventing a bias estimation. Moreover, the hold-out set, also called the validation set, gives only a single estimation of the real-world error of this model. The results depend on the validation set. To address these issues, the training data set could, for example, be split into 5 parts and the fifth part can be hold-out during the estimation of the parameters and used afterward to validate the model. The procedure can be repeated five times, so that each part is used as the validation set once. At the end, it produces an average estimation of the real error. Doing this, the amount of training data is maximized, providing more reliable estimates. This resampling method is known as 5-folds cross-validation, where folds can take any number between 3 to the number of observations in the data set. Hence, with this grid search example, 6 different models would go through the training process 5 times each, with a total of 30 trainings for the entire procedure. Finally, the smallest cross-validation error will indicate the model to be retained for the next steps of the procedure. Once the hyperparameters are known, the chosen model is trained again, but this time the training is performed on all the training data. (15)

Once the chosen model has completed the training process, it can be assessed. As previously mentioned, a different data set, from the one used in the training step, called test set, must serve the assessment process, in order to avoid estimation bias. Issues with a single hold-out set arise again, with the need for more than one estimation to ensure reliability of the estimated error. This is done using the nested cross-validation, where the entire data set is split into C parts and each part used as a test set once. The training procedure including the k-fold inner cross-validation is repeated for each outer C-fold. (9)

The evaluation of the predictive performance of regression models can be accomplished through error metrics such as mean error (ME), root mean square error (RMSE), mean absolute error (MAE), and R squared (r2).

$$ME = \frac{1}{N}\sum_{i=1}^{n}(\hat{y}_i - y_i) \quad (1)$$

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

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (3)$$

$$r^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (4)$$

The first metric (ME) evaluates the bias of the model while the other metrics evaluate its precision. Guidelines for pharmacometrics suggest choosing RMSE before MAE, since RMSE increases large error importance, just as required in a clinical setting. (17, 18)

Introduced in 1984 (19), Classification and Regression Trees (CART) is a technique that aims to split the predictor space into smaller regions to predict a label for a given observation. As the name suggests, it can be used both as a classifier and a regressor. The model can be represented by an upside-down tree, with the top node called the root, and leaves being at the bottom of the tree. To make a prediction, an observation will be processed into the tree, starting at the origin node by being assigned to the branch of the node for which it respects the conditions of the split. The observation will be going down over and over and will be assigned to one side or the other of all the encountered nodes until a specific region of the predictor space, called a leaf, is reached. Finally, the value of the leaf will be used as a prediction for this very observation.

The building of a tree, also known as the training phase, starts by choosing the composition of the first node. The predictor to be used as well as the value of the separation threshold are to

be determined. Each possible node's options will be tested. This process measures the impurity of a node which indicates how much the leaf prediction fits the observations. The option with the lowest impurity will be kept. With regression trees, the aim is to minimize the difference between the observed label and the prediction for all the observations using the Residual Sum of Squares (RSS). A pure leaf would contain only observations of the same class as its prediction. The building phase continues to add nodes in the tree until the leaves contain a certain quantity of observation. Moreover, there is usually a pruning step to verify the cost/benefit of each node in terms of the tree performance. The larger the tree, the better it adapts to training data, meaning a low bias. However, it usually means a harder time when facing non-training data and thus a larger variance. Despite restrictive rules and pruning techniques, decision trees still tend to overfit and have an unbalanced bias-variance trade-off when used for real-world data.

To address the CART issue, bagging techniques such as Random Forest have been proposed. (20) Bagging techniques deal with the high variance of the decision tree by merging the prediction of many trees, usually a couple of hundreds. Each tree uses a resampled subset of the data known as bag data set for training, which is a particular form of bootstrap. An additional step differentiates the Random Forest from the standard bagging techniques. Each tree takes only a subset of the predictors during its creation. The two last steps aim to introduce variability between trees and make sure each tree is different from each other. Otherwise the process of bagging is useless, and we will end up with a hundred of clones of the same decision tree. It is worth to notice that the Random Forest uses the out-of-bag set to get an estimation of the performance instead of the classical cross-validation.

Another solution to CART deficiency is Gradient Boosting Techniques. (21) Instead of building many fully grown trees in parallel as bagging techniques do, small trees are built sequentially. The idea here is to focus on the weakness of the previous tree by applying larger and lower weights on the misclassified and well-classified observations of the last tree, respectively. Among these techniques, Extreme Gradient Boosting, also known as XGBoost (11), proved to provide fast and state-of-the-art results in several situations.

Tree bagging and boosting techniques can identify the importance of the impact of each variable on the predictions. With the Random Forest, two types of measurements evaluate the

variable importance: the impurity-based measurement and the permutation-based measurement. (22) The impurity-based method examines the quality of each node associated with the variable. It follows the same principle as during the building phase. It measures the average decrease in the impurity that is attributable to the variable's nodes, and this is done within all the trees. For regression trees, it measures the reduction of the RSS. The permutation-based method quantifies the mean difference in the performance of the out-of-bag prediction; hence, a second measure of performance is made for each tree. This time, the out-of-bag set contains a shuffled version of the variable. This measurement looks at the difference in the prediction between the set of data with the shuffled variable and the set of data with the untouched variable. A shuffled variable that does not cause a drop in the performance implies that the tree does not attribute much importance to it. For regression trees, the measurement is called reduction of the RSS of the out-of-bag prediction. For the XGBoost, there is also an impurity-based measure called the Gain. This measure differs from the other impurity measurements because of the unique way trees are built with Gradient boosting methods. Despite this, the Gain measure still calculates the average quality of each node associated to the variable. (11)

Results

A one-compartment model with an extravascular administration and a linear elimination is considered here. The drug concentration C(t) is given by:

$$C((t))=(k_a\ F\ C_0)/(V\ (k_a-k_e)\ )\ e^{(-k_e\ t)}-e^{(-k_a\ t)} \qquad (5)$$

Where ka and ke are the absorption and elimination rate constants, respectively; V is the volume of distribution; F is the bioavailability. The exponential and additive models were used for the inter-individual variability and residual error, respectively. Model parameters are assumed to follow log-normal distributions. A covariate is associated to each parameter involved, with the addition of a dummy one. The first covariate, cov1, is acting on V and follows a normal distribution. The second covariate, cov2, acts on ke and has a uniform distribution with a power centered error model. The third covariate, cov3, is a binary variable with an

exponential switch model effect that modulates the absorption constant. The fourth covariate, cov4, is a dummy one that was not included in the population PK model. It was included as a control covariate. Using a population of n subjects, the model parameters of the i-th individual are thus expressed as:

(V_i= θ_V (〖cov1〗_i/〖cov1〗_Ref)^(θ_cov1) e^(η_i) (6))

(k_(e,i)= θ_(k_e) (〖cov2〗_i/〖cov2〗_Ref)^(θ_cov2) e^(η_i) (7))

(k_(a,i)= θ_(k_a) e^((1 θ_cov3)) e^(η_i) (8))

For each parameter: η_i~N(0,ω_ ^2) is a normal distribution of mean 0 and variance ω_ ^2.

A data set containing ID, TIME, DV, cov1, cov2, cov3, and cov4 was simulated using the pharmacokinetic model mentioned above. No information of the model was used during the subsequent steps, a condition that mimics real-life situations. The simulation parameters are given in Table 1.

