Approches agnostiques pour l'interprétation des modèles

Modélisation prédictive

Ricco Rakotomalala

Université Lumière Lyon 2

Préambule – Nécessité de l'interprétabilité des modèles

- Objectif : décrypter les « patterns » modélisés par l'algorithme d'apprentissage
- A savoir, comprendre le mécanisme d'affectation aux classes, globalement (au niveau du modèle), mais aussi individuellement (pour un individu à classer)
- Pourquoi et dans quelles circonstances cette nécessité est-elle importante ?
  - Explication : comprendre la « causalité » pour mieux l'expliquer
  - Validation : faire expertiser le mécanisme d'affectation par le métier, et aussi pour son amélioration (suggestions de variables, ou transformations)
  - Aspects réglementaires : justifier une affectation pour un individu à classer (ex. diagnostic de maladie, refus de crédit, soupçon de fraude, etc.)
  - Ou tout simplement pour obtenir l'adhésion de nos interlocuteurs (néophytes en machine learning mais experts dans leur domaine)
- « **Agnostique** » c.-à-d. l'outil doit être applicable à tout type de modèle prédictif

Plan

1. La référence : classifieurs linéaires
2. Permutation feature importance
3. ICE et PDP (Partial Dependence Plot)
4. SHAP (Shapley Addition exPlanations)
5. Conclusion
6. Références

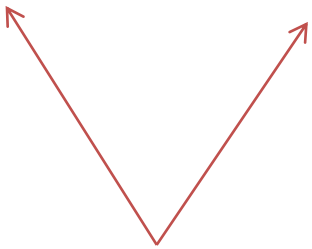
Cas de la Régression Logistique Binaire

INTERPRÉTATION POUR LES MODÈLES LINÉAIRES

Cadre Binaire Y ∈ {+, –} – Classifieur linéaire

Dans le cadre binaire, le classifieur (fonction de décision) s'exprime sous la forme d'une combinaison linéaire des variables explicatives (X_j).

$$D = a_0 + a_1 X_1 + \cdots + a_p X_p$$



D représente un « score », il traduit le degré d'appartenance à la modalité « + » de la cible Y, proportionnelle à π_+=P(Y=+ / X). Pour la régression logistique, D représente le $LOGIT = ln\frac{\pi_+}{1-\pi_+}$

- Le signe des coefficients « a_j » indique le sens de la relation de « X_j » avec π_+
- La valeur absolue de « a_j » indique l'intensité de l'association
- Pour peu que les « X_j » soient exprimées dans les mêmes unités, les « a_j » permettent de comparer les contributions des variables

Exemple : PIMA INDIANS DIABETES DATASET

	pregnant	bodymass	age
count	729.000000	729.000000	729.000000
mean	3.858711	32.469959	33.318244
std	3.357468	6.885098	11.753078
min	0.000000	18.200000	21.000000
25%	1.000000	27.500000	24.000000
50%	3.000000	32.400000	29.000000
75%	6.000000	36.600000	41.000000
max	17.000000	67.100000	81.000000

```
#lecture
import pandas
pima = pandas.read_excel("pima-indians-interpretability.xlsx")
#description
pima.describe()
#centrage-réduction
from sklearn.preprocessing import StandardScaler
std = StandardScaler()
Z = std.fit_transform(pima[pima.columns[:-1]],pima.diabete)
#régression logistique – scikit-learn
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(Z,pima.diabete)
#coefficients
print(pandas.DataFrame(lr.coef_[0],index=pima.columns[:-1],columns=['Coef.']))
```

Les variables ne sont pas exprimées dans les mêmes unités. D'où le centrage et (surtout) réduction.

PIMA (suite 1) – Interprétation des coefficients

Pregnant : nombre de fois où la personne est enceinte

Bodymass : indice de masse corporelle

Age : en années

	Coef.
pregnant	0.300044
bodymass	0.732107
age	0.404798

(1) Toutes agissent positivement c.-à-d. (a) une augmentation de la valeur d'une variable augmente le risque de diabète, ou (b) ceux qui présentent une valeur élevée ont plus de chances de développer la maladie (par rapport à ceux qui ont une valeur faible)

(2) « bodymass » est la plus influente. Ici une augmentation d'1 écart-type de « bodymass » induit une variation positive du LOGIT de 0.732107. L'étude de l'impact en écart-type (parce que variables centrées et [surtout] réduites) rend comparable les contributions des explicatives.

(3) Hypothèse restrictive ici, linéarité : l'augmentation du LOGIT consécutive à une variation d'1 écart-type est toujours la même quelle que soit la valeur prise par « bodymass » (qu'on soit gros ou mince, prendre du poids serait mauvais de la même manière… est-ce vrai ?).

PIMA (suite 2) – Processus d'affectation

N'oublions pas que les variables sont centrées et réduites !

Individu n°168 : (pregnant = -1.150 ; bodymass = 5.033 ; age = -0.623)

Calcul du LOGIT

0.300 x (-1.150)
+ **0.732 x 5.033**
+ 0.404 x (-0.623)
- 0.766 (a_0)

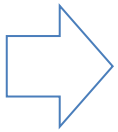
2.321

Que l'on obtient avec la fonction decision_function() dans Scikit-Learn par ex. (plus génériquement, distance à la frontière)

« Score » (probabilité d'affectation à Y = +)

$$\pi_+ = \frac{1}{1 + e^{-LOGIT}} = 0.9106$$

predict_proba() dans Scikit-Learn



Diagnostic : L'individu a de fortes de chances de développer la maladie à cause d'une corpulence (bodymass) excessive (5x l'écart-type au-dessus de la moyenne)

Classifieurs linéaires - Bilan

- Les propriétés des classifieurs linéaires sont connues et reconnues (faible variance, possibilité de régularisation pour les grandes dimensions)
- Leur popularité repose aussi sur la facilité d'interprétation des résultats qu'ils fournissent avec les coefficients :
  - Sens de la relation avec cible
  - Contribution des variables dans la décision (attention, ne capte pas les relations linéaires)
  - Hiérarchisation des contributions des explicatives dans la décision
  - Lecture du processus de prédiction pour un individu à classer

Comment faire pour les modèles « boîte noire » ? Non-linéaires ? Qui savent capter d'autres types de « patterns » par rapport aux classifieurs linéaires ?

