Université Claude Bernard, Lyon 1

Projet TER

Interprétabilité/Explicabilité en Apprentissage Automatique

Auteurs :
Himanshu GUPTA
Yassine ANAS

Superviseur :
M.Gilles COHEN

4 mai 2023

Table des matières

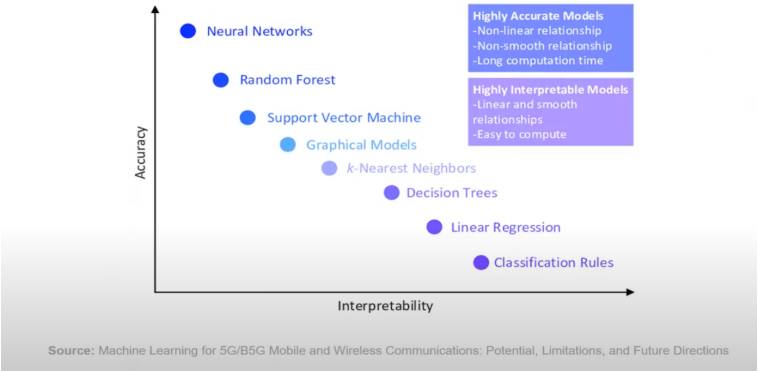
1 Introduction

Le Machine Learning tient maintenant une grande place dans les processus décisionnels des entreprises car son degré de précision a augmenté de manière accrue ces dernières décennies.

Toutefois, l'opacité des algorithmes de Machine Learning (ML) et de Deep Learning soulèvent de nos jours des questions d'ordre éthique et juridique. Le besoin de confiance et de transparence est ainsi très présent et très prisé.

Pour ces raisons, l'interprétabilité et l'explicabilité des modèles font dorénavant partie intégrante du métier des Data Scientists qui doivent s'efforcer de convaincre leurs clients et utilisateurs de l'acceptabilité du raisonnement de leur modèle. Heureusement pour eux de nombreuses méthodes existent maintenant pour faciliter ce travail.

Les modèles de machine learning (ML) sont de plus en plus complexes. En effet, un modèle sophistiqué (de boosting XGBoost ou de deep learning) permet généralement d'aboutir à des prédictions plus précises qu'un modèle simple (de régression linéaire ou arbre de décision). Il existe cependant un compromis entre la performance d'un modèle et son interprétabilité : ce qu'un modèle gagne en précision, il le perd en interprétabilité (et inversement).



1.1 Problematique

— **Pourquoi les modèles de machine learning peuvent-ils être expliqués de manière à ce que les utilisateurs puissent comprendre comment les prédictions sont effectuées ?**

— **Comment les modèles de machine learning peuvent-ils être rendus plus interprétables pour les utilisateurs non-experts ?**

— **Quelles sont les différences entre LIME et SHAP ? Comment choisir entre les deux pour une étude préçise ?**

— **Comment peut-on mesurer l'interprétabilité et l'explicabilité des modèles de machine learning, et quelles sont les limites de ces mesures ?**

De prime abord, nous allons voir en quoi consiste l'interprétablité et l'explicabilité des modèles de modèles d'apprentissage automatique. Puis nous allons exposer deux méthodes permettant de les dégager. Ensuite nous mettrons en application ces deux méthodes sur un dataset particulier (Diabète). Enfin nous verrons comment choisir entre les deux méthodes.

2 Interprétabilité et Explicabilité [5]

2.1 Qu'est-ce que l'Interprétabilité et l'Explicabilité dans le domaine du Machine Learning ?

L'interprétabilité consiste à pouvoir comprendre comment le modèle fonctionne en fournissant des informations sur le modèle de Machine Learning ainsi que sur les données utilisées. L'interprétabilité est dédiée aux experts en ML ou des données.) [5]

L'explicabilité consiste à pouvoir expliquer pourquoi le modèle a donné telle prédiction en fournissant une information dans un format sémantique complet et accessible à un data scientist.)

2.2 Objectifs de l'Interprétabilité et de l'Explicabilité dans nos modèles

Confiance et transparence : L'interprétabilité et l'explicabilité contribuent à renforcer la confiance dans le modèle en rendant son processus décisionnel transparent et compréhensible. Ceci est particulièrement important dans les domaines à fort enjeu où les conséquences des décisions d'un modèle peuvent être importantes.

Débogage et amélioration : les modèles interprétables permettent aux praticiens de comprendre pourquoi un modèle fait une prédiction particulière, ce qui facilite l'identification et la correction de tout biais ou erreur dans le modèle.

Conformité : dans de nombreux domaines, il existe des exigences légales et éthiques pour que les processus de prise de décision soient transparents et explicables.

Meilleure prise de décision : lorsque les décisions prises par un modèle d'apprentissage automatique sont compréhensibles et transparentes, cela peut aider à améliorer la qualité des décisions prises et à accroître leur alignement sur les valeurs et les objectifs humains.

2.3 Quelques Exemples

Exemple 1. Supposons qu'une banque ait développé un modèle d'apprentissage automatique pour évaluer le risque de crédit pour les demandes de prêt. Le modèle est formé sur un ensemble de données de données historiques sur les prêts et utilise des caractéristiques telles que le revenu, le statut d'emploi, le pointage de crédit et le ratio dette / revenu pour faire des prédictions. Le modèle atteint une bonne précision, mais la banque craint que certains demandeurs de prêt ne soient injustement rejetés. [5]

Pour répondre à cette préoccupation, la banque intègre des techniques d'explicabilité et d'interprétabilité dans le modèle. Plus précisément, ils utilisent **"SHAP" (Shapley Additive Explanations)** pour identifier les caractéristiques de l'ensemble de données qui sont les plus importantes pour conduire les prédictions du modèle. Ils créent également un "arbre de décision" qui montre la logique du processus de prise de décision du modèle. [3]

Avec ces techniques en place, les demandeurs de prêt qui sont rejetés par le modèle peuvent demander une explication de la décision. La banque peut ensuite fournir une explication qui met en évidence les facteurs qui ont contribué à la décision, comme une faible cote de crédit ou un ratio d'endettement élevé. Cela permet aux demandeurs de mieux comprendre pourquoi leur demande a été rejetée et de prendre des mesures pour améliorer leur solvabilité si nécessaire.

De cette façon, les techniques d'explicabilité et d'interprétabilité peuvent aider directement les gens en assurant la transparence et l'équité du processus de demande de prêt. Les demandeurs de prêt sont mieux à même de comprendre le processus de prise de décision du modèle et de prendre des mesures pour améliorer leurs chances d'approbation à l'avenir. De plus, la banque peut utiliser les connaissances acquises grâce à ces techniques pour améliorer la précision et l'équité du modèle, au profit de tous les

candidats.

Exemple 2. Supposons qu'une entreprise de soins de santé ait développé un modèle d'apprentissage automatique pour prédire les taux de réadmission des patients. Le modèle est formé sur un ensemble de données de dossiers de patients et utilise des caractéristiques telles que l'âge, le sexe, les antécédents médicaux et l'utilisation de médicaments pour faire des prédictions. Le modèle atteint une bonne précision, mais l'entreprise de soins de santé veut s'assurer que le modèle prend des décisions éthiques et transparentes.

Pour répondre à cette préoccupation, l'entreprise de soins de santé intègre des techniques d'explicabilité et d'interprétabilité dans le modèle. Plus précisément, ils utilisent une technique appelée **"LIME" (Local Interpretable Model-Agnostic Explanations)** pour expliquer les prédictions individuelles faites par le modèle. LIME génère des "cartes explicatives" qui mettent en évidence les caractéristiques du dossier du patient qui ont le plus d'influence sur la prédiction du modèle.