Population size 1000

Dose 100

Time 0-24

Bioavailability 0.85

Covariate 1 N(70,20)

Covariate 2 U(10,50)

Covariate 3 0,1

Covariate 4 N(10,2)

Fixed effects

θ ka 0.8

θ ke 0.1

θ	V	1
θ	cov1	1.8
θ	cov2	2
θ	cov3	1.5
θ	cov4	-
Random effect of level 1		
ω	ka	0.81
ω	kel	0.65
ω	V	0.78
Random effect of level 2		
Sigma		0.17

Table 1: PopPK simulation parameters

Part 1: Preselection of covariates in a pharmacometrics model

ML methods can be used for the preselection of covariates in a pharmacometrics model. To start a ML analysis, a specific task needs to be defined. This task is formulated here under the following question: which covariate(s) influences each of the PopPK model parameters (Ke, V, Ka)? To answer this question, the data set must include columns of independent variables and a column for the dependent variable. At this point, the data set only contains the independent variables (covariates), hence the dependent variable must be added. The individual PopPK parameter Ke can be extracted from the data after calculating the elimination haft-time (t½) as described in equation 9, but the other two (V, Ka) are only accessible through surrogates parameters (sP), represented by the maximal concentration (Cmax), and time of maximal drug concentration (Tmax), since Cmax is primarily influenced by V, as illustrated in Figure 3, and that Tmax could be defined by Ka and Ke, as described in equation 10. (21, 22) Here, the idea is as the following: if a covariate affects a variable (Cmax, Tmax) linked to a parameter, the covariate should affect the PopPK model parameter (V, Ka) as well. Then, the covariate should be tested into the covariate selection procedure of the corresponding PopPK model parameter. The

simplicity of the proposition lies in the fact that ML methods do not need to be provided with the structure of the relationship between covariates, PK parameters or their substitutes.

$$t_{1/2} = \ln 2 / k_e \quad (9)$$

Figure 1: Effect of the volume of distribution on the cmax

$$T_{max} = \ln\left(k_a - \ln(k_e)\right) / (k_a - k_e) \quad (10)$$

A Random Forest was trained, and two different metrics were extracted for feature importance, namely the Gini metric and the permutation metric. Additionally, an XGBoost was trained and provided the gain metric. The Caret R package was used to rescale the variable importance on a range from 0 to 100, with the latter being the maximum score. (16) All results are reported in Tables 1-3, for Ke, Cmax and Tmax, respectively. The question regarding the covariates that affect the elimination haft-life t½ was assessed using the Random Forest and the XGBoost. All three metrics identified cov2 as the covariate with a maximal importance of 100, indicating that it has a great impact on Ke, as opposed to the remaining covariates. Hence, cov2 should be included as a covariate for ke during the covariate selection step. The results are in line with what was expected.

This process was repeated for Cmax and Tmax, giving rise to similar results. Indeed, the covariates-parameter relationship of the PopPK model was reflected in the importance given to the covariates for the prediction of their respective sP. Compared with the other covariates, cov1 was found important for the prediction of Cmax, with a maximal score of 100, hence affecting the volume of distribution as well. The results in Table 3 indicate that cov3 and cov2 are important for the prediction of Tmax. However, the three metrics gave higher importance for cov3. According to the formula 10, Ka should have more impact on Tmax, if we do not assume a flip-fop kinetics (i.e we assume that Ka >> ke). Also, cov2 has already been identified to possibly be linked to the elimination process through ke. The last two observations suggest

that cov3 could be the covariate affecting the absorption process and ka. Nevertheless, both cov2 and cov3 should be tested on Tmax during the selection step to confirm the findings. The results correspond to the covariate effects of the simulated PopPK model.

ke	Impurity	Permutation	Gain
cov2 (ke)	100.00	100.00	100.00
cov1 (v)	9.87	3.40	4.11
cov4 (none)	5.77	0.00	3.83
cov3-1 (ka)	0.00	0.16	0.00

cmax	Impurity	Permutation	Gain
cov1 (v)	100.00	100.00	100.00
cov2 (ke)	22.08	8.07	5.08
cov4 (none)	15.38	0.00	2.18
cvo3-1 (ka)	0.00	0.55	0.00

Tmax	Impurity	Permutation	Gain
cov3-1 (ka)	100.00	100.0	100.00
cov2 (ke)	62.93	40.9	72.34
cov4 (none)	1.93	1.02	10.71
cov1 (v)	0.00	0.00	0.00

Table 2 : Variable importance scores for the Ke, Cmax and Tmax

Part 2: Estimation of plasmatic drug concentration

Here, the task was to estimate the plasmatic concentration using the identified covariates (dependent variable, DV). The same pharmacokinetic model is used here. The individual dose was picked for each subject from a pool of possible doses, with a sequence of 60, 120, and 180 units of dose. The data set contained the following variables: ID, TIME, DOSE, DV, cov1, cov2, cov3, and cov4. For each subject, half of the 24 sampling time points were randomly selected to mimic a sparse and unequal sampling. Each observation was processed independently during

training, but the training/evaluation split was done at the subject level to avoid data leakage in the evaluation set.

An exploratory analysis rejected no outliers and revealed no missing values into the data set.

We trained a Random forest and an XGBoost using the Caret R package, which provides a default grid for the tuning of the hyperparameters for the Random forest and an XGBoost. (16) An exhaustive grid search was performed for the hyperparameters. The resampling techniques consisted of an out-of-bag bootstrap with the default 25 repetitions for the Random Forest, and 5-fold cross-validation for the XGBoost. The assessment was made by a 5-fold cross-validation outer loop to capture variability, resulting in a nested cross-validation. The performance of the regression was evaluated by the ME, RMSE, MAE, and r squared.

Metric (SD)	Random Forest	XGBoost
ME	49.778(59.62)	64.21(94.57)
RMSE	13.51(17.61)	16.95(21.72)
MAE	49.78(59.62)	64.22(94.56)
R squared	0.90(0.06)	0.83(0.05)

Table 3: Error metrics for the predicted plasmatic concentration

Part 3: Application in clinical pharmacology

Another dataset was built to study a classical situation in clinical pharmacology. The dependant variable represents a typical effect, such as the risk to be ill, to develop severe complications after being infected with a virus, or to suffer from adverse effects of a specific treatment, etc. The dependant variable is supposed to be continuous, for which a regression analysis will be applied. The independent variables consist in the characteristics of the subjects, and could be demographic such as weight, sex, or age, or medical information such as the presence of concomitant disease or a genetic profile. The objective is to predict the individual effect knowing the characteristics of each subject.

A thousand of subjects were simulated, each one being defined by four characteristics, Xi, i=1, …, 4. The first and fourth characteristics have normal distributions with averages of 70 and 10, and variances of 20 and 2, respectively. The second one is uniformly distributed with a range between 10 and 90. The third variable takes binary values. Several conditions were used to

generate the relationship between the predictors and the dependant variable. Starting with a base risk of 8% for every subject, each of the four characteristics is associated with an additional risk. For a subject that has his first and second characteristic values, X1 and X2, higher than 90 and 70, his effect increases by 5% and 15%, respectively. If the third variable X3 is 1 rather than 0, the effect increases by 10%. X4 being a dummy variable, it did not affect the individual effect. Finally, an effect of 3% (unexplained) was included to add some variability in the data. This additional effect was randomly imposed on 10% of subjects. Table 3 summarises these characteristic-risk relationships.