Impact mesuré en variations sur un critère de performances

PERMUTATION FEATURE IMPORTANCE

Permutation Feature Importance

Principe : Mesurer le gap de performances lorsque l'on « désactive » une variable dans la prédiction. Plus la dégradation est élevée, plus la variable tient une place importante dans le modèle.

Etapes :

1. Construire un modèle prédictif Φ sur un échantillon, mesurer la performance ρ_{orig} (accuracy, ROC_AUC, autre…), en resubstitution ou sur un échantillon test
2. Pour chaque variable X_j de l'échantillon de travail :
   a. Mélanger aléatoirement les valeurs de la colonne X_j (les autres colonnes restent telles quelles, l'association de X_j avec la cible Y est rompue)
   b. Appliquer Φ sur le jeu de données modifié
   c. Mesurer l'indicateur de performances ρ_j
   d. Calculer le gap de performances (ex. $\delta_j = \frac{\rho_{orig} - \rho_j}{\rho_{orig}}$)
3. Possibilité de répéter plusieurs fois l'opération pour obtenir une mesure plus stable de δ_j

Remarque : il peut y avoir des variantes de calcul selon les librairies utilisées (ex. Scikit-Learn, etc.)

Exemple – Régression logistique

Critère ROC_AUC dans cet exemple, parce que la méthode sait fournir les probas d'appartenances aux classes.

```
#*****************************
#permutation feature importance
#cf. lr (objet généré en page 6)
from sklearn.inspection import permutation_importance
imp_lr = permutation_importance(lr,Z,pima.diabete,n_repeats=100,random_state=0,scoring='roc_auc')
print(pandas.DataFrame(imp_lr.importances_mean,index=pima.columns[:-1],columns=['P.Importance']))
```

	P.Importance
pregnant	0.025643
bodymass	0.130851
age	0.052438

Ce sont les positionnements relatifs des variables qui sont importants.

En comparaison, les coefficients standardisés de la régression, dont le calcul tient compte de la nature de la méthode.

	Coef.
pregnant	0.300044
bodymass	0.732107
age	0.404798

Exemple – SVM, noyau RBF

Dans un SVM avec un noyau RBF, aucun indicateur interne à la méthode ne permet de situer les contributions des variables.

```
#SVM avec noyau RBF
from sklearn.svm import SVC
#demander le calcul des probas pour pouvoir utiliser
#le critère ROC_AUC lors du calcul de l'importance
svm = SVC(kernel='rbf',probability=True)
svm.fit(Z,pima.diabete)
#feature importance
imp_svm = permutation_importance(svm,Z,pima.diabete,n_repeats=100,random_state=0,scoring='roc_auc')
print(pandas.DataFrame(imp_svm.importances_mean,index=pima.columns[:-1],columns=['P.Importance']))
```

	P.Importance
pregnant	0.057986
bodymass	0.125691
age	0.116093

Dans ce modèle non-linéaire, « âge » a une influence proche de celle de « bodymass » dans l'explication du « diabète ».

Permutation Feature Importance - Bilan

- A priori, δ_j devrait toujours être positif ou nul
- La répétition permet une mesure plus stable (ex. médiane, moyenne), elle donne également une idée de la distribution de δ_j.
- Coût calculatoire faible puisque l'apprentissage n'est effectué qu'une seule fois. C'est la prédiction qui est mise à contribution. Elle est très rapide pour la grande majorité des méthodes sauf cas pathologiques (ex. K-NN).
- Mesurer l'indicateur sur un échantillon à part (TEST) est préférable, on évalue ainsi l'importance des variables dans la prédiction. Mais TRAIN et TEST devraient être cohérents, une différence de résultats est symptomatique d'une situation de surapprentissage.
- L'approche permet une comparaison des contributions des variables sans qu'elles soient nécessairement exprimées sur la même échelle.
- Neutraliser individuellement les variables ne permet pas d'évaluer les possibles interactions.

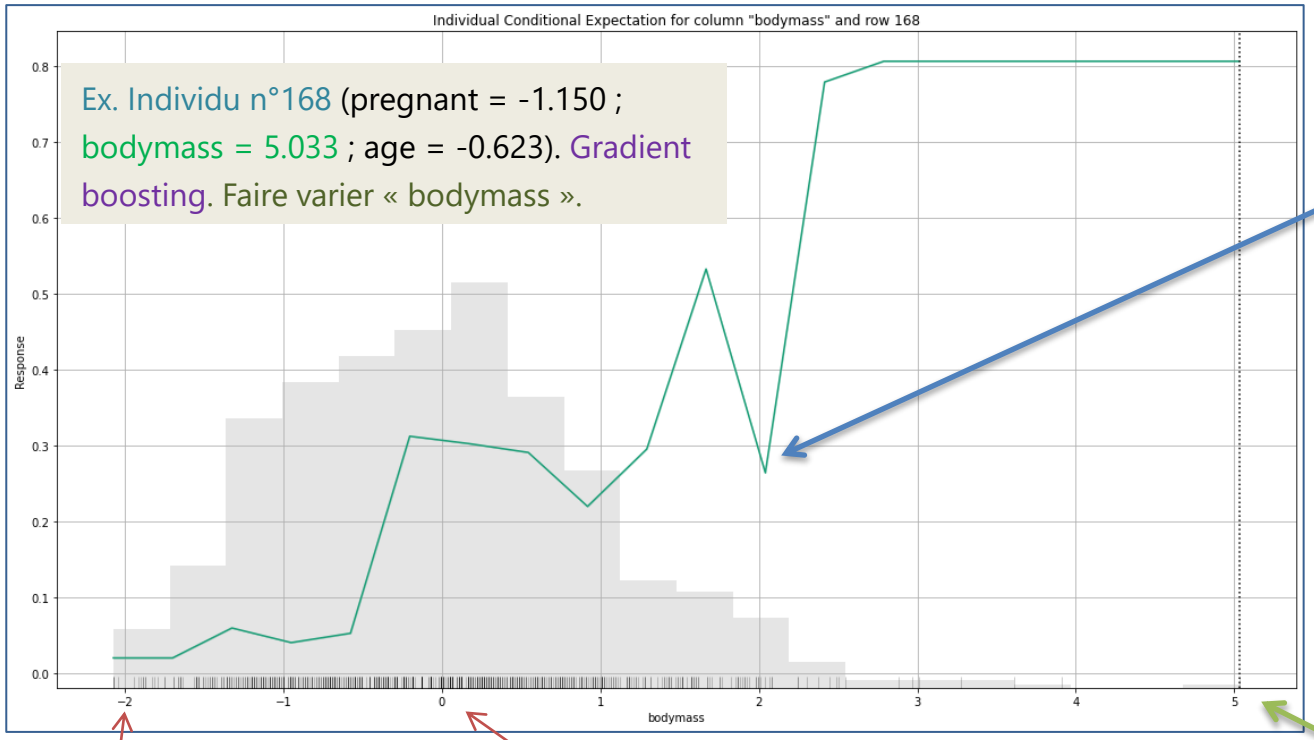
Comportement d'une variable (à valeurs fixées des autres variables)