Avec ces techniques en place, l'entreprise de soins de santé est en mesure de mieux comprendre comment le modèle fait ses prédictions et d'identifier tout biais potentiel ou problème éthique. Par exemple, ils peuvent découvrir que le modèle prend des décisions basées sur des informations sensibles telles que l'origine ou la religion. Ils peuvent ensuite ajuster les paramètres du modèle ou les données de formation pour résoudre ces problèmes et améliorer la transparence et l'équité du modèle.

En résumé, en incorporant des techniques d'explicabilité et d'interprétabilité dans son modèle d'apprentissage automatique, l'entreprise de soins de santé a pu mieux comprendre comment le modèle faisait ses prédictions, détecter les biais potentiels et améliorer la transparence et l'équité du modèle. Ce n'est qu'un exemple de la façon dont ces techniques peuvent contribuer à améliorer l'utilisation éthique et transparente de l'apprentissage automatique dans les soins de santé.

2.4 Caractéristiques des méthodes d'interprétabilité

Méthodes d'interprétation agnostique versus spécifique : Les méthodes agnostiques s'utilisent pour n'importe quel type de modèles (Random Forest, CNN, SVM...). Au contraire, les modèles spécifiques ne peuvent être utilisés que pour interpréter une famille spécifique d'algorithmes (ex. CNN). Un explicateur est **agnostique** vis-à-vis du modèle lorsqu'il traite le modèle original comme une boîte noire.

Méthodes intrinsèques Versus méthodes post-hoc : Pour les méthodes intrinsèques, l'interprétabilité est directement liée à la simplicité du modèle alors que pour les méthodes post-hoc, le modèle n'est pas interprétable parce que d'emblée trop complexe.

Méthodes locales versus globales : Les méthodes locales donnent une interprétation pour un seul ou un petit nombre d'observations. Au contraire, les méthodes d'interprétation globales permettent d'expliquer toutes les observations en même temps, globalement.

Méthodes dites a priori versus a posteriori : Les approches a priori sont employées sans hypothèse sur les données et avant la création du modèle. Au contraire, les approches a posteriori sont employées après la création du modèle.

Pour qu'une explication soit significative, elle doit au moins être **localement fidèle**, c'est-à-dire qu'elle doit correspondre à la façon dont le modèle se comporte dans le voisinage (la proximité) de l'instance prédite. Nous notons que la fidélité locale n'implique pas la fidélité globale : les caractéristiques qui sont globalement importantes peuvent ne pas l'être dans le contexte local, et vice versa.

L'état de l'art actuel autour de l'interprétabilité des modèles de machine learning montre qu'il y a une volonté forte de mixer les différentes méthodes : intrinsèques globales ou post hoc globales agnostiques ou encore post-hoc locales agnostiques.

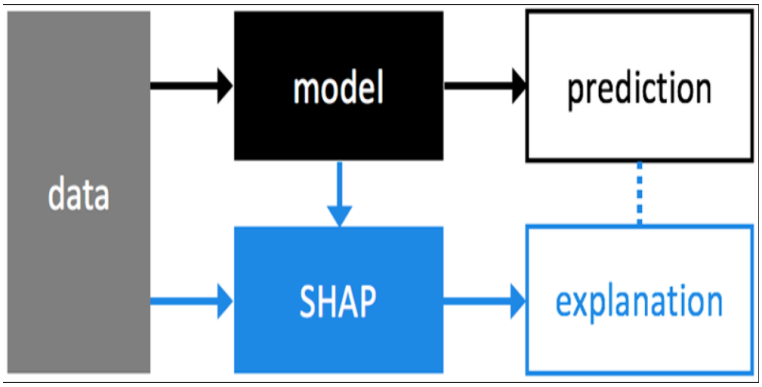
Nous proposons de vous présenter deux méthodes d'interprétation de modèles de Machine Learning : les algorithmes **LIME et SHAP**. Ces deux méthodes fonctionnent en sortie d'un modèle complexe, boîte noire dont on comprend mal le fonctionnement.

3 SHAP (SHapley Additive exPlanations) [3]

3.1 Introduction de SHAP

SHAP est une méthode mathématique pour expliquer les prédictions des modèles d'apprentissage automatique. Il est basé sur les concepts de la théorie des jeux et peut être utilisé pour expliquer les prédictions de n'importe quel modèle d'apprentissage automatique en calculant la contribution de chaque caractéristique à la prédiction.

SHAP est un explicateur individualisé indépendant du modèle. Une méthode indépendante du modèle suppose que le modèle à expliquer est une boîte noire et ne sait pas comment le modèle fonctionne en interne. Ainsi, la méthode indépendante du modèle ne peut accéder qu'aux données d'entrée et à la prédiction du modèle à expliquer.(Un explicateur individualisé indépendant du modèle est lui-même un modèle interprétable.)



3.2 Exemple pour comprendre SHAP

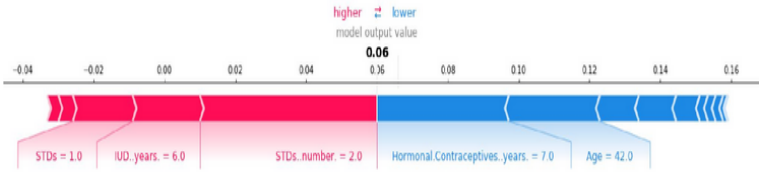
SHAP est une méthode d'explication de modèle qui peut donner une explication **globale et locale** pour notre modèle et notre point de données respectivement.

Un exemple très courant pour comprendre le concept SHAP value est de mesurer quel joueur devrait recevoir un pourcentage plus élevé du montant du prix remporté dans une compétition par une équipe. Nous ne pouvons pas directement répartir le prix équitablement entre les joueurs car certains joueurs sont plus forts et ont une plus grande responsabilité dans une équipe que d'autres.

Mais ce n'est pas facile de décider et de le calculer, nous utilisons donc le concept de **Théorie des jeux co-orporés** dans lequel nous décidons de l'importance d'une feature/personne dans la prédiction en faisant une prédiction avec et sans lui, en prenant plusieurs sous-ensembles de ce particulier feature avec d'autres features, puis en calculant la prédiction.

À la fin, en utilisant ce concept, nous calculons les valeurs **SHAPLEY** pour chaque caractéristique de notre ensemble de données et analysons quelle caractéristique est importante dans notre prédiction pour une classe particulière.

Nous pouvons visualiser cela à l'aide de ce graphique, qui nous montre graphiquement quelle caractéristique est importante ou non pour notre prédiction.



3.3 Formulation mathématique de SHAP

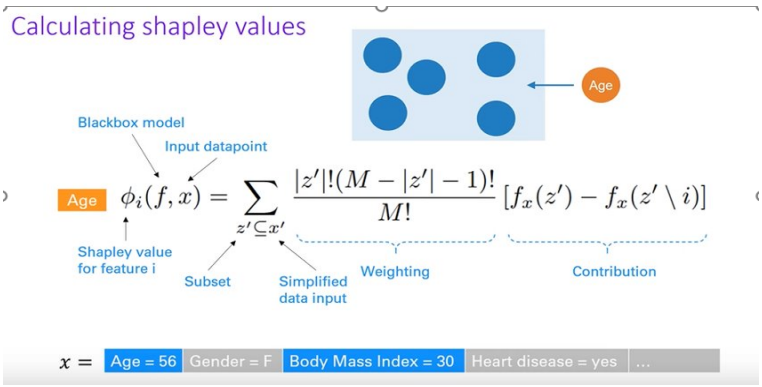
Les valeurs SHAP sont calculées à l'aide d'une formule mathématique complexe qui combine divers éléments de la théorie des jeux, de l'algèbre linéaire et de l'optimisation. L'idée de base derrière la formule est de déterminer la contribution de chaque caractéristique à la prédiction, en tenant compte à la fois de l'ampleur et de la direction de la contribution.