Base effect	8%
If X1 > 90	+5%
If X2 > 70	+15%
If X3 = 1	+10%
X4	+0%
Unexplained variability	+3%

Table 4: The individual risk associated to each characteristic.

The exploratory analysis revealed no outliers or missing values into the data set. Also, no collinearity was found between the independent variables. We trained a linear model as a baseline, a Random Forest, and an XGBoost with the Caret R package. (16) The training and assessment decision were the same as in part 2. The decision trees based methods outperform the linear model. These results were expected due to the nonlinear relationship involved. The Random Forest and the XGBoost have a smaller bias compared to the linear regression represented by the ME as well as a smaller performance expressed in RMSE and MAE. The results are summarised in Table 4.

Metric (SD)	Random Forest	XGBoost	Linear regression
ME	6.94e-4 (1.62e-3)	2.07e-4(1.14e-3)	-5.27(4.86e-3)
RMSE	1.17e-2(1.00e-3)	1.66e-2(1.34e-2)	4.65e-2(2.56e-3)
MAE	0.01 (5.84e-4)	0.01(6.39e-4)	0.03(2.29e-3)
R squared	0.96(7.10e-3)	0.96(7.27e-3)	0.69(4.12e-2)

Table 5:Error metrics for the predicted individual risk

Discussion

In 1959, Arthur Samuel, a pioneer in the domain, said: “Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed”. By definition, a program is learning when its performance for a task is improving with experience. (13) ML’s main application is under supervised learning, where learning is from a system involving independent and dependent variables and statistical learning methods are used. (22) Statistical learning methods are used for two purposes: inference and prediction. A task is driven by inference when the interest is to understand the way the dependent variable is affected by the independent variables. In such cases, the preference is given to a method that provides good interpretability. When the goal is achieving a fair estimation of the dependent variable, methods with good prediction accuracy are preferred. More than often, a method achieves good accuracy because of its flexibility. A model is flexible when it has few restrictions in the range of shapes allowed to estimate the dependent variable. The linear regression is an example of an inflexible method because it is restricted to linear functions to estimate the dependent variable. On the other hand, multilayer perceptron or neural networks are highly flexible but much harder to interpret. As a rule of thumb, the interpretability of a method decreases as the flexibility increases. This concept is referred to as the trade-off between prediction accuracy and model interpretability. (15)

Previous research aiming to implement ML methods into the covariate evaluation of PopPK modeling have been opting for decision trees and specific type of lasso regression, which are known for their high interpretability.(6, 25, 26) Unfortunately, these two methods can have a disability in prediction accuracy. Moreover, the Lasso regression can be too inflexible when facing a nonlinear relationship and decision trees can be too flexible leading to a perfect function to express the relationship. This lack of restriction may lead to overfitting.(15) Previous studies dedicated to the prediction of plasmatic concentrations or other drug-related topics, as optimal dosing or adverse effects, have proposed neural networks and Support Vector Machine for these purposes.(2-5, 8) Whereas, the latter methods may have good predictive power, their complexity makes them unattractive when time comes to interpret the results. Therefore, in the current work, we were looking for one method (or family of methods) to accommodate the two

objectives, which are to familiarize the pharmacometrics community with the fundamental concepts and the promising methods of ML from one side, and demonstrate, through typical examples, how to extract information from the models and to interpret the obtained results. Methods derived from decision trees, such as Random Forests and XGBoost present the merit of great prediction accuracy while keeping good model interpretability. Despite the easy training process of Random Forest, it has an impressive performance in a variety of different situations.(22) For its part, XGBoost is known as state-of-the-art technology and as being over-represented in the winning solutions of many prestigious ML competitions.(11) Choosing these two methods over others was meant to serve our two objectives.

In the first part, the goal was to demonstrate the possibility to implement Random Forest and XGBoost as a tool to strengthen the reliability of the preselection step of the covariate evaluation, which is a critical step in the building process of a PopPK model. It consists of a preselection phase, followed by an assessment phase where these covariates are tested to determine whether they must be added or dropped from the model. The decision of adding or dropping a covariate from a model can lead to different outcomes. In a perfect situation, all the possible models should be tested. However, covariate evaluation could be time-consuming and real-world limitations prevent from testing every possibility during the evaluation. Getting a good idea of the relationships between covariates and parameters beforehand could prevent from testing every possible covariate-parameter combination. It is therefore important to have an efficient tool during the preselection. (27) To have an initial understanding of the covariate-parameter combination, one must rely on prior clinical knowledge and graphical diagnostic plots. These plots consist of empirical Bayesian estimates of parameters versus covariates and can be of use to detect trends between the covariates and the parameters. However, there is a reluctance to blindly using these graphics since they can be misleading in situations where an important shrinkage in the distribution of the empirical Bayesian estimates occurs. (28) In presence of shrinkage, which is the phenomenon that occurs when a model is over-parameterized for the amount of information contained in the data. (32) it has been suggested to test all covariates, and to analyze the conditional-weighted residuals diagnostic plot or use other types of plots, such as simulation-based diagnostic plots. (29-31) In the current work, we

proposed an additional way for the preselection of the covariates, whether or not there is a shrinkage. An algorithm using ML methods such as Random Forest and XGBoost could be used alongside prior clinical knowledge and the typical diagnostic plots to strengthen the confidence in the preselection step and provide more efficiency to the subsequent steps, by testing only covariates with reasonable potential. Random Forests and XGBoost have been able to successfully identify the presence of a relation between a pharmacokinetic parameter and its covariate.

In the second and third parts, we intended to enlighten on common ML procedures and apply them to a specific pharmacokinetic modeling task as well as a typical (abstract but plausible) pharmacology situation. Random Forests and XGBoost were tested through two examples to demonstrate how ML predictive power could be useful in pharmacokinetic modeling and drug therapy. In the second part, both methods were found capable to adapt and provide an accurate prediction of plasmatic drug concentration. This gives insight into the fact that Random Forests and XGBoost can grasp the specific pharmacokinetic relationship just like the differential equation methods derived from the population pharmacokinetic approach. The third part provided an overview of the advantages of using the ML procedure for complex and non-linear tasks so often encountered in drug therapy.

ML methods apply to a wide variety of problems that come from several fields. The intentions behind their origins and the needs that generated their development are different from the reasons behind traditional pharmacometric methods. Therefore, ML methods cannot be taken out of context, and blindly used to fill the needs of pharmacometrics. This attitude could lead to frustrating and unsuccessful conclusions. A deep understanding of the philosophy behind ML, as well as the acquirement of technical skills is needed to benefit from their coveted benefits. Our results demonstrate the potential of the ML approach in clinical pharmacology analysis. We hope that these results will motivate the leveraging ML techniques in the field, to evolve side by side to other pharmacometrics methodologies for the patients benefit.

References

1. Brier ME, Zurada JM, Aronoff GR. Neural network predicted peak and trough gentamicin concentrations. Pharm Res. 1995;12(3):406-12.

2. Nestorov IS, Hadjitodorov ST, Petrov I, Rowland M. Empirical versus mechanistic modelling: comparison of an artificial neural network to a mechanistically based model for quantitative structure pharmacokinetic relationships of a homologous series of barbiturates. AAPS PharmSci. 1999;1(4):E17.