INDIVIDUAL CONDITIONAL EXPECTATION (ICE)

PARTIAL DEPENDENCE PLOT (PDP)

Individual Conditional Expectation (ICE)

Principe pour une variable : Pour un individu à traiter, mesurer la valeur d'un indicateur de discrimination (fonctions de décision [distance à la frontière de séparation], probabilité d'appartenance à la classe cible) en faisant varier les valeurs de la variable à étudier sur son étendue entière, à valeur fixées des autres variables.



Ex. Individu n°168 (pregnant = -1.150 ; bodymass = 5.033 ; age = -0.623). Gradient boosting. Faire varier « bodymass ».

On dispose d'un outil de diagnostic (expertisée par le gradient boosting) : Il faudrait une très forte perte de poids pour diminuer le risque de diabète

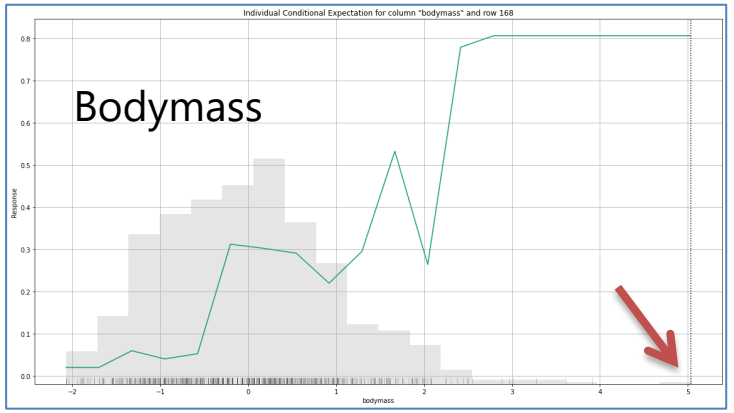
Si « bodymass = -2 », alors $\pi_+ \approx$ 0.02 pour l'individu n°168 (toutes choses égales par ailleurs)

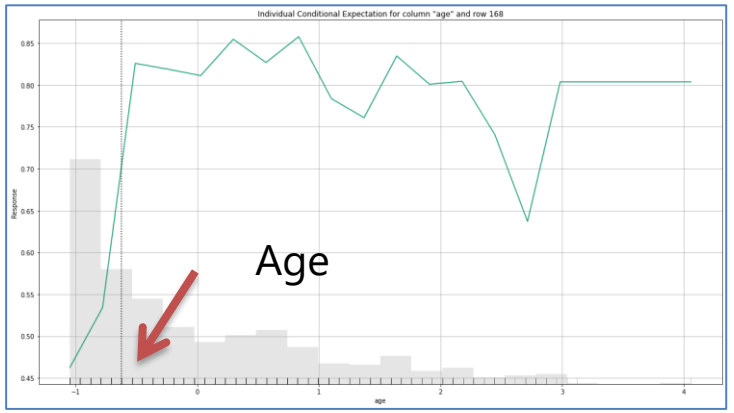
Si « bodymass = 0 », alors $\pi_+ \approx$ 0.3 pour l'individu n°168

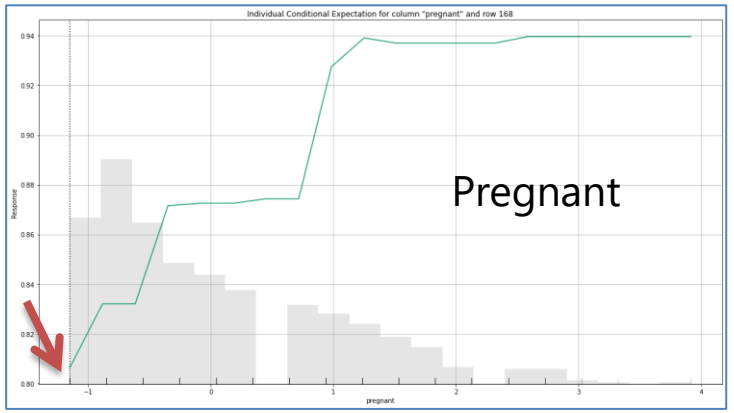
Valeur actuelle (0.503) de « bodymass » pour l'individu n°168

ICE – Exemple avec H2O (Gradient Boosting)

```
#H2O
import h2o
print(h2o.__version__)
#démarrage
h2o.init(nthreads=7)
#jeu de données avec les variables standardisées
D = pandas.DataFrame(Z,columns=pima.columns[:-1])
D['diabete'] = pima.diabete
#format H2O des données
DH = h2o.H2OFrame(D)
#instancier un gradient boosting
from h2o.estimators import H2OGradientBoostingEstimator
gbm = H2OGradientBoostingEstimator(seed=0)
gbm.train(y='diabete',training_frame=DH)
#expertise pour l'individu n°168
#pregnant = -1.15, bodymass = 5.03, age = -0.62
gbm.explain_row(DH,168)
```