La formule de calcul des valeurs SHAP pour une prédiction donnée est basée sur les valeurs de Shapley de la théorie des jeux coopératifs, qui fournissent une allocation équitable d'une valeur aux facteurs contributifs. La formule prend en compte l'influence de chaque caractéristique sur la prédiction, ainsi que les interactions entre les caractéristiques.

Ici Φ est appelée la valeur de Shapley de l'élément {i} qui est la contribution moyenne de {i} dans toutes les permutations de F. C'est la part mathématiquement équitable du joueur {i} dans le gain total de tous les joueurs de F Comme nous l'avons montré précédemment, chaque coalition S, peut faire $S!(|F| - |S| - 1)!$ permutations. Le nombre total de permutations étant $|F|!$, on peut écrire :

$$\phi(f,x) = \sum_{S \subseteq F-i} \frac{|S|!(|F|-|S|-1)!}{|F|!}[f_x(S) - f_x(S \setminus i)]$$

F désigne l'ensemble des joueurs
S est un sous-ensemble de F ($S \subseteq F$)



3.4 Proprietés associe avec model SHAP :

Les valeurs SHAP de la caractéristique (i), notées $\phi_i(x)$, mesurent la contribution de la caractéristique i à la prédiction du modèle. Plus précisément, les valeurs SHAP satisfont les propriétés suivantes :

Additivité : la somme des valeurs SHAP de toutes les caractéristiques est égale à la différence entre la prédiction du modèle pour l'instance x et la prédiction moyenne du modèle sur toutes les

instances possibles :

$$\sum_i \phi_i(x) = f(x) - E[f(Z)]$$

où Z est une variable aléatoire qui prend des valeurs dans le domaine de x, et E[f(Z)] est la valeur attendue de la prédiction du modèle sur toutes les instances possibles.

Cohérence : Si la caractéristique i n'est pas pertinente pour la prédiction du modèle, sa valeur SHAP est nulle :

$$\phi_i(x) = 0$$

si la caractéristique i n'est pas pertinente.

Précision locale : les valeurs SHAP se rapprochent de la contribution réelle de chaque caractéristique à la prédiction du modèle pour l'instance spécifique x :

$$f(x) = \sum_i \phi_i(x).$$

3.5 Avantages de SHAP

Comme SHAP calcule des valeurs de Shapley, tous les avantages des valeurs de Shapley s'appliquent : SHAP repose sur une base théorique solide en matière de théorie des jeux. La prédiction est équitablement répartie entre les valeurs des caractéristiques. Nous obtenons des explications contrastives qui comparent la prédiction à la prédiction moyenne.

SHAP relie LIME et les valeurs de Shapley. Cela est très utile pour mieux comprendre les deux méthodes. Il contribue également à unifier le domaine de l'apprentissage automatique interprétable.

SHAP dispose d'une implémentation rapide pour les modèles basés sur les arbres. Je pense que c'est la clé de la popularité de SHAP, car le plus grand obstacle à l'adoption des valeurs de Shapley est la lenteur des calculs.

3.6 Application de SHAP

SHAP (SHapley Additive exPlanations) a des applications pratiques dans divers domaines, notamment :

Finance : SHAP peut être utilisé pour expliquer les résultats des modèles utilisés pour la notation du crédit, la détection des fraudes et l'évaluation des risques. Cela peut aider les institutions financières à se conformer aux réglementations et à instaurer la confiance avec les clients.

Soins de santé : SHAP peut être utilisé pour expliquer les résultats des modèles utilisés pour le diagnostic des maladies, la planification du traitement et la prédiction des résultats pour les patients. Cela peut aider les professionnels de la santé à prendre des décisions éclairées et à améliorer les soins aux patients.

Marketing : SHAP peut être utilisé pour expliquer le résultat des modèles utilisés pour la segmentation de la clientèle, la publicité ciblée et les recommandations personnalisées. Cela peut aider les entreprises à améliorer leurs stratégies de marketing et à accroître la satisfaction de leurs clients.

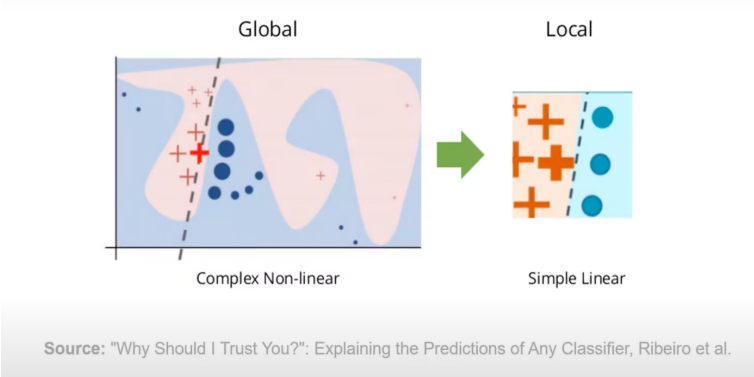
Fabrication : SHAP peut être utilisé pour expliquer la sortie des modèles utilisés pour le contrôle qualité, la maintenance prédictive et l'optimisation des processus. Cela peut aider les fabricants à améliorer leurs processus de production et à réduire leurs coûts.

Transport : SHAP peut être utilisé pour expliquer les résultats des modèles utilisés pour la prévision du trafic, l'optimisation des itinéraires et la prévision de la demande. Cela peut aider les entreprises de transport à améliorer leurs services et à réduire la congestion.

Traitement du langage naturel : SHAP peut être utilisé pour expliquer la sortie des modèles utilisés pour l'analyse des sentiments, la reconnaissance d'entités nommées et la classification de texte. Cela peut aider à améliorer la précision des modèles NLP et à renforcer la confiance dans leurs prédictions.

4 LIME(Local Interpretable Model-agnostic Explanations) [5]

4.1 Présentation/Contexte



Le graphique ci-dessus résume le fonctionnement de LIME dans le cas d'un modèle de classification binaire, la zone bleu repésente les points prédit dans la classe 0 tandis que la zone rouge représente les points prédits dans la classe 1. En premier lieu, on se place sur un point de vue local. Les croix représentent les points simulés par le modèle de substitution et la droite en pointillés représente le modèle linéarie simple obtenu par l'algorithme Lime.

L'algorithme LIME est un modèle local qui cherche à expliquer la prédiction d'un individu par analyse du voisinage, point de vue local. Les modèles de substitution locaux sont des modèles interprétables qui sont utilisés pour expliquer les prédictions individuelles des modèles d'apprentissage automatique de type boîte noire.

Local Interpretable Models-agnostics Explainable (LIME) [5] est un article dans lequel les auteurs proposent une mise en œuvre concrète des modèles de substitution locaux. Les modèles de substitution sont formés pour approximer les prédictions de modèles de type boîte noire. Au lieu de former un modèle de substitution global, LIME se concentre sur la formation de modèles de substitution locaux pour expliquer les prédictions individuelles.

L'idée est assez intuitive. Tout d'abord, oubliez les données d'apprentissage et imaginez que vous n'avez que le modèle de la boîte noire dans lequel vous pouvez entrer des données et obtenir les prédictions du modèle. Vous pouvez sonder la boîte aussi souvent que vous le souhaitez. Votre objectif est de comprendre pourquoi le modèle d'apprentissage automatique a fait une certaine prédiction.