3. Chow HH, Tolle KM, Roe DJ, Elsberry V, Chen H. Application of neural networks to population pharmacokinetic data analysis. J Pharm Sci. 1997;86(7):840-5.

4. Poynton MR, Choi BM, Kim YM, Park IS, Noh GJ, Hong SO, et al. Machine learning methods applied to pharmacokinetic modelling of remifentanil in healthy volunteers: a multi-method comparison. J Int Med Res. 2009;37(6):1680-91.

5. Tang J, Liu R, Zhang YL, Liu MZ, Hu YF, Shao MJ, et al. Application of Machine-Learning Models to Predict Tacrolimus Stable Dose in Renal Transplant Recipients. Sci Rep. 2017;7:42192.

6. Koch G, Pfister M, Daunhawer I, Wilbaux M, Wellmann S, Vogt JE. Pharmacometrics and Machine Learning Partner to Advance Clinical Data Analysis. Clinical Pharmacology & Therapeutics. 2020;107(4):926-33.

7. Cosgun E, Limdi NA, Duarte CW. High-dimensional pharmacogenetic prediction of a continuous trait using machine learning techniques with application to warfarin dose prediction in African Americans. Bioinformatics. 2011;27(10):1384-9.

8. Kang SH, Poynton MR, Kim KM, Lee H, Kim DH, Lee SH, et al. Population pharmacokinetic and pharmacodynamic models of remifentanil in healthy volunteers using artificial neural network analysis. Br J Clin Pharmacol. 2007;64(1):3-13.

9. Cawley GC, Talbot NLC. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. 2010;11:2079–107.

10. Krstajic D, Buturovic LJ, Leahy DE, Thomas S. Cross-validation pitfalls when selecting and assessing regression and classification models. Journal of Cheminformatics. 2014;6(1):10.

11. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; San Francisco, California, USA: Association for Computing Machinery; 2016. p. 785–94.

12. Grus J. Data Science from Scratch: First Principles with Python: O'Reilly Media, Inc.; 2015.

13. Gron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems: O'Reilly Media, Inc.; 2017.

14. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. 2011;12(null %J J. Mach. Learn. Res.):2825–30.

15. James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning: with Applications in R: Springer Publishing Company, Incorporated; 2014.

16. Kuhn M. Building Predictive Models in R Using the caret Package. 2008. 2008;28(5):26 %J Journal of Statistical Software.

17. Sheiner LB, Beal SL. Some suggestions for measuring predictive performance. J Pharmacokinet Biopharm. 1981;9(4):503-12.

18. Sarem S. Limited Sampling Strategies for Estimation of Cyclosporine Exposure in Pediatric Hematopoietic Stem Cell Transplant Recipients:Methodological Improvement and Introduction of Sampling Time Deviation Analysis. Montréal: Université de Montréal; 2014.

19. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees: CRC press; 1984.

20. Breiman L. Random forests. Machine learning. 2001;45(1):5-32.

21. Friedman JHJAos. Greedy function approximation: a gradient boosting machine. 2001:1189-232.

22. Hastie T, Tibshirani R, Friedman JH. The Elements of Statistical Learning: Data Mining, Inference, and Prediction: Springer; 2009.

23. Jambhekar SS, Breen PJ, editors. Basic Pharmacokinetics. 2 ed: Pharmaceutical Press; 2012.

24. Beaulieu P, editor. Précis de pharmacologie. Du fondamental à la clinique: PRESSES DE L'UNIVERSITE DE MONTREAL; 2011.

25. Ette EI, Ludden TM. Population pharmacokinetic modeling: the importance of informative graphics. Pharm Res. 1995;12(12):1845-55.

26. Haem E, Harling K, Ayatollahi SM, Zare N, Karlsson MO. Adjusted adaptive Lasso for covariate model-building in nonlinear mixed-effect pharmacokinetic models. J Pharmacokinet Pharmacodyn. 2017;44(1):55-66.

27. Owen JS, Fiedler-Kelly J. Model Building. Introduction to Population Pharmacokinetic / Pharmacodynamic Analysis with Nonlinear Mixed Effects Models2014. p. 138-72.
28. Savic RM, Karlsson MO. Importance of shrinkage in empirical bayes estimates for diagnostics: problems and solutions. Aaps j. 2009;11(3):558-69.
29. Joerger M. Covariate Pharmacokinetic Model Building in Oncology and its Potential Clinical Relevance. The AAPS Journal. 2012;14(1):119-32.
30. Hooker AC, Staatz CE, Karlsson MO. Conditional weighted residuals (CWRES): a model diagnostic for the FOCE method. Pharm Res. 2007;24(12):2187-97.
31. Bourguignon L, Ducher M, Matanza D, Bleyzac N, Uhart M, Odouard E, et al. The value of population pharmacokinetics and simulation for postmarketing safety evaluation of dosing guidelines for drugs with a narrow therapeutic index: buflomedil as a case study. Fundam Clin Pharmacol. 2012;26(2):279-85.
32. webdev. What is Shrinkage? Certara; 2020 [Available from: https://www.certara.com /knowledge-base/what-is-shrinkage/.

Chapitre 4 – Discussion

Application aux sciences pharmaceutiques

Les sciences pharmaceutiques regroupent tous les domaines qui couvrent le développement du médicament, des premières phases d'identification de la cible thérapeutique à la surveillance de l'utilisation à grande échelle des médicaments dans la population à la suite de leur mise en marché, en passant bien entendu par les études cliniques. Les besoins analytiques des sciences pharmaceutiques sont omniprésents. Les prises de décisions doivent être basées sur des données probantes. Chaque phase du développement implique des prises de décision sur un grand nombre de problèmes, nécessitant de l'information clé pour assurer le succès durant le développement du médicament. Ces prises de décision requièrent une bonne connaissance des avantages et des inconvénients des traitements considérés, des mécanismes biologiques impliqués et de la gravité de l'état du patient au regard de sa condition i.e. maladie ou infection. (25)