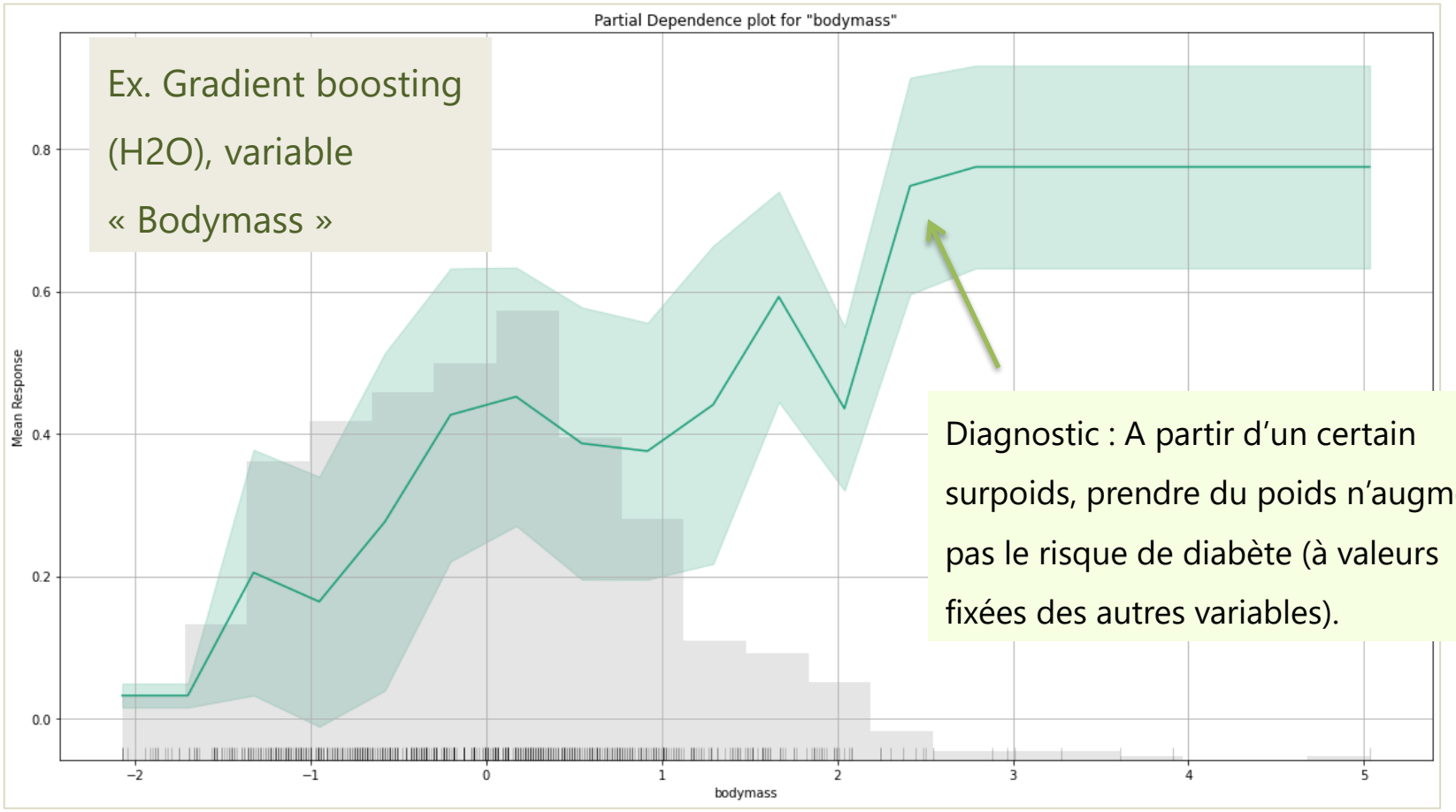




Ex. de lecture pour l'individu n°168 : les éventuelles grossesses à venir vont augmenter le risque de diabète, mais vieillir n'aura pas d'impact.

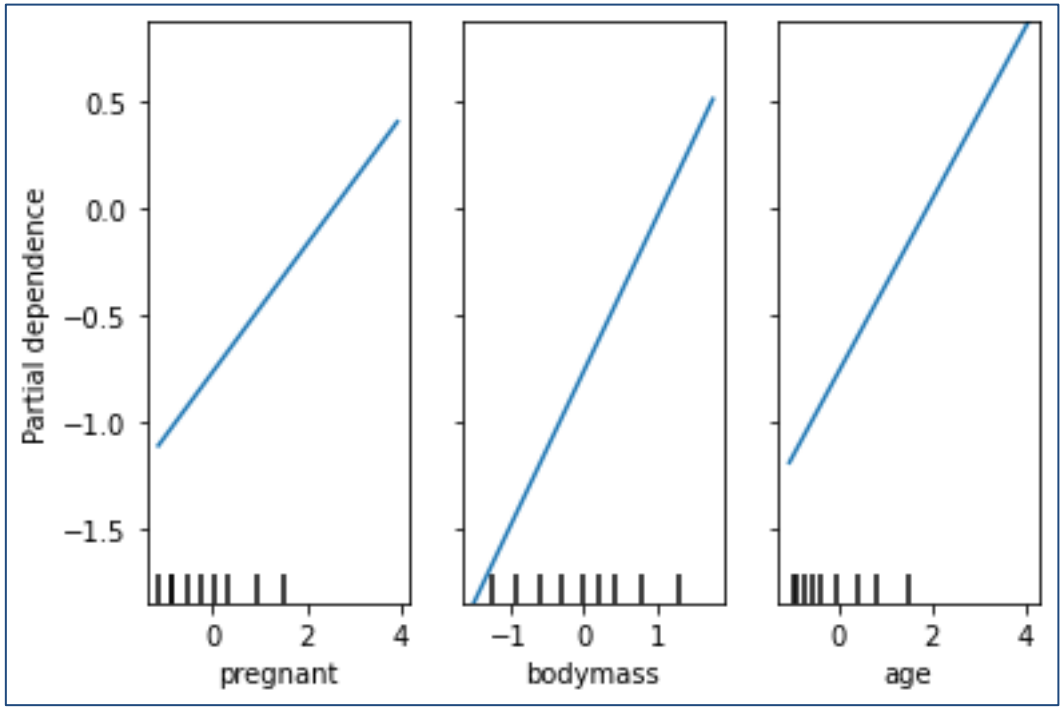
Partial Dependence Plot (PDP)

Principe pour une variable : Agrégation (ex. moyenne) des ICE de l'ensemble des individus de la base de données. Pour obtenir une vue plus globale de la relation entre la variable et la caractéristique de discrimination (distance à la frontière ou probabilité de la classe cible)