LIME teste ce qui arrive aux prédictions lorsque vous donnez des variations de vos données au modèle d'apprentissage automatique. LIME génère un nouvel ensemble de données composé d'échantillons perturbés et des prédictions correspondantes du modèle de la boîte noire. Sur ce nouvel ensemble de données, LIME forme ensuite un modèle interprétable, qui est pondéré par la proximité des instances échantillonnées avec l'instance d'intérêt.

LIME a la particularité d'être un modèle :

Interprétable Il fournit une compréhension qualitative entre les variables d'entrée et la réponse. Les relations entrées-sortie sont faciles à comprendre.

Simple localement Le modèle est globalement complexe, il faut alors chercher des réponses localement plus simples.

Agnostique Il est capable d'expliquer n'importe quel modèle de machine learning.

Pour ce faire :
1 : L'algorithme LIME génère des nouvelles données, dans un voisinage proche de l'individu à expliquer.

2 : LIME entraîne un modèle transparent sur les prédictions du modèle « boîte noire » complexe qu'on cherche à interpréter. Il apprend ainsi à l'aide d'un modèle simple et donc interprétable (par exemple, une régression linéaire ou un arbre de décision).

En résumé ,pour former des modèles locaux de substitution vous devez :
- Sélectionner votre instance d'intérêt pour laquelle vous voulez avoir une explication de sa prédiction boîte noire.
- Perturber votre ensemble de données et obtenez les prédictions de la boîte noire pour ces nouveaux points.
- Pondérer les nouveaux échantillons en fonction de leur proximité avec l'instance d'intérêt.
- Entraîner un modèle pondéré et interprétable sur l'ensemble de données avec les variations.
- Expliquer la prédiction en interprétant le modèle local.

Le modèle transparent joue donc le rôle de modèle de substitut pour interpréter les résultats du modèle complexe d'origine.

Le principal inconvénient de la méthode LIME est lié à son fonctionnement local. Et, LIME ne permet pas de généraliser l'interprétabilité issue du modèle local à un niveau plus global.

4.2 Formulation Mathématique

Mathématiquement, disons que nous avons un modèle boîte noire $f(x)$ qui associe les entrées x aux sorties y, et que nous voulons expliquer la prédiction $f(x_0)$ pour une entrée spécifique x_0. Pour ce faire, nous échantillonnons d'abord m instances $x_i, i = 1, 2, ..., m$ dans un voisinage de x_0 et obtenons les prédictions correspondantes $f(x_i)$. Nous ajustons ensuite un modèle interprétable simple $g(x; \theta)$ aux paires $(x_i, f(x_i))$ en utilisant une fonction de perte $L(g(x_i; \theta), f(x_i))$. Les paramètres θ du modèle interprétable sont choisis de telle sorte que la perte soit minimisée.

L'explication produite par LIME est obtenue par la formule suivante :

$$f(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Où l'on a :

- G est l'ensemble des modèles interprétable, tel que les modèles linéaire, les arbres de décision etc

- Le domaine de g est $\{0, 1\}^d$. C'est à dire g agit sur l'absence ou la présence du composé interprétable. Comme tous les g ne sont pas forcément interprétables, on ajoute donc $\Omega(g)$ qui serait une mesure de la complexité (en opposition à l'interprétabilité) de g.

- Notons $f : R^d \to R$. En classification, f(x) est la probabilité (ou un indicateur binaire) que x appartienne à une certaine classe. Nous utilisons en outre $\pi * x(z)$ comme mesure de proximité entre une instance z et x, afin de définir la localité autour de x.

- Le modèle d'explication pour l'instance x est le modèle g (par exemple, un modèle de régression linéaire) qui minimise la perte L (par exemple, l'erreur quadratique moyenne), qui mesure la proximité de l'explication avec la prédiction du modèle original f (par exemple, un modèle xgboost).

- $\Omega(g)$ est maintenue faible (par exemple, préférer moins de caractéristiques). G est la famille d'explications possibles, par exemple tous les modèles de régression linéaire possibles. La mesure de proximité π_x définit la taille du voisinage de l'instance x que nous considérons pour l'explication. En pratique, LIME optimise uniquement la partie perte. L'utilisateur doit déterminer la complexité, par exemple en sélectionnant le nombre maximum de caractéristiques que le modèle de régression linéaire peut utiliser.

4.3 Les avantages de LIME

LIME (Local Interpretable Model-Agnostic Explanations) est une méthode utilisée pour expliquer les prédictions de tout modèle d'apprentissage automatique, quelle que soit son architecture sous-jacente. Voici quelques avantages de LIME :

1.**Indépendant du modèle** : LIME est indépendant du modèle, ce qui signifie qu'il peut être appliqué à n'importe quel modèle d'apprentissage automatique, y compris des modèles complexes tels que les réseaux de neurones, les forêts aléatoires et les machines à vecteurs de support.

2.**Explications locales** : LIME fournit des explications locales pour les prédictions individuelles, ce qui signifie qu'il explique comment le modèle arrive à ses prédictions pour une instance ou une observation spécifique, plutôt que de fournir une explication globale qui s'applique à l'ensemble de données.

3.**Compréhensible par l'homme** : LIME fournit des explications dans un format compréhensible par l'homme, tel que du texte ou des visualisations, ce qui permet aux parties prenantes non techniques de comprendre facilement comment le modèle effectue ses prédictions.

4.**Interprétable** : LIME génère des explications qui sont interprétables et transparentes, ce qui peut aider à renforcer la confiance dans le modèle et à améliorer les performances globales du modèle.

5.**Flexible** : LIME est une méthode flexible qui peut être appliquée à tout type de données, y compris le texte, l'image et les données tabulaires. Il peut également être utilisé pour les tâches de classification et de régression.

6.**Évolutif** : LIME est une méthode évolutive qui peut gérer de grands ensembles de données et qui est efficace en termes de calcul, ce qui en fait un outil pratique pour les applications du monde réel.

5 Les K+ proche voisins (KNN) [2]

5.1 Introduction

KNN (k-plus proches voisins) est un algorithme d'apprentissage automatique utilisé pour les tâches de classification et de régression. Il s'agit d'un type d'algorithme d'apprentissage supervisé qui utilise une méthode non paramétrique pour faire des prédictions.

L'idée de base derrière l'algorithme KNN est de trouver les **k voisins** les plus proches d'un point de données dans l'espace des caractéristiques, puis d'utiliser la classe majoritaire de ces voisins comme prédiction pour le nouveau point de données. Par exemple, dans une tâche de classification, l'algorithme KNN trouverait les k voisins les plus proches d'un point de données donné, puis classerait le nouveau point de données en fonction de la classe majoritaire de ces voisins.

Le choix de la valeur de k est un paramètre important dans l'algorithme KNN, car il détermine le nombre de voisins à considérer lors des prédictions. Une plus grande valeur de k signifierait un modèle plus robuste, mais cela pourrait également conduire à une sur-généralisation, tandis qu'une plus petite valeur de k signifierait un modèle plus flexible, mais cela pourrait également conduire à un surajustement.

L'algorithme KNN est facile à mettre en œuvre et est souvent utilisé comme algorithme de référence pour évaluer les performances d'autres modèles d'apprentissage automatique. Cependant, cela peut être coûteux en calcul, en particulier pour les grands ensembles de données, car cela nécessite de calculer les distances entre toutes les paires de points de données.

5.2 KNN avec le dataset Diabètes

Dans notre travail pratique, nous avons formé l'ensemble de données du **modèle KNN** sur le **Diabète** pour prédire les résultats. Nous présentons ici quelques résultats qui lui sont associés.

Précision : Nous obtenons **75**% de bien classés par la méthode KNN.