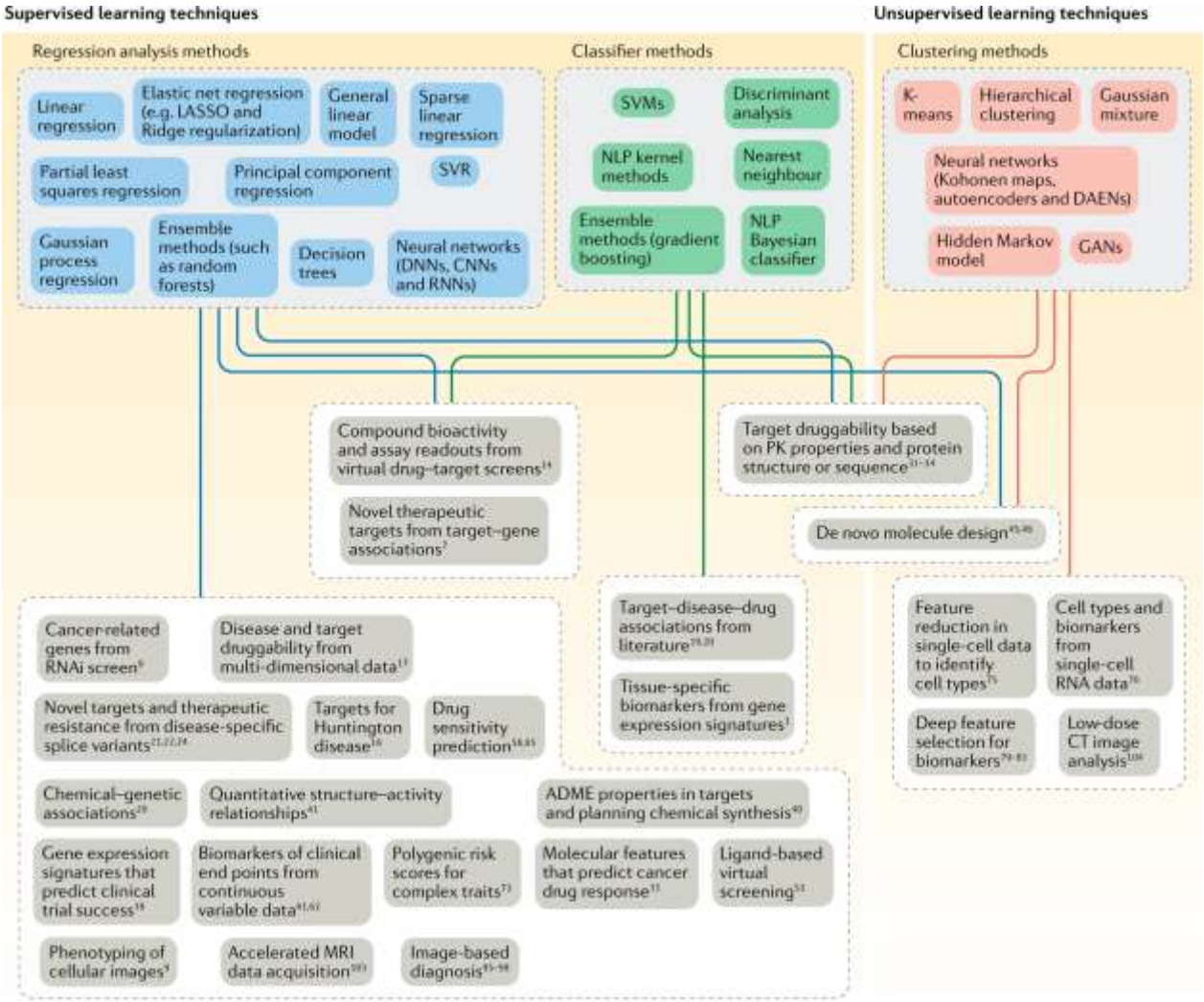

Figure 18. – Utilisation de l'apprentissage automatique dans le processus de développement de médicaments. (26)

Les connaissances nécessaires à l'obtention des réponses sont acquises grâce à différents types d'outils mathématiques. On parle ici de modélisation mathématique et statistique. La modélisation s'applique à une variété importante de problèmes pharmaceutiques comme le montre la figure 18 et le tableau 3. Les outils proposés par l'apprentissage automatique peuvent être appliqués à l'ensemble des stages de développement du médicament dans le but d'aider à surmonter les défis actuels auxquels fait face le domaine. Ces défis se traduisent en 3 objectifs principaux : empêcher l'explosion des coûts, accélérer le processus du développement et

réduire le grand taux d'attrition des médicaments candidats présents à chaque étape du développement.(25)

Tableau 3. – Exemples d'applications pharmaceutiques reliés à chaque branche de l'apprentissage automatique.

Apprentissage automatique		
Apprentissage supervisé	**Apprentissage non-supervisé**	**Apprentissage par renforcement**
Classification	**Analyse de regroupement**	**Prise de décision**
• Aider au diagnostic de maladie	• Découvrir des sous-populations de patients pour une maladie	• Concevoir et optimiser *in vitro* des nouveaux principes actifs
Régression	**Sélection des variables**	**Exécution de la décision**
• Prédire l'efficacité d'un traitement • Prédire des propriétés ADMET	• Aider à l'identification de cibles thérapeutiques	• Supporter la conception d'étude clinique basée sur l'identification de biomarqueurs prédictifs

Durant les dernières années, les tentatives pour intégrer des méthodes d'apprentissage automatique en sciences pharmaceutiques se sont multipliées. En voici quelques exemples. Certaines de ces tentatives ont visé l'évaluation des covariables lors de la modélisation PopPK avec des arbres de décision et une régression avec régularisation L1, aussi appelé régression LASSO. (27-29) D'autres chercheurs se sont consacrés à la prédiction des concentrations plasmatiques ou d'autres entités relatives aux médicaments, comme le dosage optimal ou les effets indésirables, à l'aide de réseaux de neurones et des machines à vecteurs de soutien.(1, 30-34) Des forêts aléatoires de survie, des réseaux de neurones et des machines à vecteurs de soutien ont donné des résultats semblables au modèle de Cox lors d'une étude de survie. (35, 36) L'apprentissage par renforcement a permis l'optimisation d'une stratégie de traitement selon une balance efficacité-toxicité au niveau individuel et au niveau de la cohorte, en s'appuyant sur l'état de santé des patients basé sur la mesure d'un biomarqueur.(37-39)

Un domaine qui bénéficie beaucoup des outils de l'apprentissage automatique est l'identification de cible thérapeutique.(26) L'identification de cible thérapeutique se pense sur la pharmacabilité des protéines comme les récepteurs cellulaires ou les récepteurs nucléaires. En identifiant les facteurs caractéristiques des cibles à succès du passé, il est possible trancher sur le potentiel des cibles sous investigation. Par exemple, une étude de 2014 a identifié de nouvelles cibles pour les cancers du sein, du pancréas et des ovaires à l'aide d'une variété donnée génomique en classant des protéines selon leur pharmacabilité.(40) Les caractéristiques utilisées étaient l'essentialité du gène, l'expression de l'ARN messager, l'occurrence de mutations et d'autres données et d'autres informateurs génomiques. On identifie avec cet exemple, une classification binaire avec un ensemble de données à haute dimension. Ce type de tâche est type du genre d'analyse fait en apprentissage automatique.

Une application qu'on n'imagine pas immédiatement, c'est le support à la revue de littérature. En début de développent, la revue de littérature est l'une des principales sources d'informations sur une maladie et la recherche qui y gravite. Tous peuvent convenir de la lourdeur du processus. Cependant, une récente avancée dans le domaine du traitement automatique du langage naturel (NLP), permet d'automatiser le processus. En 2017, un nouveau type de modèle, appelé *Transforer* est proposé pour analyser le langage naturel. (41) Avec ce genre de modèle, on est capable de faire de la réponse de questions ou de la classification de pairs de phrases. Pour une optique de recherche translationnelle, une équipe de chercheur de l'université Pompeu Fabra de Barcelone on développé un système NLP capable d'extraire les associations entre les gênes les maladies à partir de résumés de la banque d'articles Medline.(42)

Après avoir identifié une cible thérapeutique, il faut désigner puis optimiser la structure des molécules. Encore ici, on tente d'identifier des caractéristiques spécifiques à un groupe gagnant pour ensuite prédire le succès d'un nouvel élément. Par exemple, il existe plusieurs recherches du type suivant prédire la sélectivité et la spécificité de nouvelles molécules générées à partir d'une molécule *lead* à l'aide de l'apprentissage profond. Une étude de 2017 a établi comme supérieur leur modèle d'apprentissage profond par rapport aux techniques statistiques classiques. (43) Pour ce genre de tâche où la quantité de données disponible est au contraire

très réduite, il est possible d'utiliser un modèle du type apprentissage en un coup comme il a été fait en 2018 pour identifier le mécanisme de liaison des opioïdes aux récepteurs μ.(44)