PDP – Exemple pour la régression logistique

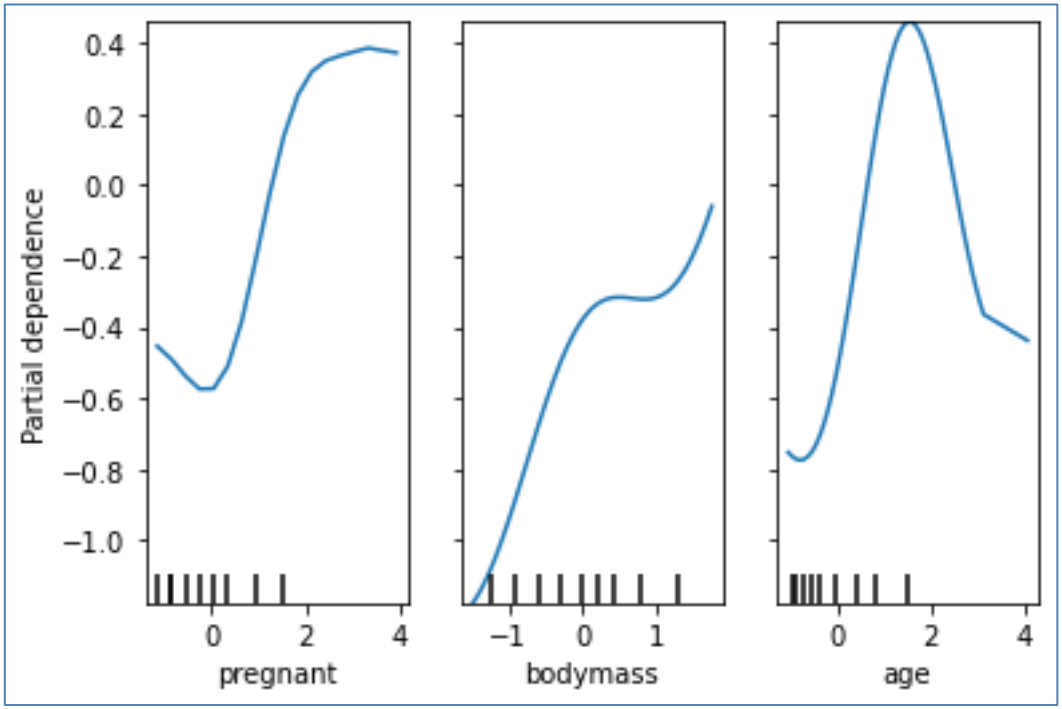
```
#partial dependence plot - régression logistique
from sklearn.inspection import PartialDependenceDisplay
PartialDependenceDisplay.from_estimator(lr, Z, features=[0,1,2],
feature_names=pima.columns[:-1], kind='average', response_method='decision_function')
```



La régression logistique est un classifieur linéaire, les fonctions de décision (LOGIT ici) forment des droites forcément. Les coefficients sont tous positifs (page 7), les relations sont croissantes.

PDP – Exemple pour le SVM (RBF)

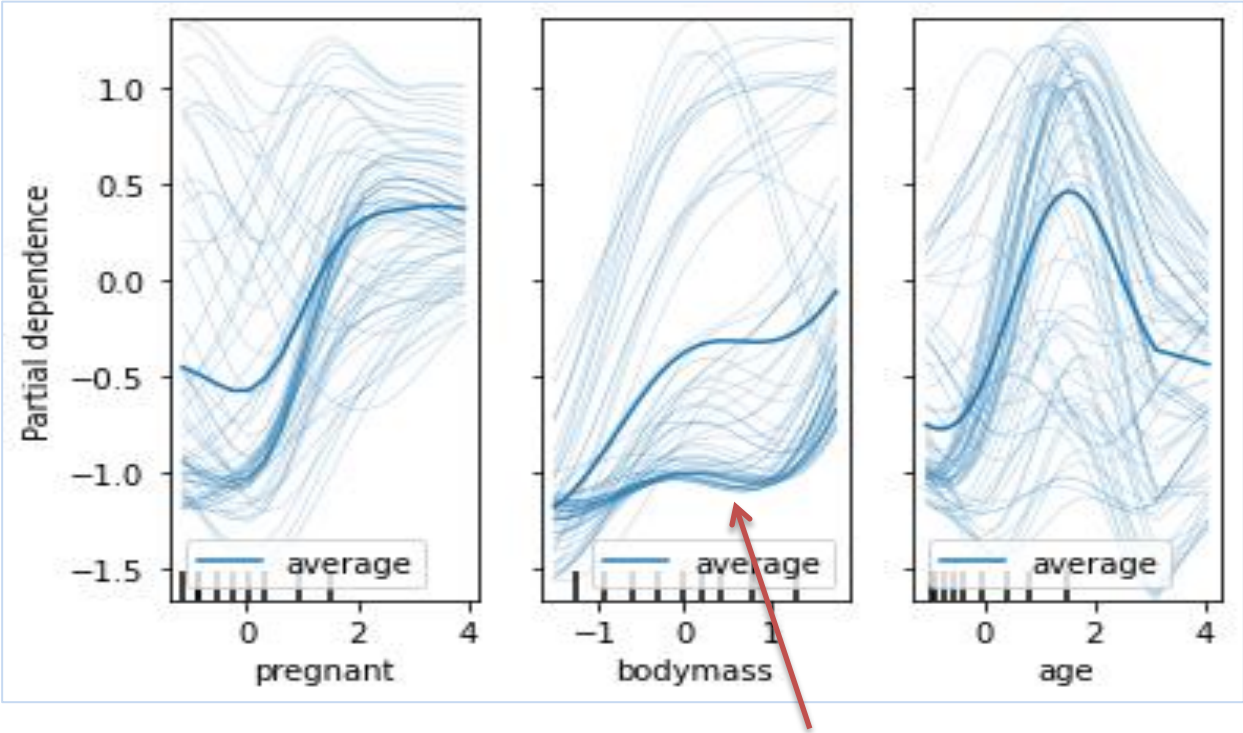
```
#partial dependence plot - SVM
PartialDependenceDisplay.from_estimator(svm, Z, features=[0,1,2],
feature_names=pima.columns[:-1], kind='average', response_method='decision_function')
```



Là pour le coup, on voit bien que SVM(RBF) est capable de capter les relations non-linéaires. Ex. à partir d'un certain âge, vieillir induit une réduction du risque de diabète (pourquoi ? l'expert métier entre en jeu ici)

PDP – Apprécier la « distribution » des courbes

```
#partial dependence plot - SVM (échantillons - distribution)
#en plus de la courbe « moyenne », 10% des individus sont tracés ici
PartialDependenceDisplay.from_estimator(svm, Z, features=[0,1,2], feature_names =
pima.columns[:-1], kind='both', subsample=0.1, response_method='decision_function')
```

