Etude globale sur l'interprétabilité des modèles du Machine Learning

Abdelraouf KESKES
Aurélia DOLO

January 21, 2020

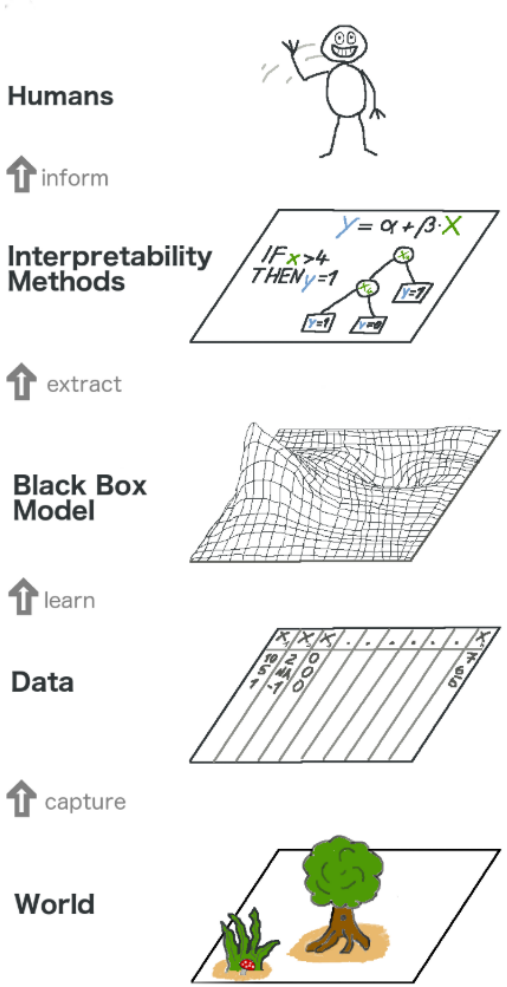
SCIENCES SORBONNE UNIVERSITÉ

Sommaire

Introduction

Problématiques

- Métriques (accuracy, F1-score,...) peu informatives , pas très fiables
- Elles ne répondent ni au "Comment?" ni au "Pourquoi?"
- Comment trouver ce qui est pertinent et informatif pour expliquer le modèle de machine learning ?



Définitions et objectifs

Définition : Interprétabilité

- explication compréhensible par un humain quelconque
- pourquoi un résultat plutôt qu'un autre ?
- le plus souvent, une explication par instance

Objectifs

- Développement d'applications à faible risque (domaines sensibles)
- Debuggage avancé pour les développeurs de modèles ML et amélioration des performances
- Développement de modèles très robustes et fiables
- AI fairness

Notions générales

Notions fondamentales

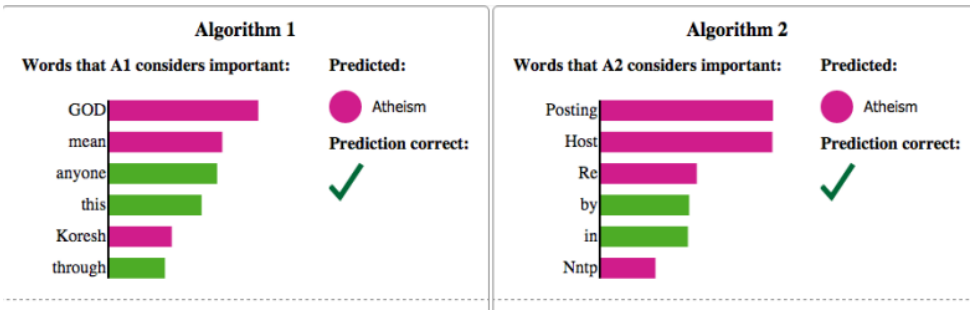
- Il existe des systèmes qui n'ont pas besoin d'interpretabilité !
- Nous avons 2 Modèles (**"ML"** et **"EXplainability"**)
- En général, les features du ML ne sont pas forcément les mêmes que ceux du EX (par ex, NLP => **ML**:word- embeddings, **EX**:Présence des mots)

2 familles de modèles ML

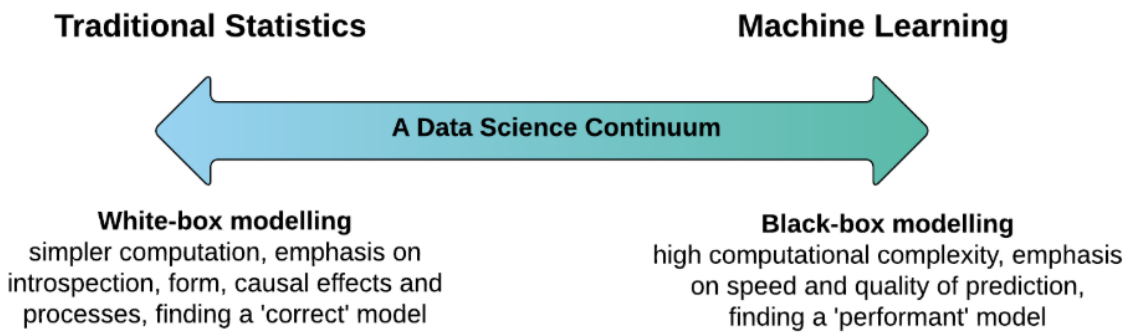
– **White box** : interprétables de nature mais peu performants et expressifs (Modèles linéaires, Arbres de décision,...)