Un autre intérêt de recherche à la frontière de l'apprentissage automatique et des sciences pharmaceutique c'est la représentation et la prédiction des interactions molécules-molécules ou molécules-cibles basées sur des techniques de similarité et la représentation par graphs.(45-50)Leur utilisation demande moins de ressources et est plus rapide que les techniques traditionnels.(51)

Les exemples les plus visuels restent l'utilisation de modèles d'apprentissage automatique profond pour le diagnostic de différentes maladies via l'analyse d'images médicales. Par exemple, plusieurs travaux ont pour sujet le diagnostic de différents cancers (i.e. poumons, sein, etc.) ou d'activités tumorales.(52-58)

Une application étonnante qu'on peut faire, c'est l'extraction automatique d'information pertinente directement des appareils médicaux personnels. Ça pourra utiliser pour récolter des données en temps réelle pour caractériser l'adhérence au traitement ou carrément l'évolution de la maladie. (59)

En terminant, comme précédemment mentionné, les solutions proposées par l'apprentissage automatique s'appliquent également aux questions de nature quantitatives des études cliniques. C'est ce qu'il a été présent au cours de ce mémoire. Actuellement ces analyses sont généralement faites avec les outils du domaine de la pharmacométrie. La pharmacométrie est la science qui interprète et décrit la pharmacologie de manière quantitative.(60) Autrement dit, c'est la discipline qui permet de quantifier les informations relatives aux traitements, aux maladies et aux essais cliniques afin d'obtenir un développement efficace et faciliter les décisions réglementaires.(61) Elle a comme aspiration de permettre un traitement optimal et individualisé. Le principal type de modèles utilisés en pharmacométrie est le modèle non-linéaire à effets mixtes, essentiellement utilisé pour des analyses populationnelles en pharmacocinétique, pharmacodynamie et en progression de maladie. (62)

Retour

Les méthodes d'apprentissage automatique ont suscité beaucoup d'intérêt en recherche. Elles peuvent être considérées avantageuses par rapport aux modèles traditionnels ou mécanistiques. En fait, les méthodes d'apprentissage automatique ne sont pas basées sur des hypothèses mécanistiques comme le sont, en général, les modèles en pharmacométrie. Bien que les méthodes d'apprentissage automatique puissent être considérées par certains comme des méthodes insondables de type boîte noire, elles ont l'avantage de simplifier le processus étant donné qu'aucune connaissance biologique préalable n'est requise, et ainsi réduire le biais humain associé. En outre, les méthodes d'apprentissage automatique sont connues pour leur capacité à traiter simultanément de nombreuses variables avec succès et à gérer facilement des relations complexes et non linéaires, ce qui en fait des méthodes potentiellement adéquates pour le domaine biopharmaceutique. Malheureusement, l'interprétabilité est un aspect important dans les sciences médicales et semble faire défaut aux méthodes d'apprentissage automatique. Cela semble empêcher leur adoption plus systématique par les pharmacométriciens. Il est à noter par contre, que tout récemment, l'apprentissage automatique a été introduit lors de la Conférence Américaine de pharmacométrie (ACoP) en 2018, et que l'intérêt pour ces méthodes semble croitre dans cette communauté scientifique. Il reste cependant encore un long chemin à parcourir avant que l'apprentissage automatique ne soit pleinement adopté par les chercheurs du domaine pharmaceutique.

Au cours des dernières années, l'apprentissage automatique a évolué et ses méthodologies se sont améliorées grâce à des percées. (16, 24)Le moment est opportun de mettre à jour et de présenter les bonnes pratiques de l'apprentissage automatique pour qu'elles soient utilisées à leur plein potentiel en pharmacométrie, en exploitant leur interprétabilité et en tirant partie de leur capacité de précision, tout en s'attaquant aux problèmes de plus en plus complexes d'aujourd'hui.

Ce travail avait deux objectifs. D'abord, présenter des méthodologies et des techniques de pointe de l'apprentissage automatique, notamment le réglage fin, la recherche par grille, la

validation croisée, les forêts aléatoires et l'XGBoost. Ensuite, présenter la façon d'extraire et d'interpréter des informations en utilisant les deux dernières méthodes. Trois exemples ont été utilisés pour mettre en lumière le potentiel de ces méthodes. Le premier exemple servait à montrer la façon d'interpréter l'importance des variables obtenues avec des forêts aléatoires et l'XGBoost dans le but de renforcer la confiance de la décision durant la procédure de présélection des covariables d'une classique analyse de pharmacocinétique de population. Le second exemple présentait une autre façon d'estimer la concentration plasmatique des médicaments, une tâche qui nécessite généralement un modèle de type non linéaire à effets mixtes. Le troisième exemple fournissait une solution à un type de relation non linéaire souvent rencontré en pharmacologie clinique, mais différent des relations pharmacocinétiques traditionnelles.

Les méthodes dérivées des arbres de décision, telles que les forêts aléatoires et l'XGBoost, présentent l'avantage d'une grande précision de prédiction tout en conservant une bonne interprétabilité. Malgré la facilité d'apprentissage des forêts aléatoires, leur performance est impressionnante, et ce, dans une grande variété de situations différentes. Pour sa part, l'XGBoost est reconnu comme une technologie de pointe qui est surreprésentée parmi les solutions gagnantes à de nombreux prestigieux concours de d'apprentissage automatique.(24) Choisir ces deux méthodes plutôt que d'autres, c'était un moyen de servir nos deux objectifs.

Dans la première partie, il s'agissait de démontrer la possibilité de mettre en œuvre des forêts aléatoires et l'XGBoost comme outil pour renforcer la fiabilité de l'étape de présélection et de l'évaluation des covariables. L'évaluation des covariables est une étape critique dans le processus de construction d'un modèle PopPK. Elle consiste en une phase de présélection, suivie d'une phase d'évaluation au cours de laquelle ces covariables sont testées pour déterminer si elles doivent être ajoutées ou supprimées du modèle. La décision d'ajouter ou de supprimer une covariable d'un modèle peut conduire à des résultats différents. Dans une situation parfaite, toutes les combinaisons paramètres-PK/covariables possibles doivent être testées. Cependant, l'évaluation des covariables peut prendre beaucoup de temps et les limitations intrinsèques au domaine commercial ou médical empêchent de tester toutes les possibilités pendant l'évaluation. Avoir une bonne idée des relations entre les covariables et les

paramètres avant l'évaluation pourrait éviter de tester toutes les combinaisons possibles de covariables et de paramètres. Il est donc important de disposer d'un outil efficace lors de la présélection. Pour avoir une première compréhension de la combinaison covariables-paramètres, il faut s'appuyer sur des connaissances cliniques préalables et des graphiques de diagnostic. Ces graphiques consistent en des estimations bayésiennes empiriques des paramètres par rapport aux covariables, et peuvent être utiles pour détecter les tendances entre les covariables et les paramètres. Cependant, il y a une réticence à utiliser aveuglément ces graphiques car ils peuvent être trompeurs. Dans le cadre des travaux actuels, nous avons proposé une méthode supplémentaire de présélection des covariables. Un algorithme utilisant des méthodes de l'apprentissage automatique telles qu'une forêt aléatoire et l'XGBoost peut être utilisé pour soutenir la présélection des covariables. L'objectif durant le développement d'un modèle PopPK étant de seulement tester les covariables ayant indiqué un potentiel durant la présélection pour permet de maximiser l'efficience de l'étape de sélection des covariables. Les forêts aléatoires et l'XGBoost ont pu identifier avec succès la présence d'une relation entre un paramètre pharmacocinétique et sa covariable.