Dans cet exemple, nous chargeons d'abord l'ensemble de données sur le diabète dans un pandas Data Frame. Nous avons ensuite divisé les données en ensembles d'apprentissage et de test à l'aide de la fonction train_test_split de **scikit-learn**. Nous ajustons un classifieur KNN avec n_neighbors=5 sur les données d'apprentissage en utilisant la classe KNeighborsClassifieur de scikit-learn. Nous utilisons ensuite le modèle formé pour faire des prédictions sur les données de test. Enfin, nous évaluons les performances du modèle à l'aide du score de précision du module de métriques de scikit-learn.

6 Decision tree [1]

6.1 Introduction

Un **arbre de decision** est un modèle d'apprentissage automatique qui permet de prendre des décisions en fonction de la valeur des variables prédictives. Il est composé d'un ensemble de nœuds de décision qui sont reliés entre eux par des branches, et qui représentent les différentes décisions à prendre en fonction des valeurs des variables prédictives.

Le premier nœud de l'arbre, appelé nœud racine, représente la première décision à prendre en fonction de la valeur d'une variable prédictive. À chaque nœud de l'arbre, on choisit une variable prédictive et on définit une condition sur sa valeur pour décider du chemin à suivre dans l'arbre. Les nœuds suivants de l'arbre représentent ainsi les différentes décisions à prendre en fonction des valeurs des variables prédictives jusqu'à atteindre une feuille de l'arbre qui représente la décision finale.

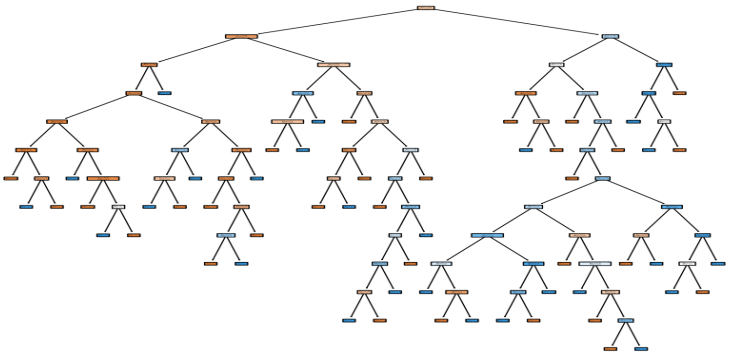
Voici un exemple simple d'arbre de décision pour un problème de classification :
Supposons que nous avons un ensemble de données contenant des informations sur des fruits, notamment leur couleur, leur taille et leur forme, et que nous souhaitons prédire leur type (pomme ou orange).
Dans cet arbre de décision, le nœud racine est basé sur la couleur du fruit. Si la couleur est rouge ou verte, on passe au nœud suivant qui prend en compte la forme du fruit. Si la forme est ronde, on prédit que le fruit est une pomme, sinon on prédit que c'est une orange. Si la couleur n'est pas rouge ou verte, on prédit directement que c'est une orange.

6.2 Decision tree avec le dataset Diabètes

Dans notre étude, nous avons formé l'ensemble de données du modèle Decsion tree sur le Diabète pour prédire les résultats. Nous présentons ici quelques résultats qui lui sont associés.

Pour cela, nous avons utilisé les mêmes fonctions que précédemment. Puis, nous avons ajusté un classifieur Random Forest sur les données d'apprentissage et utilisé le modèle formé pour faire des prédictions sur les données de test. Enfin, nous avons évalué les performances du modèle en utilisant le score de précision fourni par le module de métriques de Scikit-Learn.

Précision : Nous avons ici obtenu une précision de prédiction d'environ **72%**.



7 Forêt Aléatoire [4]

7.1 Introduction

Forêt Aléatoire est un algorithme d'apprentissage automatique supervisé qui est utilisé pour résoudre des problèmes de classification, de régression et de détection d'anomalies. Il fait partie de la famille des méthodes d'ensemble (ensemble methods) qui consistent à combiner les prédictions de plusieurs modèles simples pour obtenir une prédiction finale plus précise.

L'algorithme de Forêt Aléatoire utilise un ensemble d'arbres de décision, qui sont des modèles simples de type "si-alors" qui permettent de prendre des décisions en fonction de la valeur des variables prédictives. Pour construire un modèle de Forêt Aléatoire, on entraîne plusieurs arbres de décision sur des sous-ensembles aléatoires des données d'entraînement, en utilisant des échantillons bootstrap (bootstrap samples) et en sélectionnant un sous-ensemble aléatoire des variables prédictives à chaque nœud de décision. Cela permet de réduire le risque de sur-apprentissage (overfitting) et d'obtenir une prédiction plus robuste.

Pour effectuer une prédiction avec un modèle de Forêt Aléatoire, on fait passer les données à travers chaque arbre de décision et on combine les prédictions de tous les arbres en prenant la moyenne pour la classification, ou la moyenne pondérée pour la régression. L'algorithme de Forêt Aléatoire est capable de gérer des ensembles de données de grande dimension avec des milliers de variables prédictives et peut être utilisé dans une grande variété de contextes, tels que la finance, la santé, la biologie, la géologie, la reconnaissance vocale et d'image, la sécurité, etc.

Les avantages de l'algorithme de Forêt Aléatoire sont sa robustesse aux données manquantes et aux valeurs aberrantes, sa capacité à traiter des ensembles de données de grande dimension avec des milliers de variables prédictives, et sa résilience aux problèmes de sur-apprentissage. De plus, l'algorithme de Forêt Aléatoire est relativement facile à implémenter et à utiliser, même pour les débutants en apprentissage automatique.

En résumé, l'algorithme de Forêt Aléatoire est une méthode d'ensemble populaire en apprentissage automatique, qui est utilisée pour résoudre des problèmes de classification, de régression et de détection d'anomalies dans une grande variété de contextes. Son utilisation est justifiée par sa capacité à traiter des données de grande dimension avec des milliers de variables prédictives, sa robustesse aux données manquantes et aux valeurs aberrantes, et sa résilience aux problèmes de sur-apprentissage.

7.2 Forêt Aléatoire avec le dataset Diabètes

Dans notre travail pratique, nous avons formé l'ensemble de données du modèle Forêt Aléatoire sur le Diabète pour prédire les résultats. Nous présentons ici quelques résultats qui lui sont associés.

Dans cet exemple, nous avons appliqué les mêmes procédés que précédemment. Ensuite, nous ajustons un classifieur Random forest sur les données d'apprentissage. Nous utilisons ensuite le modèle formé pour faire des prédictions sur les données de test. Enfin, nous évaluons les performances du modèle à l'aide du score de précision du module de métriques de scikit-learn.

Précision : Nous avons obtenu une précision de prédiction d'environ **82.3%** en utilisant les Forêt Aléatoire.

8 Partie Analytique sur le dataset Diabète

Dans cette partie, nous vous fournirons une brève introduction de notre jeu de données sur **le Diabète** avec une étude statistique réalisée sur **Python** à l'aide de bibliothèques telles que **Numpy, Pandas** etc.

8.1 Introduction

Cet ensemble de données contient des informations sur 393 personnes du peuple amerindien Pima, peuple réputé pour avoir un taux d'obésité et de diabète élevé. Les personnes étudiées sont des femmes âgées d'au moins 21 ans. L'ensemble de données comprend huit variables prédictives médicales et une variable binaire qui indique si le patient a développé (1) ou non (0) un diabète dans les cinq ans.

Les variables prédictives sont :
1. Nombre de fois où la personne a été enceinte **(Pregnant)**
2. Concentration plasmatique du glucose **(Glucose)**
3. Pression artérielle diastolique (mm Hg) **(Blood Pressure)**
4 .Épaisseur de la peau (mm) **(Skin Thickness)**
5. Insuline(mu U/ml) **(Insulin)**
6. Indice de masse corporelle (poids en kg/(taille en m^2)) **(BMI)**
7. Diabetes pedigree function
8. Age **(Age)**

La variable de sortie est **une variable binaire** qui prend la valeur 0 (indiquant l'absence de diabète) ou 1 (indiquant le diabète).