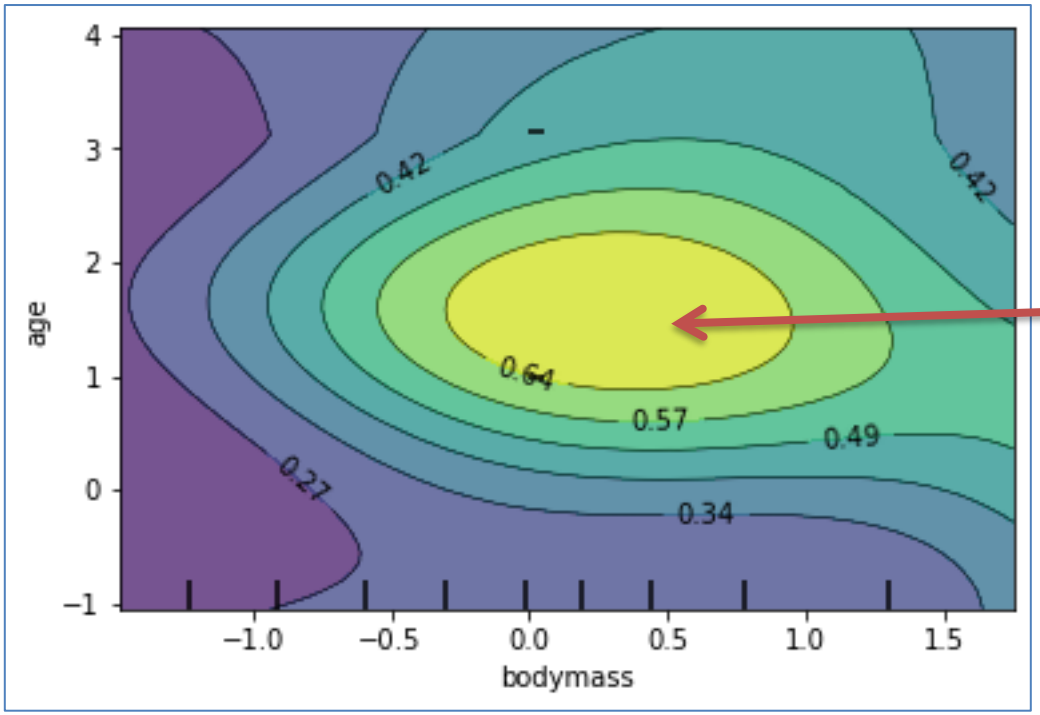


On se rend compte que la courbe « moyenne »

peut masquer des cas individuels assez disparates.

PDP – Prise en compte des interactions

```
#interaction (bodymass et age) - SVM(RBF)
PartialDependenceDisplay.from_estimator(svm, Z, features=[(1,2)],
feature_names=pima.columns[:-1], kind='average', response_method='predict_proba')
```



Modélisé par SVM(RBF), on constate que la conjonction d'être plus âgé (que la moyenne) et d'avoir un léger surpoids (idem) induit la situation la plus risquée (probabilité forte de diabète).

Contribution des variables dans l'affectation des classes en prédiction

SHAP : SHAPLEY ADDITIVE EXPLANATIONS

Shapley Value - Idée

Objectif : Indiquer la contribution (qui peut être positif ou négatif) d'une variable (X_j) à la probabilité π_+ d'appartenir à la classe cible pour un individu à classer n°i. Avec pour idée de mettre en évidence l'écart par rapport à la « prédiction moyenne » du modèle. Cet outil est très important en déploiement, lorsque nous souhaitons justifier l'attribution d'une classe à un individu supplémentaire.

Classifieur linéaire : La « prédiction moyenne » (du modèle trivial) est représentée par la constante (a_0). Les variables étant centrées et réduites, la contribution de (X_j) dans le LOGIT pour l'individu n°i est obtenue avec ($a_j \times x_{ij}$). Voir page 8. Elle peut être positive ($\uparrow \pi_+$) ou négative ($\downarrow \pi_+$).

Comment transposer l'idée de manière agnostique c.-à-d. sans connaître les caractéristiques internes du classifieur étudié ?

Shapley Value - Calcul

Calcul : Beaucoup de fantasmes autour de la théorie des jeux.

Mais en pratique, il s'agit surtout de mesurer l'effet de la désactivation de la variable (X_j) dans la prédiction de l'individu n°i.

Approximate Shapley estimation for single feature value:

- Output: Shapley value for the value of the j-th feature
- Required: Number of iterations M, instance of interest x, feature index j, data matrix X, and machine learning model f
  - For all m = 1,…,M:
    - Draw random instance z from the data matrix X
    - Choose a random permutation o of the feature values
    - Order instance x: $x_o = (x_{(1)}, \ldots, x_{(j)}, \ldots, x_{(p)})$
    - Order instance z: $z_o = (z_{(1)}, \ldots, z_{(j)}, \ldots, z_{(p)})$
    - Construct two new instances
      - With j: $x_{+j} = (x_{(1)}, \ldots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
      - Without j: $x_{-j} = (x_{(1)}, \ldots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \ldots, z_{(p)})$
    - Compute marginal contribution: $\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$
- Compute Shapley value as the average: $\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$

C. Molnar, « Interpretable Machine Learning », Section 9.5.3.

« Vraie » valeur de (X_j) pour l'individu à classer vs. Valeur de substitution récupérée chez un autre individu pris au hasard.

