TP2 : Expliquer le comportement d'un modèle prédisant le risque de décès chez des patients insuffisants cardiaques (2H)

Contexte clinique

L'interprétabilité des modèles revêt une importance cruciale dans le contexte clinique, notamment pour garantir la compréhension des décisions prises, favoriser la confiance des praticiens, et assurer une prise de décision éclairée. Dans le cadre de la santé, où des choix peuvent influencer directement la vie des patients, la transparence des modèles est une exigence essentielle. L'interprétabilité se réfère à la capacité à comprendre et interpréter les décisions prises par un modèle.

Au cours de ce TP, nous allons tenter d'expliquer comment un modèle Random Forest prédit le décès chez les patients atteints d'insuffisance cardiaque. Ce modèle a été évalué avec une concordance équilibrée de 0.82+/-0.05, une sensibilité de 0.81+/- 0.04 et une spécificité de 0.83 +/- 0.08. Pour cela, il utilise 12 variables, décrites dans le Tableau 1. Plusieurs méthodes seront utilisées.

Tableau 1 Description des variables du jeu de données

Variable	Description
age	Age du patient
anemia	Diminution des globules rouges ou de l'hémoglobine
creatinine_phosphokinase	Taux de l'enzyme CPK dans le sang (mcg/L)
diabetes	Présence de diabète
ejection_fraction	Pourcentage de sang quittant le cœur à chaque contraction
high_blood_pressure	Présence d'hypertension
platelets	Plaquettes dans le sang (kiloplaquettes/mL)
serum_creatinine	Taux de créatinine sérique dans le sang (mg/dL)
serum_sodium	Taux de sodium sérique dans le sang (mEq/L)
sex	Sexe du patient
smoking	Consommation de tabac
time	Période de suivi (jours)

I. Les ressources

Récupérer l'archive *TP2_explicabilite_modeles.zip* sur Moodle. Elle contient :

- ***0_TP_explicabilite_visualisation_et.py*** : le script à compléter qui vous permettra de répondre aux questions de la partie II.
- ***1_TP_explicabilite_feat_importance_et.py*** : le script à compléter qui vous permettra de répondre aux questions des parties III.
- ***2_TP_explicabilite_pdp_et.py*** : le script à compléter qui vous permettra de répondre aux questions de la partie IV.
- ***3_TP_explicabilite _shapley_plots_et.py*** : le script à compléter qui vous permettra de répondre aux questions de la partie V.
- ***helping_functions.py*** : un ensemble de fonctions prêtes à l'emploi et à utiliser pour certaines questions de ce TP.

- ***heart_failure_dataset_test.csv*** : le jeu de données étudié.

II. Visualisation du modèle (15 minutes)

Les méthodes à base d'arbres sont couramment utilisées car leurs structures sont intrinsèquement compréhensibles. En effet, nous pouvons visualiser un arbre de décision et comprendre le cheminement menant à une décision. Cela est aussi vrai pour les Random Forest, lorsque ceux-ci se composent de peu d'arbres… Le modèle que nous étudions est composé de 20 arbres. Nous pouvons observer ces arbres 1 par 1 grâce la fonction plot_tree() de scikit-learn.

1. Suivez les instructions dans le script ***0_TP_explicabilite_visualisation_et.py***.
2. Observez les 20 arbres générés dans votre dossier de projet, dans le répertoire arbres_random_forest.
3. Que pouvez-vous en dire ? Est-ce que leurs structures sont cohérentes avec les hyperparamètres retenus pour le modèle ?

III. Importance des variables (15 minutes)

Il est difficile d'observer le comportement de 20 arbres simultanément. Pour cette raison, il existe des manières d'expliquer les décisions du modèle de manière **globale**. Ici, nous allons nous intéresser à l'importance des variables.

Feature importance de Random Forest

Une première méthode est intrinsèque au Random Forest : ***feature_importance_***. Pour cette méthode, les importances des variables sont calculées en évaluant la somme de leur impact sur la qualité des divisions effectuées à chaque nœud de chaque arbre. Cet impact est mesuré par la diminution de l'impureté, qui peut être quantifiée à l'aide de mesures telles que l'indice de Gini ou l'entropie (utilisée lors de l'apprentissage du modèle).