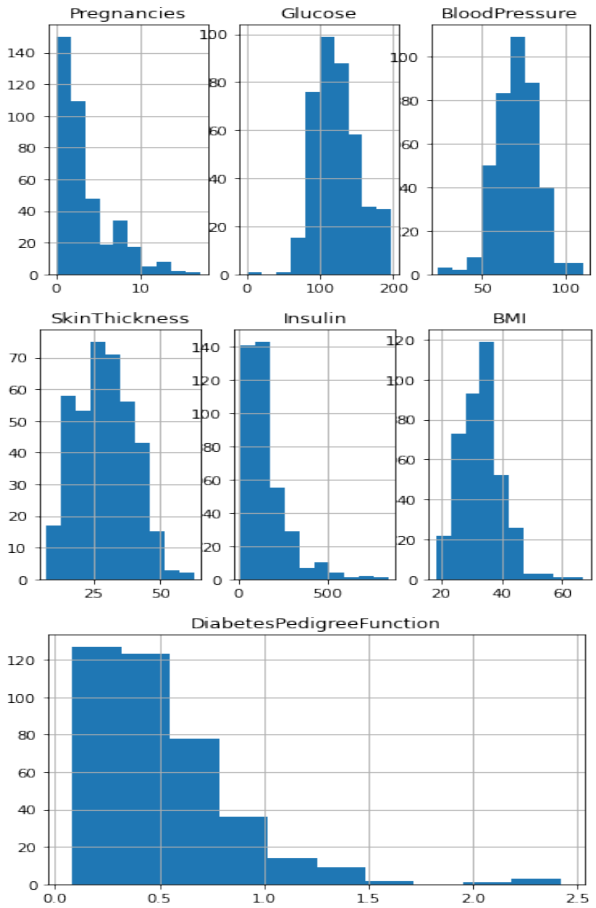
Cet ensemble de données est souvent utilisé dans la recherche sur l'apprentissage automatique pour développer des modèles prédictifs de diagnostic du diabète basés sur ces variables prédictives. Il peut également être utilisé pour explorer les relations entre ces variables et la probabilité de développer un diabète.

8.2 Statisqtique Descriptive

8.2.1 Etude des différentes features

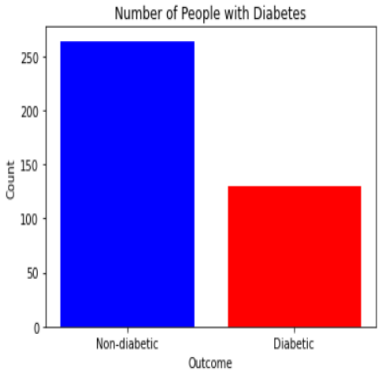
A présent, nous faisons une étude decriptive des nos différents features, nous allons aborder les différentes moyennes, écart-types, le min et le max de nos différentes features ainsi que la proportion des résultats finaux. Nous présenterons aussi les diagrammes des différentes features afin d'avoir une idée sur leur répartition.

	Pregnancies	Glucose	BloodPressure	SkinThick	Insulin	BMI	Pedigree	Age	Outcome
Count	394	394	394	394	394	394	394	394	394
Mean	3.28	122.30	70.65	29.10	155.54	32.98	0.52	30.81	0.32
Std	3.20	31.39	12.46	10.50	118.77	7.21	0.35	10.19	0.47
Min	0	0	24	7	14	0	0.085	21	0
Max	17	198	110	63	846	67.10	2.420	81	1



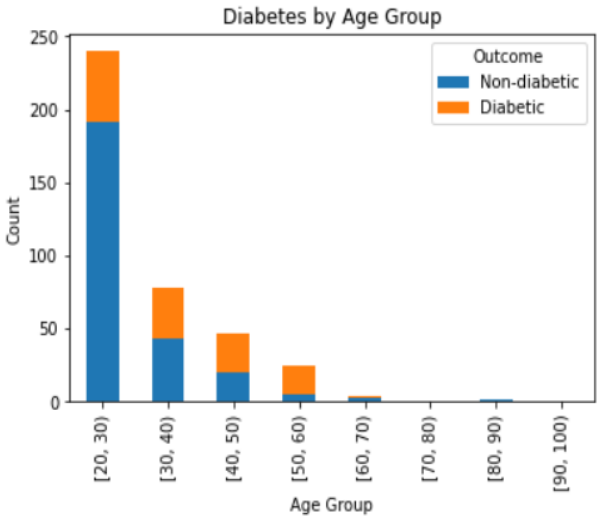
8.2.2 Diabète v/s Non Diabète

Sur les 394 personnes étudiées, 130 sont diabétiques et 264 ne le sont pas.



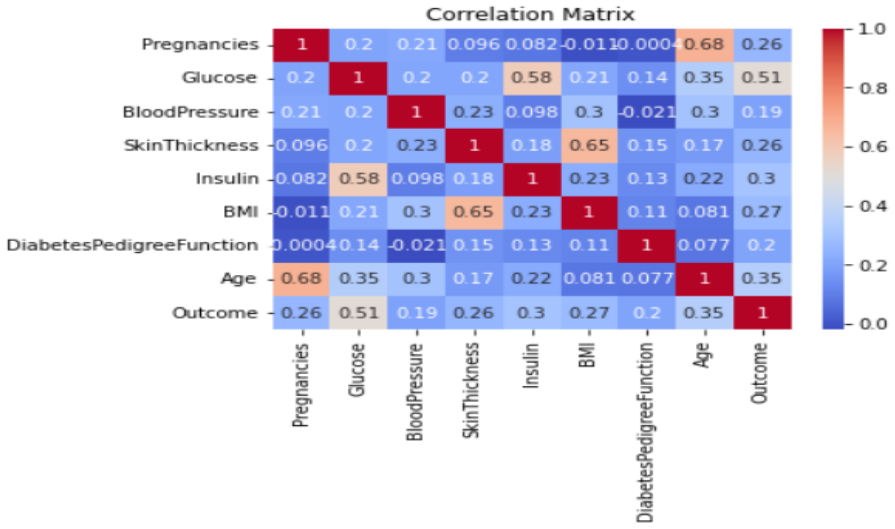
8.2.3 Diabètes v/s Age

Dans cette partie, nous étudierons le graphique qui donne le nombre de personnes diabétiques ou pas dans un groupe d'âge particulier. On constate que le diabète se déclenche très tôt dans cette population et que déjà chez les 20-30 ans, il y a plus de personnes diabétique que de personnes non diabétique. On constate aussi que la population n'a pas une éspérance très élevée.



8.2.4 Correlation entre les différentes features

Dans cette partie, nous présenterons le tableau de corrélations des différentes features.



Cela génère une Heatmap qui affiche les coefficients de corrélation entre chaque paire de variables dans l'ensemble de données, où les couleurs chaudes indiquent une forte corrélation, et les couleurs plus froides indiquent une faible corrélation.Dans notre cas, nous sommes plus intéréssés par les corrélations entre les différentes features et la variable de sortie. Nous pouvons constater que la corrélation la plus élevé est celle avec le glucose, ce qui est cohérent.