Dans la deuxième et la troisième partie, nous avons voulu mettre en lumière les bonnes pratiques de l'apprentissage automatique et les appliquer à une tâche caractéristique de la modélisation pharmacocinétique ainsi qu'à une situation typique de pharmacologie clinique qu'on pourrait qualifier de situation abstraite mais plausible. Les deux exemples ont permis de mettre en œuvre les méthodes de forêts aléatoires et d'XGBoost afin de démontrer le pouvoir prédictif de l'apprentissage automatique en sciences pharmaceutiques. Dans la seconde partie, les deux méthodes se sont révélées aptes à s'adapter à la relation présente dans les données et de fournir une prédiction précise de la variable réponse. Cela donne un aperçu du fait que les forêts aléatoires et l'XGBoost peuvent expliquer une relation pharmacocinétique spécifique tout comme le fait l'approche de pharmacocinétique de population basée sur les équations différentielles. La troisième partie a donné un aperçu des avantages de l'utilisation de la procédure de l'apprentissage automatique pour des relations complexes et non linéaires si souvent rencontrées en pharmacothérapie, et pour lesquelles ces supposées relations nous sont inconnues au moment de la modélisation.

Conclusion

En terminant, il faut savoir se poser les bonnes questions pour utiliser les méthodes de l'apprentissage automatique. Il faut prendre le temps de bien cerner la tâche et bien définir ses besoins. Le contexte est motivé par la prédiction plutôt que par l'interprétation. Le phénomène ou le système d'intérêt n'a pas de structure ou de mécanisme bien défini. Si ces deux conditions sont satisfaites, il sera pertinent de profiter des méthodes de l'apprentissage automatique.

Les méthodes de l'apprentissage automatique s'appliquent à une grande variété de problèmes qui proviennent de plusieurs domaines. Les intentions à l'origine de leur création et les besoins qui ont généré leur développement sont différents des raisons qui sous-tendent les méthodes traditionnelles de la pharmacométrie. Par conséquent, les méthodes de l'apprentissage automatique ne peuvent pas être sorties de leur contexte ni utilisées aveuglément pour répondre aux besoins des sciences pharmaceutiques. Ce genre d'attitude pourrait mener à des conclusions frustrantes et infructueuses. Une compréhension approfondie de la philosophie derrière l'apprentissage automatique est nécessaire, ainsi que l'acquisition de compétences techniques pour bénéficier de ses avantages tant convoités. Nos résultats démontrent le potentiel de l'apprentissage automatique pour l'analyse de données de pharmacologie clinique. Nous espérons que ces résultats motiveront l'utilisation des techniques d'apprentissage automatique sur le terrain, afin d'évoluer côte à côte vers d'autres méthodologies pharmacométriques pour le bénéfice des patients.

Références bibliographiques

1. Nestorov IS, Hadjitodorov ST, Petrov I, Rowland M. Empirical versus mechanistic modelling: comparison of an artificial neural network to a mechanistically based model for quantitative structure pharmacokinetic relationships of a homologous series of barbiturates. AAPS PharmSci. 1999;1(4):E17.
2. Hastie T, Tibshirani R, Friedman JH. The Elements of Statistical Learning: Data Mining, Inference, and Prediction: Springer; 2009.
3. Breiman L. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). Statistical Science. 2001;16.
4. Owen JS, Fiedler-Kelly J. Population Model Concepts and Terminology. Introduction to Population Pharmacokinetic / Pharmacodynamic Analysis with Nonlinear Mixed Effects Models2014. p. 9-27.

5. Owen JS, Fiedler-Kelly J. Model Building. Introduction to Population Pharmacokinetic / Pharmacodynamic Analysis with Nonlinear Mixed Effects Models2014. p. 138-72.
6. Gron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems: O'Reilly Media, Inc.; 2017.
7. Vapnik VN. The nature of statistical learning theory: Springer-Verlag; 1995.
8. Mohri M, Rostamizadeh A, Talwalkar A. Foundations of Machine Learning: The MIT Press; 2012.
9. James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning: with Applications in R: Springer Publishing Company, Incorporated; 2014.
10. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. 2011;12(null %J J. Mach. Learn. Res.):2825–30.
11. Sheiner LB, Beal SL. Some suggestions for measuring predictive performance. J Pharmacokinet Biopharm. 1981;9(4):503-12.
12. Sarem S. Limited Sample Strategy for Estimation of Cyclosporine Exposure in Pediatric Hematopoietic Stem Cell Transplant Recipients. Montréal: Université de Montréal; 2014.
13. Grus J. Data Science from Scratch: First Principles with Python: O'Reilly Media, Inc.; 2015.
14. Ozcan T, Basturk AJCC. Human action recognition with deep learning and structural optimization using a hybrid heuristic algorithm. 2020:1-14.
15. Cawley GC, Talbot NLC. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. 2010;11:2079–107.
16. Krstajic D, Buturovic LJ, Leahy DE, Thomas S. Cross-validation pitfalls when selecting and assessing regression and classification models. Journal of Cheminformatics. 2014;6(1):10.
17. Kuhn M. Building Predictive Models in R Using the caret Package. 2008. 2008;28(5):26 %J Journal of Statistical Software.
18. Breiman L, Friedman J, Stone CJ, Olshen RA. Classification and regression trees: CRC press; 1984.
19. Louppe G, Wehenkel L, Sutera A, Geurts P. Understanding variable importances in forests of randomized trees. Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1; Lake Tahoe, Nevada: Curran Associates Inc.; 2013. p. 431–9.
20. Breiman LJMl. Bagging predictors. 1996;24(2):123-40.
21. Breiman L. Random forests. Machine learning. 2001;45(1):5-32.
22. Friedman JHJAos. Greedy function approximation: a gradient boosting machine. 2001:1189-232.
23. Schapire RE. The strength of weak learnability. Machine Learning. 1990;5(2):197-227.
24. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; San Francisco, California, USA: Association for Computing Machinery; 2016. p. 785–94.
25. Mak K-K, Pichika MRJDdt. Artificial intelligence in drug development: present status and future prospects. 2019;24 3:773-80.
26. Vamathevan J, Clark D, Czodrowski P, Dunham I, Ferran E, Lee G, et al. Applications of machine learning in drug discovery and development. Nature Reviews Drug Discovery. 2019;18(6):463-77.
27. Koch G, Pfister M, Daunhawer I, Wilbaux M, Wellmann S, Vogt JE. Pharmacometrics and Machine Learning Partner to Advance Clinical Data Analysis. Clinical Pharmacology & Therapeutics. 2020;107(4):926-33.
28. Ette EI, Ludden TM. Population pharmacokinetic modeling: the importance of informative graphics. Pharm Res. 1995;12(12):1845-55.
29. Haem E, Harling K, Ayatollahi SM, Zare N, Karlsson MO. Adjusted adaptive Lasso for covariate model-building in nonlinear mixed-effect pharmacokinetic models. J Pharmacokinet Pharmacodyn. 2017;44(1):55-66.
30. Chow HH, Tolle KM, Roe DJ, Elsberry V, Chen H. Application of neural networks to population pharmacokinetic data analysis. J Pharm Sci. 1997;86(7):840-5.