4. Suivez les instructions dans le script ***1_TP_explicabilite_feat_importance_et.py***.
5. Quelles variables sont les plus importantes pour le modèle ?

Attention : l'interprétation des importances des caractéristiques dans un modèle Random Forest offre un aperçu de la contribution de chaque variable à la prédiction globale, ce qui peut être utile pour la sélection de caractéristiques et la compréhension du modèle. Cependant, il est essentiel de les considérer comme des indicateurs relatifs plutôt que comme des mesures absolues de l'importance.

Feature permutation importance

Une seconde méthode s'appelle la **feature permutation importance**. La permutation d'importance, évalue l'importance d'une variable en permutant aléatoirement ses valeurs et en mesurant l'impact sur la performance du modèle (mesurée par une métrique de performance spécifique, comme la précision, le rappel, etc.). Une valeur élevée indique que la permutation aléatoire des valeurs de cette variable a un impact significatif sur la métrique de performance évaluée, suggérant ainsi que la variable joue un rôle crucial dans les prédictions du modèle. En revanche, une valeur proche de zéro suggère une moindre influence de la variable sur la performance du modèle.

1. Suivez les instructions dans le script ***1_TP_explicabilite_feat_importance_et.py***.

2. Quelles variables sont les plus importantes pour le modèle ? Quels sont les variables les moins importantes ?
3. Retenez les 3 variables identifiées comme importantes à la fois par le Random Forest et par la permutation.

Attention : Il est important de noter que la permutation importance évalue l'impact **univarié** d'une variable, ne prenant en compte que les variations individuelles de cette variable sur la performance du modèle.

IV. Partial Dependence Plots (PDP) (30 minutes)

Les graphiques de dépendance partielle (**Partial Dependence Plot**, PDP) offrent un moyen de visualiser **l'influence globale** d'une ou deux variables sur les prédictions d'un modèle. Ces graphiques fournissent un aperçu de la relation entre ces variables et les prédictions du modèle. Pour cela, une grille de valeurs couvrant la plage de la variable (ou des deux) est créée, et des prédictions sont effectuées en utilisant chaque valeur de la grille à la variable étudiée pour toutes (ou une partie) des observations du jeu de données, tandis que les autres variables restent constantes. La moyenne des prédictions obtenues est ensuite calculée et utilisée pour générer le graphique de dépendance partielle.

L'interprétation du graphique se fait en observant comment la prédiction évolue en fonction des valeurs de la variable ou des deux variables. Si le graphique montre une relation linéaire, non linéaire, ou s'il y a des seuils critiques, cela peut fournir des informations importantes sur la manière dont le modèle utilise ces caractéristiques pour faire des prédictions.

1. Observer les partial dépendance plots pour les 3 variables les plus influentes identifiées dans la partie III (une à une, puis deux à deux) - suivez les instructions dans le script ***2_TP_explicabilite_pdp_et.py***.
2. Quelle(s) interprétation(s) pouvez-vous en faire ?
3. Observer les interactions avec l'âge des patients. Cela vous parait-t 'il cohérent ?

V. SHAP (SHapley Additive exPlanations) (45 minutes)

Les **valeurs de Shapley**, issues de la théorie des jeux, évaluent l'importance de chaque variable dans un modèle prédictif en quantifiant leur contribution marginale à la prédiction. Dans le cadre de l'apprentissage automatique, les valeurs des variables d'une observation sont considérées comme les membres d'une coalition. Les valeurs de Shapley permettent de déterminer équitablement la répartition du "gain" (i.e., la valeur prédite) entre ces variables, éclairant ainsi sur l'importance de la contribution de chaque variable à la prédiction. Cela s'inspire du scénario où une coalition de joueurs collabore pour atteindre un bénéfice commun, et les valeurs de Shapley offrent une solution pour répartir équitablement les bénéfices entre les "joueurs", qu'ils soient des valeurs de variables individuelles ou des ensembles de valeurs de variables.