9 Application de LIME et SHAP sur nos 3 modèles

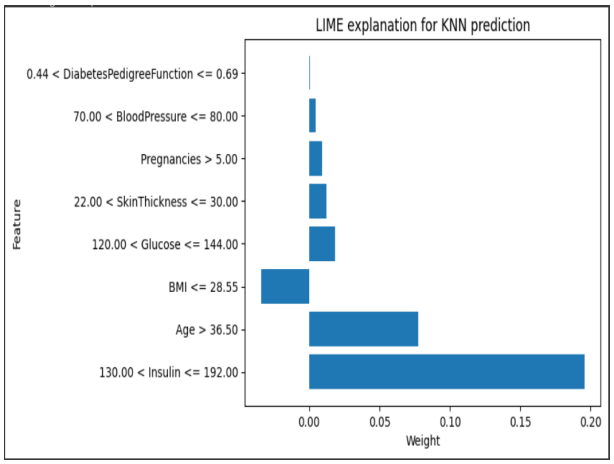
Pour la partie programmation, on va utiliser le language Python ainsi que les bibliothéques **Numpy, Matplot, Pandas, SHAP, LIME, KNN, Decision Tree, Random Forest...**.

9.1 Les K Plus Proche Voisin

Comme on sait déjà que l'algorithme de KNN est explicable et interpretable mais qu'il a une précision pas très élevée comparer aux autres modèles de type boite noire. On a vu dans le partie 6.2 qu'on a eu une précision de 75% pour les KNN.

9.1.1 LIME pour KNN

On a essayé d'utiliser l'algorithme de LIME pour voir l'explicabilité et l'interpretabilité pour KNN dans notre partie pratique.



Grâce à ce graphe, on peut conclure que le décision par KNN est basée sur les 3 features importants qui sont : **Insulin, Age, Glucose**.

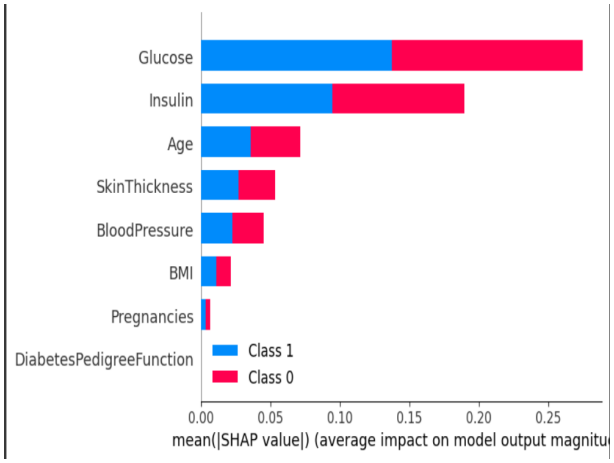
Nous avons créé un classificateur KNN avec numéro de voisins égale à 5 sur les données d'apprentissage. Nous utilisons ensuite la méthode LIME pour expliquer les prédictions sur une seule instance de test (dans ce cas, la première instance de l'ensemble de test).

La caractéristique avec la plus grande contribution positive de la classe « Diabète » est « Insulin », ce qui suggère que des niveaux d'insuline plus élevés sont associés à un risque plus élevé de diabète. Ce qui est cohérent avec les connaissances antérieures sur le diabète, car l'insuline est un biomarqueur clé pour le diagnostic et la gestion du diabète.

La caractéristique avec la plus grande contribution négative de la classe « Diabète » est « l'IMC », ce qui suggère que des valeurs d'IMC plus faibles sont associées à un risque moins élevé de diabète. Ce qui est cohérent encore une fois.

Les autres caractéristiques du graphique (BloodPressure, SkinThickness et DiabetesPedigreeFunction) ont des contributions plus faibles à la classe prédite, qu'elles soient positives ou négatives. Cela suggère qu'ils sont moins importants pour prédire le risque de diabète dans ce cas particulier, ou que leurs relations avec le risque de diabète sont plus complexes et difficiles à saisir avec un modèle KNN simple.

9.1.2 SHAP pour KNN



Grâce à ce graphe on peut conclure que le décision par KNN est basé sur les 3 mêmes features que précédemment qui son **Insulin, Glucose et l'âge**.

L'axe y du tracé affiche les noms des caractéristiques et l'axe x affiche les valeurs SHAP correspondantes. Les valeurs SHAP représentent la contribution de chaque entité à la sortie du modèle pour chaque échantillon individuel du jeu de données. Les valeurs SHAP positives indiquent que la fonctionnalité augmente la valeur de sortie, tandis que les valeurs SHAP négatives indiquent que la fonctionnalité diminue la valeur de sortie.

Chaque point du graphique représente un seul échantillon de l'ensemble de données. La couleur du point indique la valeur de la caractéristique pour cet échantillon, le bleu représentant les valeurs de caractéristique faibles et le rouge les valeurs de caractéristique élevées. La position du point le long de l'axe des abscisses indique l'importance de la caractéristique pour cet échantillon, les points les plus élevés indiquant les caractéristiques les plus importantes.

Le tracé comprend également une ligne verticale pour chaque entité qui indique la plage de valeurs SHAP possibles pour cette entité sur tous les échantillons du jeu de données. Cela aide à donner une idée de l'importance globale de chaque caractéristique dans le modèle.

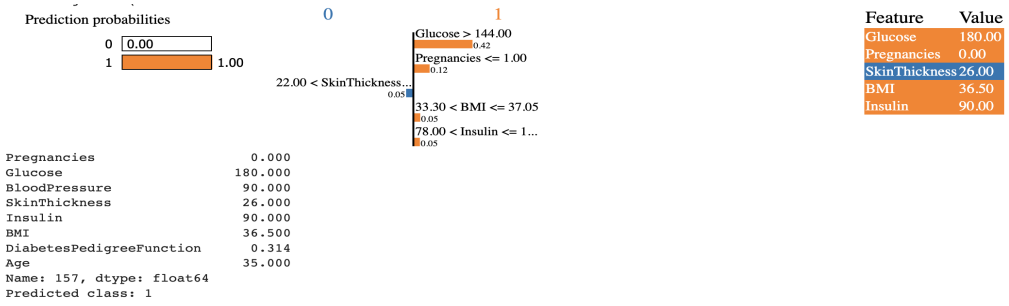
Dans l'ensemble, le tracé donne un résumé visuel des valeurs d'importance des caractéristiques pour le jeu de données, ce qui vous permet d'identifier rapidement les caractéristiques les plus importantes et leur contribution à la sortie du modèle.

9.2 Modèle d'arbre de décision

Comme on sait que l'algorithme utilisant les arbres de décision est trés explicable et interpretable mais qu'il a un précision moyenne comparé aux autres modèles de type boite noire. De plus, on a vu dans le partie 7.2 qu'on a eu une précision de 72% pour l' arbre de décision.

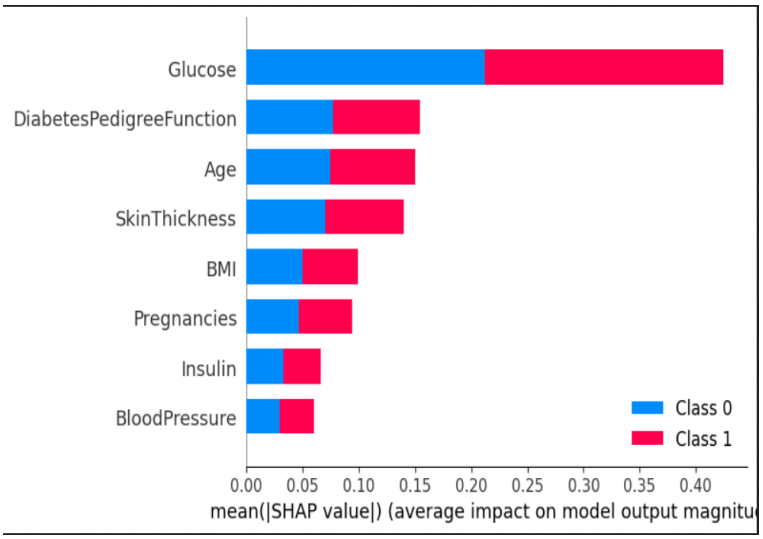
9.2.1 LIME pour Arbre de décision

On a essayé d'utiliser l'algorithme de LIME pour voir l'explicabilité et l'interpretabilité pour les arbres de décision dans notre partie pratique.