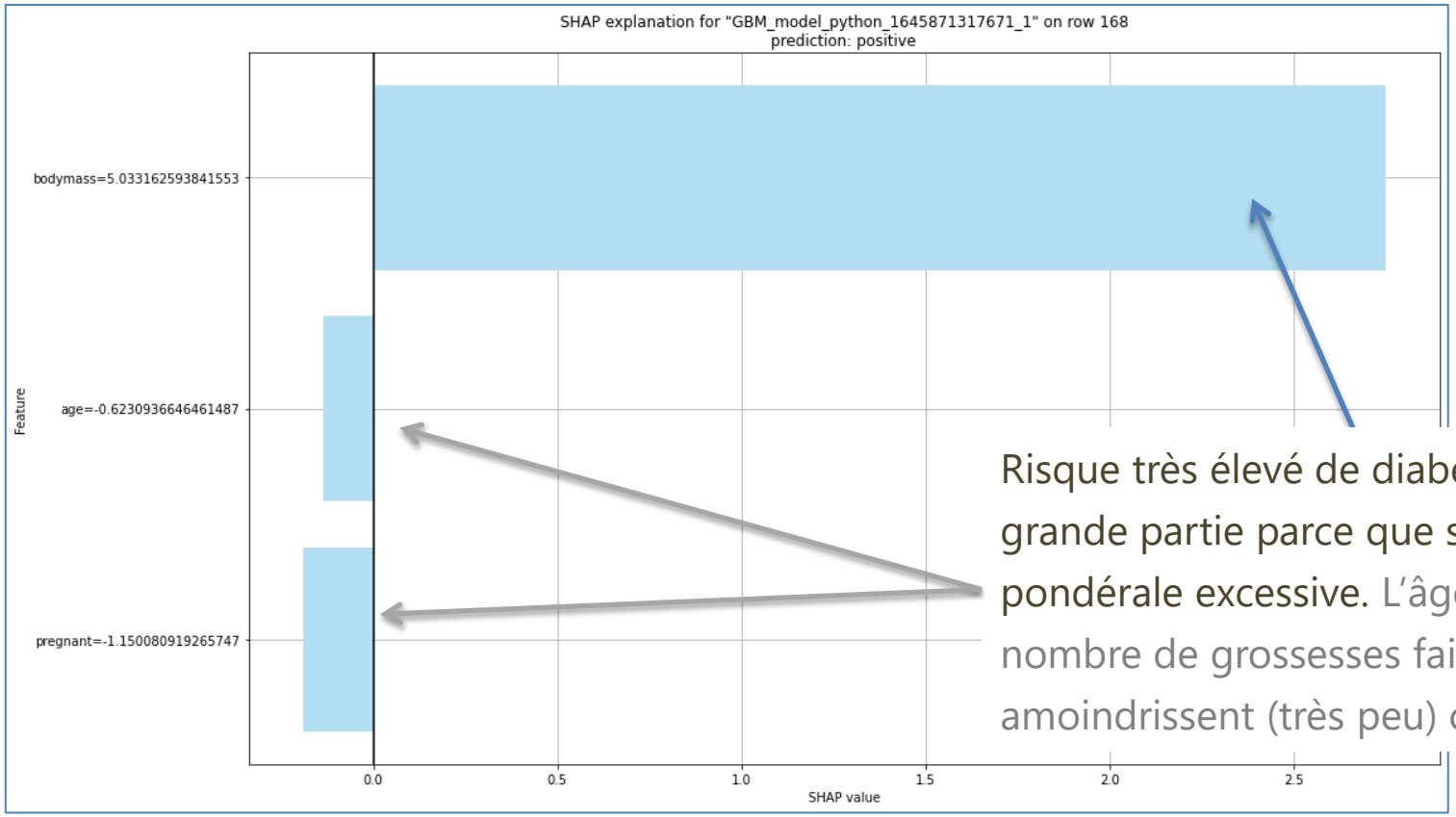
On peut utiliser (π_+) en guise de $\hat{f}$, ça peut être également la « decision function »

Réitérer M fois l'opération et prendre la moyenne pour disposer d'une mesure plus stable.

Exemple – Individu n°168 – Gradient Boosting Machine (H2O)

Individu n°168 (pregnant = -1.150 ; bodymass = 5.033 ; age = -0.623).

$\pi_+ = 0.8064$

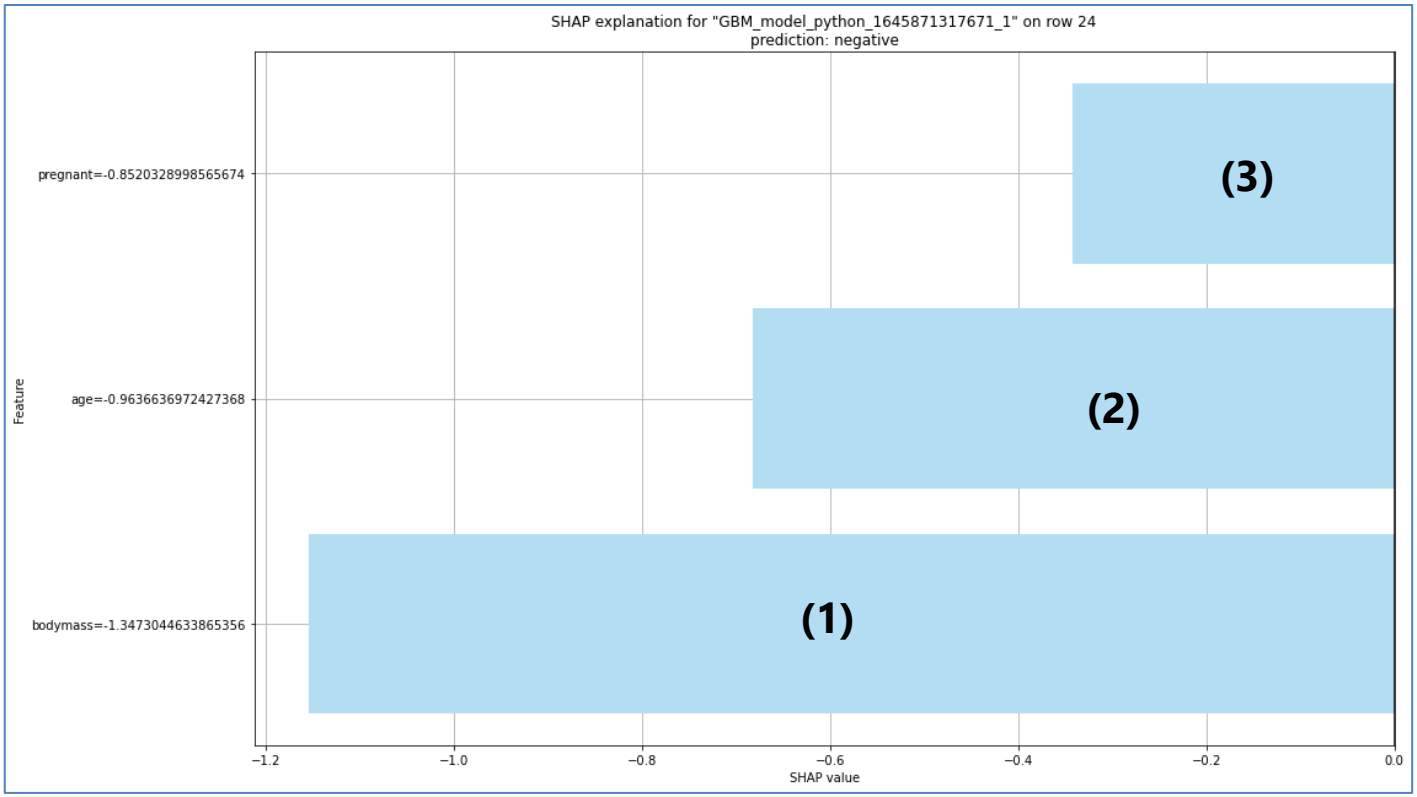


Risque très élevé de diabète, en très grande partie parce que surcharge pondérale excessive. L'âge et le nombre de grossesses faibles amoindrissent (très peu) ce risque.