Global

1. Générer la visualisation summary grâce à la librairie shap. Suivez les instructions dans le script ***3_TP_explicabilite _shapley_plots_et.py.***
2. Quelle interprétation pouvez-vous en faire ?
3. Générer la visualisation « beeswarm » grâce à la librairie shap. Suivez les instructions dans le ***3_TP_explicabilite _shapley_plots_et.py***.

4. Quelle interprétation pouvez-vous en faire ?

Local

Jusqu'ici, nous avons étudié des méthodes d'analyse du comportement global du modèle. Il existe aussi des méthodes qui permettent cette analyse à **un niveau local**, c'est-à-dire pour une ou pour un sous ensemble d'observations. C'est aussi possible avec les valeurs de Shapley.

1. Générer la visualisation waterfall pour une observation au hasard de à la librairie shap. Suivez les instructions dans le script ***3_TP_explicabilite _shapley_plots_et.py***.
2. Sélectionner des observations illustrant des Vrais positifs et des Vrai négatifs et observer le comportement du modèle. Quelle interprétation pouvez-vous en faire ?
3. Le modèle n'est pas parfait. Vous pouvez utiliser cette visualisation pour mieux comprendre les erreurs faites par le modèle.
    a. Sélectionner des erreurs de type Faux Négatifs et analyser
    b. Sélectionner des erreurs de type Faux Positifs et analyser

VI. Discussion (15 minutes)

Nous avons parcouru un ensemble de méthodes permettant de mieux comprendre la manière dont un modèle renvoie ses prédictions. Ces méthodes permettent à la fois de développer de meilleurs modèles mais aussi de définir précisément le domaine de fonctionnement d'un outil.

1. A partir de toutes vos observations, pensez-vous qu'il est possible de créer un modèle plus compact ? Si oui, quel en serait l'intérêt ?

VII. Pour aller plus loin…

Si vous avez suivi le TP1 « Importance de la préparation des données pour le développement d'un outil de prédiction de la survie des patients souffrant d'insuffisance cardiaque ». Reprenez-le en utilisant un ensemble de variables plus limité et observer les performances d'un modèle plus compact.

D'autres méthodes sont aussi disponibles pour décrire le fonctionnement des modèles (e.g., LIME) et n'ont pas été étudiées dans ce TP. Christoph Molnar nous en offre un panorama dans son livre Interpretable Machine Learning - A Guide for Making Black Box Models Explainable [1].

Ce TP a été construit à partir du jeu de données exploité par Chicco et al. dans leur étude : *Davide Chicco, Giuseppe Jurman: Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making 20, 16 (2020).* https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data

Ressources sur l'explicabilité :

[1] https://christophm.github.io/interpretable-ml-book/

[2] https://juanitorduz.github.io/interpretable_ml/

[3] https://www.statcan.gc.ca/fr/science-donnees/reseau/apprentissage-explicable

[4] https://scikit-learn.org/stable/modules/partial_dependence.html

[5] https://www.aidancooper.co.uk/a-non-technical-guide-to-interpreting-shap-analyses/