– **Black box** : beaucoup plus performants, complexes, et expressifs →Impossible à comprendre leurs décisions

Notions générales



Source : (LIME paper)



Source : (Applied AI)

Notions générales

Model-Agnostic vs Model-specific

- **Agnostic** : un modèle d'interprétabilité qui marche pour tous les modèles ML en le considèrant comme une "black box"
- **Specific** : un modèle d'interprétabilité qui explique qu'une (ou quelques) certaine famille de modèles par définition (Ex de familles : "Neural Networks", "Tree-Based", ...)

Scope : Global vs Local

- **Local** : expliquer une certaine prediction pour une certaine instance
- **Global** : expliquer le comportement général du modèle

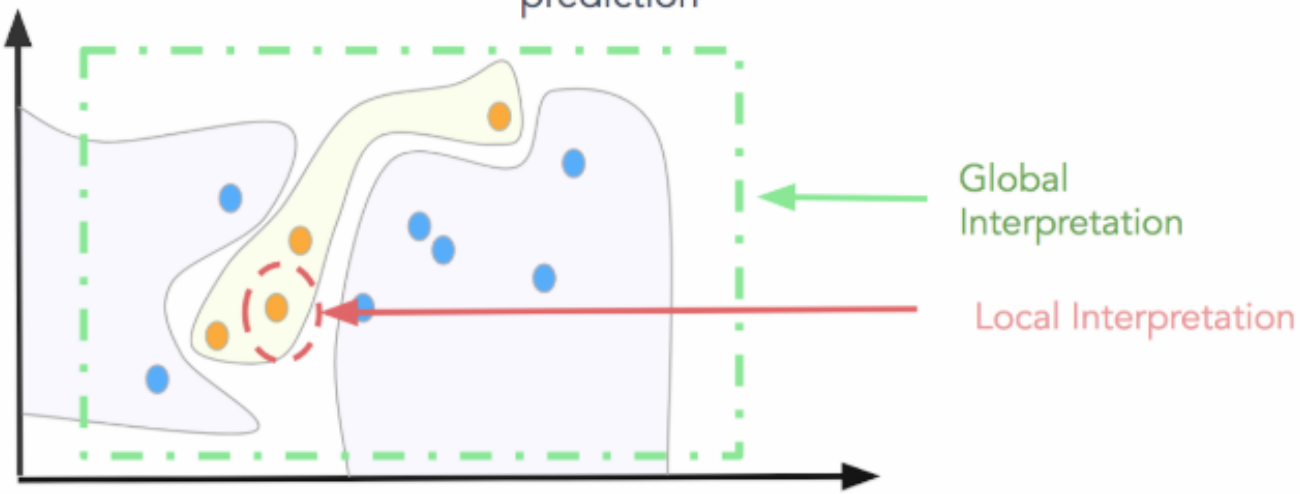
Notions générales

Global Interpretation

Being able to explain the conditional interaction between dependent(*response*) variables and independent(*predictor, or explanatory*) variables based on the complete dataset

Local Interpretation

Being able to explain the conditional interaction between dependent(*response*) variables and independent(*predictor, or explanatory*) variables wrt to a single prediction



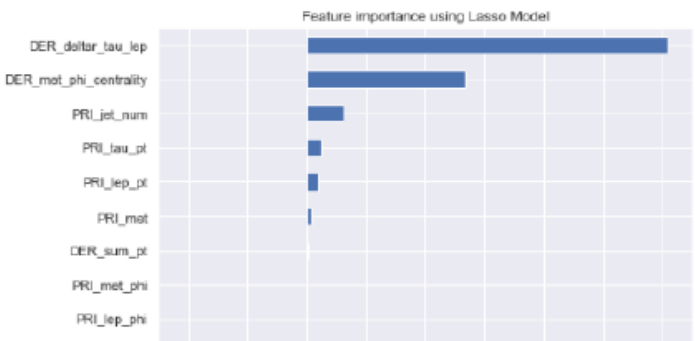
Source : (Datascience.com)

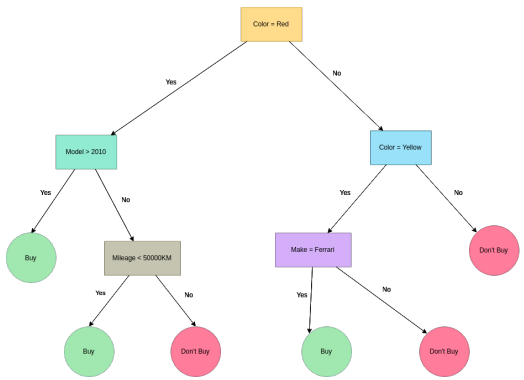
Modèles interprétables

Ce sont des modèles simples (White box) ...