Par l'étude de ce graphe on peut conclure que le décision par arbre de décision est basé sur les 2 features importants **Glucose, Pregnancies**.

9.2.2 SHAP pour Arbre de décision



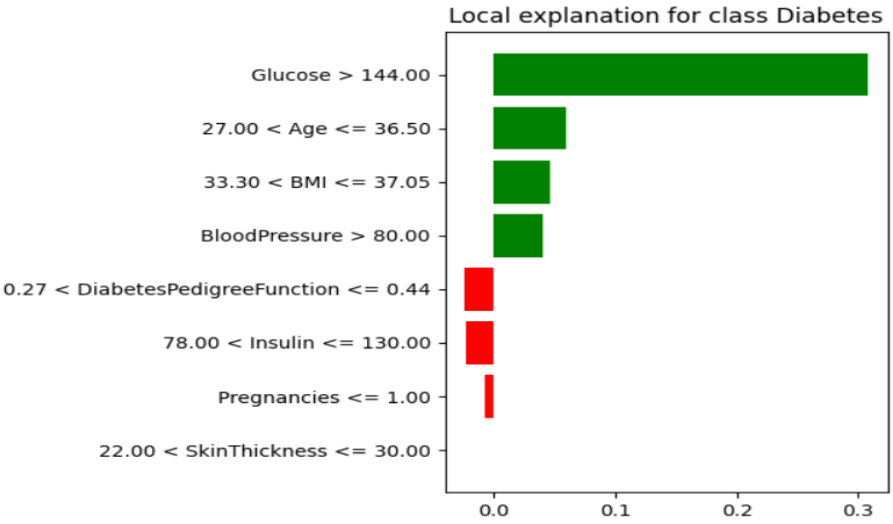
Par l'étude de ce graphe on peut conclure que le décision par arbre de Décision est majoriterement basée sur les features **Glucose, DiabetesPedigreeFunction,âge et SkinThickness**.

9.3 Modèle de Forêt Aléatoire

On a vu que l'algorithme utilisant les forêts aléatoires ne sont pas trés explicable ni interpretable mais qu'ils ont une précision élevé comparé aux autres modèles qui sont plus explicable et interprétable. On a dit dans le partie 8.2 qu'on a reçu le précision de 82.3% pour les forêts aléatoires.

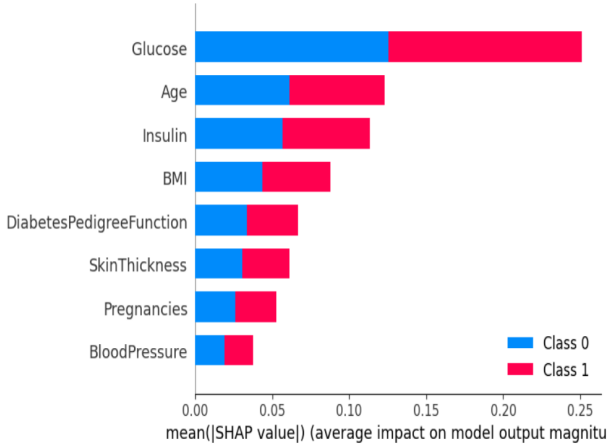
9.3.1 LIME pour Forêt Aléatoire

On a essayé d'utiliser l'algorithme de LIME pour voir l'explicabilité et l'interpretabilité pour les forêts aléatoires dans notre partie pratique.



Par l'étude de ce graphe, on peut conclure que le décision par forêt aléatoire est basé sur 4 features importants que sont : **le Glucose, l'âge, l'IMC et BloodPressure**.

9.3.2 SHAP pour Forêt Aléatoire



Grâce à ce graphe on peut conclure que la décision par forêt aléatoire est basé sur les features : **Glucose, Insulin ,IMC, Age**.

10 Conclusion

10.1 LIME ou SHAP : lequel choisir ?

Dans ce partie on va parler de quelques notions qui vont nous aider à choisir entre **LIME et SHAP**.

LIME (Local Interpretable Model-Agnostic Explanations) et SHAP (SHapley Additive exPlanations) sont des méthodes populaires pour expliquer les prédictions des modèles d'apprentissage automatique. Cependant, il existe des différences essentielles entre les deux méthodes, et le choix entre les deux dépendra des besoins et des exigences spécifiques de notre cas d'utilisation.

10.2 Voici quelques considérations à garder à l'esprit lors du choix entre LIME et SHAP :

10.2.1 Type de modèle

LIME est une méthode indépendante du modèle, ce qui signifie qu'elle peut être utilisée avec n'importe quel modèle de boîte noire.

SHAP, d'autre part, a été conçu spécifiquement pour les modèles additifs, tels que les modèles linéaires, les arbres de décision et les réseaux de neurones. Si vous travaillez avec un modèle complexe et non linéaire, **LIME** peut être un meilleur choix.

10.2.2 Interprétabilité

LIME génère des explications locales, ce qui signifie qu'il se concentre sur l'explication des prédictions d'un modèle pour une instance spécifique ou un sous-ensemble d'instances.

SHAP, d'autre part, génère des explications globales, ce qui signifie qu'il explique le modèle dans son ensemble, à travers toutes les instances.

Si vous avez besoin d'expliquer des prédictions individuelles, **LIME** peut être un meilleur choix, tandis que si vous avez besoin de comprendre comment le modèle fonctionne en général, **SHAP** peut être un meilleur choix.

10.2.3 Performances

LIME est généralement plus rapide que **SHAP**, en particulier pour les ensembles de données plus volumineux et les modèles plus complexes.

Cependant, **LIME** peut être moins précis que **SHAP**, en particulier pour les modèles non linéaires avec des espaces de caractéristiques de grande dimension.

10.2.4 Implémentation

LIME est relativement facile à implémenter et dispose d'un certain nombre de bibliothèques open source disponibles pour Python et R.

SHAP peut être plus difficile à implémenter, en particulier pour les modèles non additifs, mais dispose également de bibliothèques open source.

10.2.5 Compatibilité

LIME et SHAP sont compatibles avec différents types de données, telles que le texte, l'image et les données tabulaires. Selon le type de données avec lesquelles vous travaillez, une méthode peut être plus appropriée qu'une autre.

En définitive, le choix entre LIME et SHAP dépendra de vos besoins spécifiques et des caractéristiques de vos données et de votre modèle. Il peut être utile d'essayer les deux méthodes et de comparer leurs résultats avant de prendre une décision finale.

10.3 Réponse à la problématique

Suite à notre précédente étude, on peut conclure que différentes méthodes existent afin que les utilisateurs puissent comprendre comment notre algorithme d'apprentissage automatique fait des prédictions, parmi elles on compte LIME et SHAP que nous avons présenté ici. Ces deux méthodes permettent aux utilisateurs non-experts de pouvoir avoir une idée des caractéristiques qui ont été prises en priorité afin de faire la prédiction. De même, on a pu constaté que plus le modèle fait de bonnes prédictions moins il est interprétable ce qui nous limite beaucoup.

Références

[1] What is a decision tree ?, 2017.

[2] Qu'est ce que l'algorithme knn ?, 2018. Accessed : March 27, 2023.

[3] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 12 2017.

[4] Sruthi E R. Understand random forest algorithms with examples (updated 2023), 2023.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you ?" : Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.