Annexe : exemples d'interprétation des graphes

Partial dependence plots

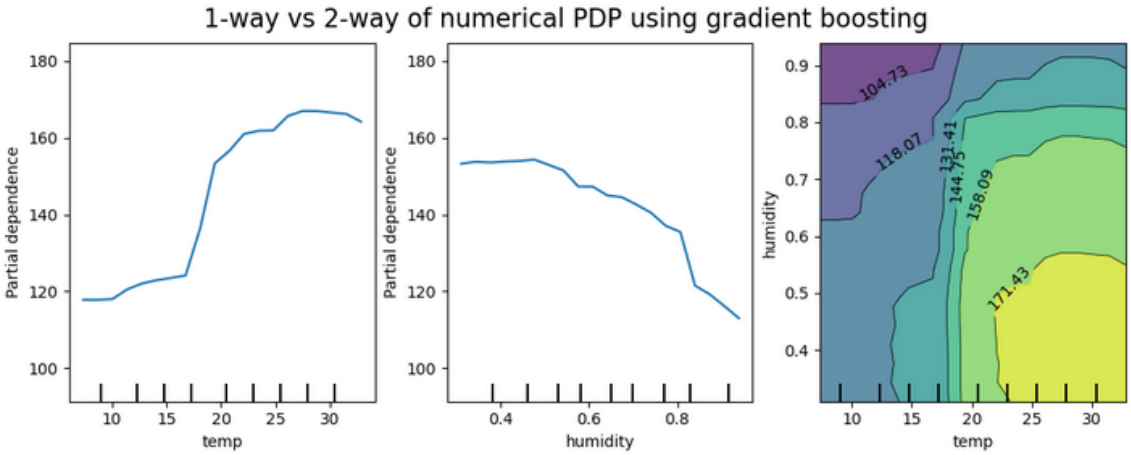

Figure 1 Exemple issu de scikit-learn pour un modèle de prédiction du nombre de locations de vélo en fonction des variables température et humidité

Les PDP univarié nous renseignent sur l'interaction entre la prédiction et une variable d'entrée particulière (par exemple, linéaire, non linéaire). Le graphique de gauche dans la figure ci-dessus montre l'effet de la température sur le nombre de locations de vélos ; nous pouvons clairement voir qu'une température plus élevée est liée à un nombre plus élevé de locations de vélos. De manière similaire, nous pourrions analyser l'effet de l'humidité sur le nombre de locations de vélos (graphique du milieu).

Les PDP avec deux variables d'entrée d'intérêt montrent les interactions entre les deux caractéristiques. Par exemple, le PDP à deux variables dans la figure ci-dessus montre la dépendance du nombre de locations de vélos aux valeurs conjointes de la température et de l'humidité. Nous pouvons clairement observer une interaction entre les deux caractéristiques : avec une température supérieure à 20 degrés Celsius, principalement l'humidité a un impact fort sur le nombre de locations de vélos. Pour des températures plus basses, à la fois la température et l'humidité ont un impact sur le nombre de locations de vélos.

Summary plots with shap

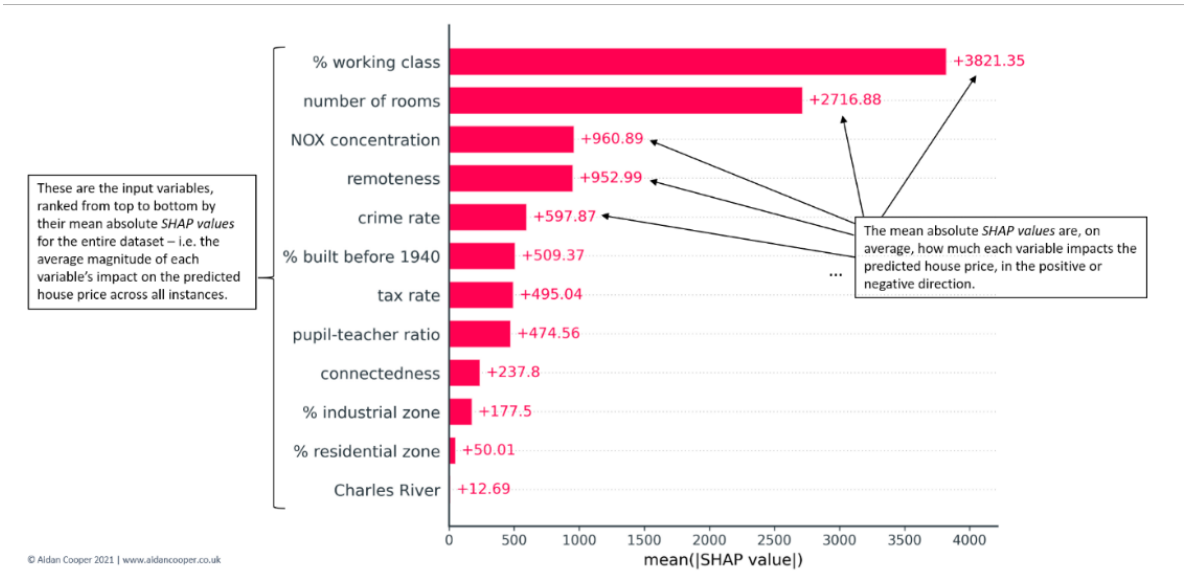

Figure 2 Exemple de summary plot obtenu avec la librairie shap dans le cas d'une régression (estimation du prix de vente d'une maison), issu de [5]