Exemple – Individu n°24 – Gradient Boosting Machine (H2O)

Individu n°24 (pregnant = -0.852 ; bodymass = -1.347 ; age = -0.963).

$\pi_+ = 0.040$

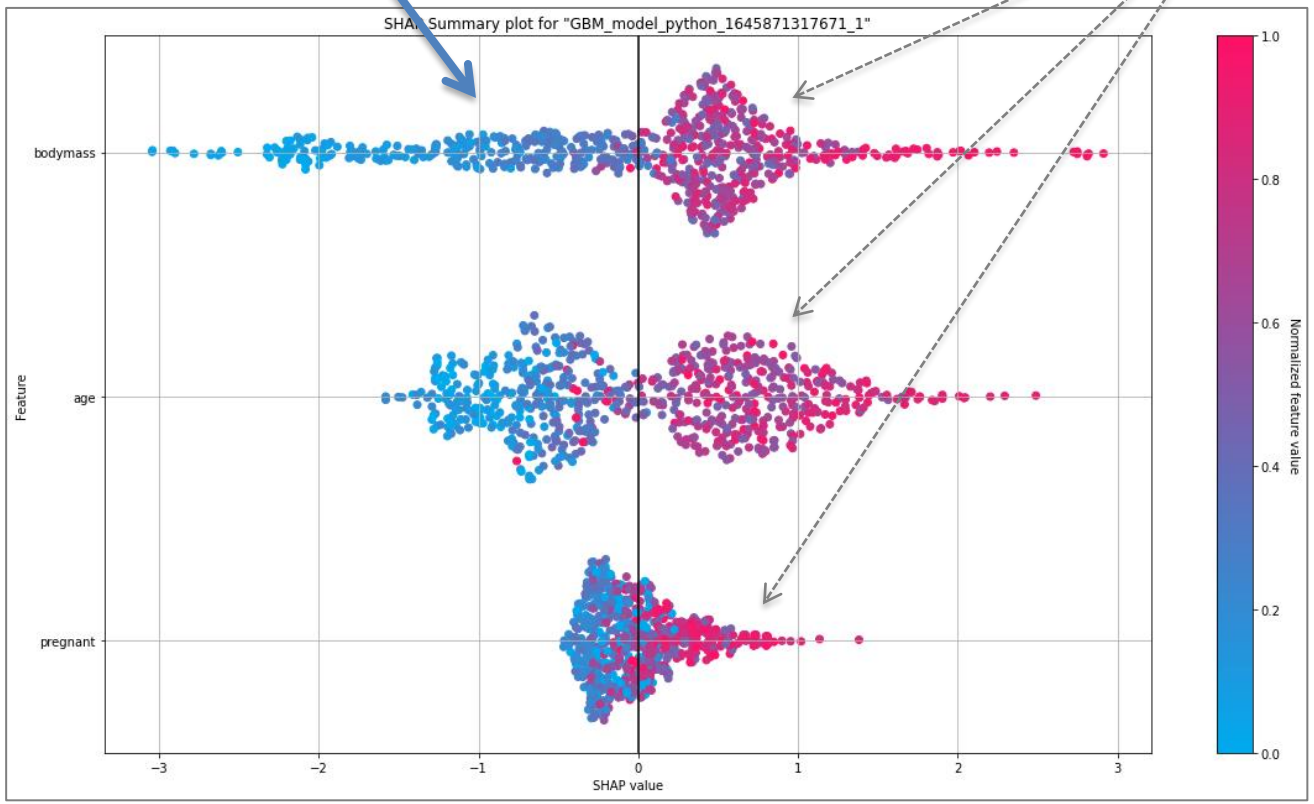


Toutes les caractéristiques de l'individu n°24 concourent au faible risque de diabète, mais à des degrés divers : (1) bodymass d'abord fortement, (2) âge ensuite, (3) pregnant enfin.

Shapley Values – Vue globale sur le modèle (GBM – H2O)

Chaque point = 1 individu. « En moyenne » (visuellement), les valeurs faibles de « bodymass » (bleu) induisent un faible risque de diabète (à gauche sur l'abscisse).

L'étalement des valeurs montre aussi le pouvoir de discrimination de la variable comparativement aux autres (dans la distinction des diabètes + vs. diabètes -). Voir cohérence avec « feature importance ».



Rouge, valeurs élevées de la variable (normalisée entre 0 et 1)

Bleu, valeurs faibles de la variable (normalisée entre 0 et 1)

Risque faible de diabète

Risque élevé de diabète

CONCLUSION

Conclusion

- L'interprétation des modèles est cruciale dans certains domaines (santé, etc.)
- Certaines méthodes sont nativement « lisibles » (ex. classifieurs linéaires, mais aussi les approches à base de règles)
- La situation est moins reluisante pour certains modèles « boîtes noires » (ex. méthodes ensemblistes, réseaux de neurones profonds, méthodes à base de voisinage)
- Les techniques agnostiques (non-dépendantes de l'algorithme d'apprentissage) permettent de mieux discerner le rôle des variables dans le modèle et dans le processus prédictif, de mieux comprendre « ce qu'il y a dans la boîte »
- Il y a quand-même un bémol fort, les variables sont traitées individuellement dans les techniques présentées, on ne distingue pas les possibles interactions (ex. dans le PDP, faire varier le nombre de grossesses de 0 à 20 n'est pas réaliste pour une personne de 18 ans)

RÉFÉRENCES

Références

- (« La » référence) C. Molnar, « Interpretable Machine Learning – A guide for Making Black Box Models Explainable », version utilisée : 21/02/2022.
- Tutoriel Tanagra, « (Vidéo) Outils pour l'interprétation des modèles », avril 2021.
- Tutoriel Tanagra, « (Vidéo) Model Explainability par H2O », avril 2021.
- Tutoriel Tanagra, « Importance des variables dans les modèles », février 2019.
- Tutoriel Tanagra, « Graphique de dépendance partielle – R et Python », avril 2019.