31. Brier ME, Zurada JM, Aronoff GR. Neural network predicted peak and trough gentamicin concentrations. Pharm Res. 1995;12(3):406-12.
32. Poynton MR, Choi BM, Kim YM, Park IS, Noh GJ, Hong SO, et al. Machine learning methods applied to pharmacokinetic modelling of remifentanil in healthy volunteers: a multi-method comparison. J Int Med Res. 2009;37(6):1680-91.
33. Tang J, Liu R, Zhang YL, Liu MZ, Hu YF, Shao MJ, et al. Application of Machine-Learning Models to Predict Tacrolimus Stable Dose in Renal Transplant Recipients. Sci Rep. 2017;7:42192.
34. Cosgun E, Limdi NA, Duarte CW. High-dimensional pharmacogenetic prediction of a continuous trait using machine learning techniques with application to warfarin dose prediction in African Americans. Bioinformatics. 2011;27(10):1384-9.
35. Gong X, Hu M, Zhao L. Big Data Toolsets to Pharmacometrics: Application of Machine Learning for Time-to-Event Analysis. Clin Transl Sci. 2018;11(3):305-11.
36. Katzman JL, Shaham U, Cloninger A, Bates J, Jiang T, Kluger Y. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. BMC Medical Research Methodology. 2018;18(1):24.
37. Yauney G, Shah P. Reinforcement Learning with Action-Derived Rewards for Chemotherapy and Clinical Trial Dosing Regimen Selection. In: Finale D-V, Jim F, Ken J, David K, Rajesh R, Byron W, et al., editors. Proceedings of the 3rd Machine Learning for Healthcare Conference; Proceedings of Machine Learning Research: PMLR %J Proceedings of Machine Learning Research; 2018. p. 161--226.
38. Houy N, Le Grand F. Optimal dynamic regimens with artificial intelligence: The case of temozolomide. PLoS One. 2018;13(6):e0199076-e.
39. Zhao Y, Kosorok MR, Zeng D. Reinforcement learning design for cancer clinical trials. Stat Med. 2009;28(26):3294-315.
40. Jeon J, Nim S, Teyra J, Datti A, Wrana JL, Sidhu SS, et al. A systematic approach to identify novel cancer drug targets using machine learning, inhibitor design and high-throughput screening. Genome Medicine. 2014;6(7):57.
41. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems; Long Beach, California, USA: Curran Associates Inc.; 2017. p. 6000–10.
42. Bravo À, Piñero J, Queralt-Rosinach N, Rautschka M, Furlong LI. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. BMC Bioinformatics. 2015;16(1):55.
43. Ramsundar B, Liu B, Wu Z, Verras A, Tudor M, Sheridan RP, et al. Is Multitask Deep Learning Practical for Pharma? J Chem Inf Model. 2017;57(8):2068-76.
44. Barati Farimani A, Feinberg E, Pande VJBJ. Binding Pathway of Opiates to μ-Opioid Receptors Revealed by Machine Learning. 2018;114:62a.
45. Thafar MA, Olayan RS, Ashoor H, Albaradei S, Bajic VB, Gao X, et al. DTiGEMS+: drug–target interaction prediction using graph embedding, graph mining, and similarity-based techniques. Journal of Cheminformatics. 2020;12(1):44.
46. Mohamed SK, Nováček V, Nounu A. Discovering protein drug targets using knowledge graph embeddings. Bioinformatics. 2020;36(2):603-10.
47. Yang F, Fan K, Song D, Lin H. Graph-based prediction of Protein-protein interactions with attributed signed graph embedding. BMC Bioinformatics. 2020;21(1):323.
48. Xiao Z, Deng Y. Graph embedding-based novel protein interaction prediction via higher-order graph convolutional network. PLoS One. 2020;15(9):e0238915.
49. Huang K, Xiao C, Glass LM, Zitnik M, Sun J. SkipGNN: predicting molecular interactions with skip-graph networks. Scientific Reports. 2020;10(1):21092.

50. Mohamed S, Novácek V, Nounu A. Discovering Protein Drug Targets Using Knowledge Graph Embeddings. Bioinformatics (Oxford, England). 2019;36.
51. Su X-R, You Z-H, Hu L, Huang Y-A, Wang Y, Yi H-C. An Efficient Computational Model for Large-Scale Prediction of Protein–Protein Interactions Based on Accurate and Scalable Graph Embedding. 2021;12(165).
52. Sharma H, Zerbe N, Klempert I, Hellwich O, Hufnagl P. Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology. Comput Med Imaging Graph. 2017;61:2-13.
53. Korbar B, Olofson A, Miraflor A, Nicka C, Suriawinata M, Torresani L, et al. Deep learning for classification of colorectal polyps on whole-slide images. 2017;8(1):30-.
54. Bychkov D, Linder N, Turkki R, Nordling S, Kovanen PE, Verrill C, et al. Deep learning based tissue analysis predicts outcome in colorectal cancer. Scientific Reports. 2018;8(1):3395.
55. Cruz-Roa A, Gilmore H, Basavanhally A, Feldman M, Ganesan S, Shih NNC, et al. Accurate and reproducible invasive breast cancer detection in whole-slide images: A Deep Learning approach for quantifying tumor extent. Scientific Reports. 2017;7(1):46450.
56. Romo-Bucheli D, Janowczyk A, Gilmore H, Romero E, Madabhushi A. Automated Tubule Nuclei Quantification and Correlation with Oncotype DX risk categories in ER+ Breast Cancer Whole Slide Images. Scientific Reports. 2016;6(1):32706.
57. Saltz J, Gupta R, Hou L, Kurc T, Singh P, Nguyen V, et al. Spatial Organization and Molecular Correlation of Tumor-Infiltrating Lymphocytes Using Deep Learning on Pathology Images. Cell Rep. 2018;23(1):181-93.e7.
58. Romo-Bucheli D, Janowczyk A, Gilmore H, Romero E, Madabhushi A. A deep learning based strategy for identifying and associating mitotic activity with gene expression derived risk categories in estrogen receptor positive breast cancers. Cytometry A. 2017;91(6):566-73.
59. Yang Z, Huang Y, Jiang Y, Sun Y, Zhang Y-J, Luo P. Clinical Assistant Diagnosis for Electronic Medical Record Based on Convolutional Neural Network. Scientific Reports. 2018;8(1):6329.
60. Huang XH, Li J. Pharmacometrics: The Science of Quantitative Pharmacology: Am J Pharm Educ. 2007 Aug 15;71(4):75.
61. Administration USFD. Division of Pharmacometrics: Center for Drug Evaluation and Research; 2018 /28 Nov 2020 [Available from: https://www.fda.gov/about-fda/center-drug-evaluation-and-research-cder/division-pharmacometrics.
62. Owen JS, Fiedler-Kelly J. The Practice of Pharmacometrics. Introduction to Population Pharmacokinetic / Pharmacodynamic Analysis with Nonlinear Mixed Effects Models2014. p. 1-8.