Le point de départ le plus simple pour l'interprétation globale avec SHAP consiste à examiner la valeur SHAP moyenne absolue pour chaque caractéristique à travers l'ensemble des données. Cela quantifie, en moyenne, l'ampleur (positive ou négative) de la contribution de chaque caractéristique aux prix des maisons prédits. Les caractéristiques avec des valeurs SHAP moyennes absolues plus élevées sont plus influentes. Les valeurs SHAP moyennes absolues sont essentiellement un remplacement direct pour des mesures de l'importance des caractéristiques plus traditionnelles, mais présentent deux avantages clés :

- Les valeurs SHAP moyennes absolues sont plus théoriquement rigoureuses et se rapportent à quelles caractéristiques impactent le plus les prédictions (ce qui est généralement ce qui nous intéresse). Les importances traditionnelles des caractéristiques sont mesurées de manière plus abstraite et spécifique à l'algorithme, et sont déterminées par la mesure de l'amélioration de la performance prédictive du modèle apportée par chaque caractéristique.
- Les valeurs SHAP moyennes absolues ont des unités intuitives - dans cet exemple, elles sont quantifiées en dollars, comme la variable cible. Les importances des caractéristiques sont souvent exprimées dans des unités contre-intuitives basées sur des concepts complexes tels que les impuretés des nœuds dans les algorithmes d'arbre.

Beeswarm plots with Shap

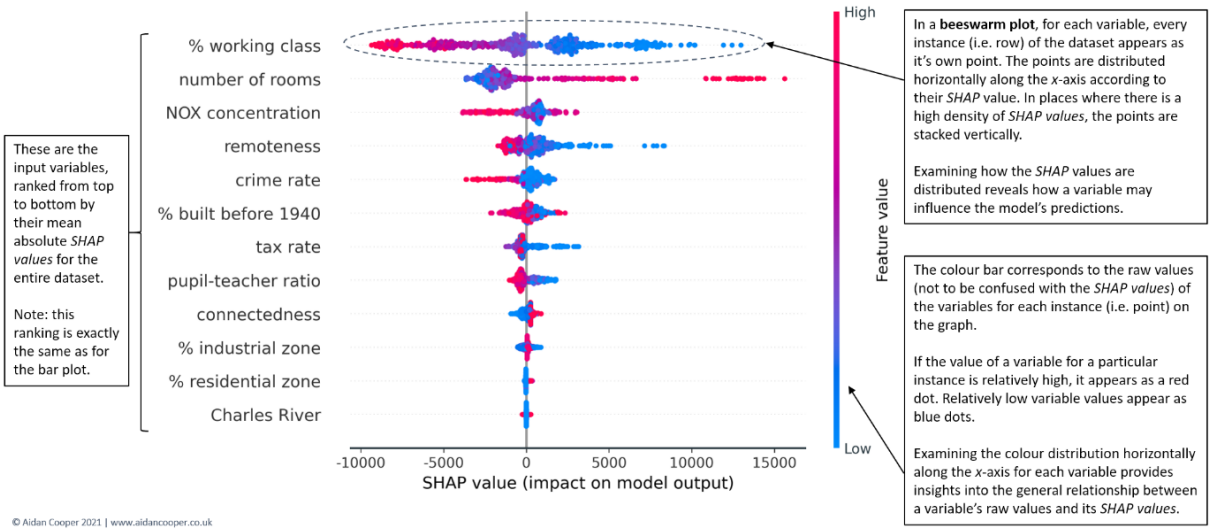

Figure 3 Exemple de beeswarm plot obtenu avec la librairie shap dans le cas d'une régression (estimation du prix de vente d'une maison), issu de [5]

Les graphiques « Beeswarm » sont une représentation plus complexe et riche que les graphiques « summary » en informations des valeurs SHAP, révélant non seulement l'importance relative des caractéristiques, mais aussi leurs relations réelles avec le résultat prédit.

Waterfall pour une observation

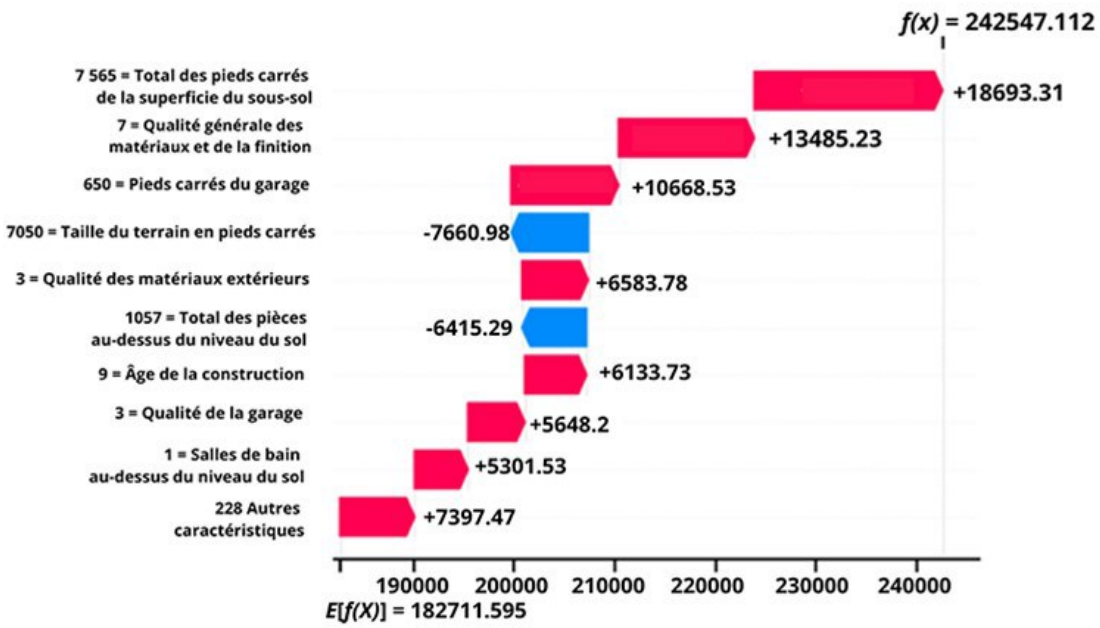

Figure 4 Exemple de représentation Waterfall pour un modèle de régression (prédiction du prix de vente d'une maison), issu de [3].

Un graphique « Waterfall » montrant l'évolution prévue de la valeur d'une maison en fonction de caractéristiques telles que la taille de la surface habitable, la taille du garage, la superficie en pieds carrés, la salle de bain, etc. La sortie du modèle pour cette prédiction varie en fonction de chaque caractéristique pour obtenir une valeur prédite complète de la maison. L'axe y contient la liste des caractéristiques et leur valeur associée. L'axe x représente la valeur attendue de la sortie du modèle, E[f(X) = 182711.595. Les caractéristiques, et leur valeur, sont listées avec leur contribution positive ou négative comme suit :

7 565 = Total des pieds carrés de la superficie du sous-sol +18693.31
7 = Qualité Générale des matériaux et de la finition +13485.23
650 = Pieds carrés du garage +10668.53
7050 = Taille du terrain en pieds carrés -7660.98
3 = Qualité des matériaux extérieurs +6583.78
1057 = Total des pièces au-dessus du niveau du sol -6415.29
9 = Âge de la construction +6133.73
3 = Qualité de la garage +5648.2
1 = Salles de bain au-dessus du niveau du sol +5301.53
228 autres caractéristiques +7397.47