Le modèle ML est lui même le modèle EX (EXplanability)

- Modèles linéaires (**)
- Arbres de décisions(**)
- Autres(*): Règles de décision , Naive Bayes , KNN, ...



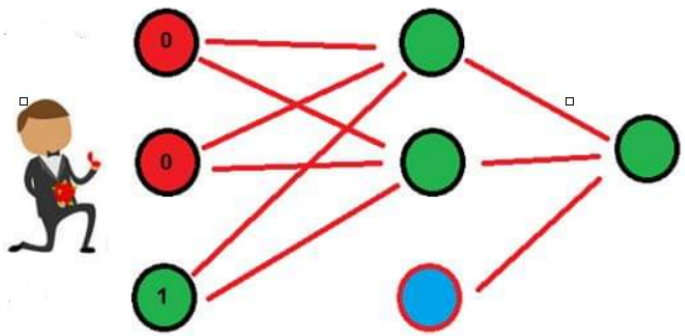


Modèles Complexes : généralités

Ce sont des modèles très performants mais "Black box" ...

Nous aurons besoin d'un modèle à part pour les expliquer

- Réseaux de neurones
- Bagging , Boosting , Random forests et autres



XGBoost
RandomForests
ConvolutionalNets
RecurrentNeuralNetwork
FeedForwardNets
LightGBM
GANs

LIME

LIME (Local Interpretable Model-agnostic Explanations)

Idée : pour une instance on approxime le modèle ML localement

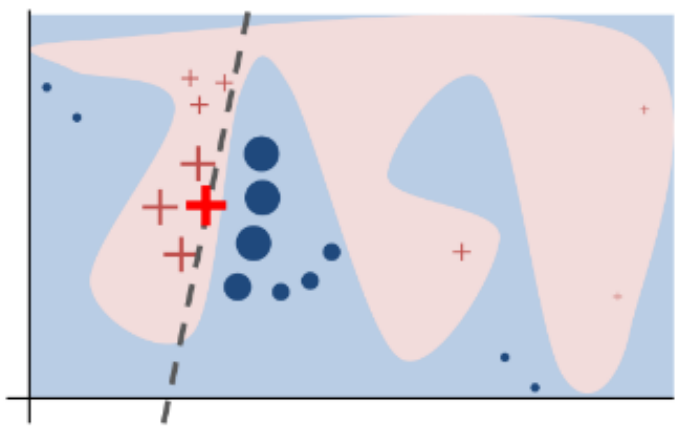
Framework

- une instance $x \in R^d$ et sa representation EX $x' \in \{0,1\}^{d'}$
- le modèle ML $f : R^d \mapsto R$, et le modèle EX $g : \{0,1\}^d \mapsto R$
- des instances samplés $z_i \in R^d$, leur rep z_i' et un kernel de proximité $\pi_x(z)$
- Problème d'optimisation : $\xi(x) = \arg\min_{g \in G} [\mathcal{L}(f, g, \pi_x) + \Omega(g)]$
- Coût MSE pondéré par le kernel : $\mathcal{L}(f, g, \pi_x) = \sum_i \pi_x(z_i)(f(z_i) - g(z_i'))^2$

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

```
Z ← {}
for i ∈ {1, 2, 3, ..., N} do
    z_i' ← sample_around(x')
    Z ← Z ∪ ⟨z_i', f(z_i), π_x(z_i)⟩
end for
w ← K-Lasso(Z, K)  ▷ with z_i' as features, f(z) as target
return w
```



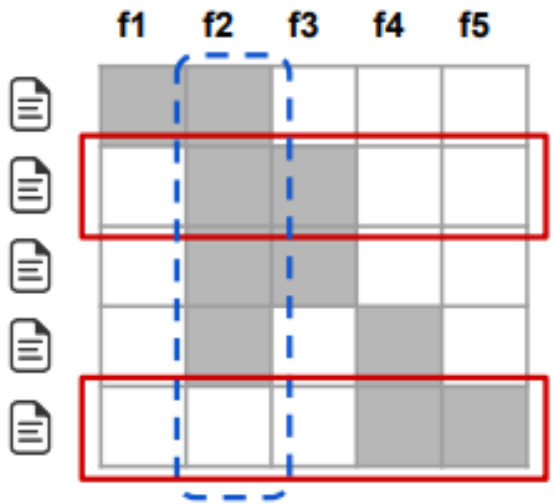
SP-LIME : Extension globale de LIME

SP-LIME (Sub-modular Pick LIME)

Idée : repérer des instances représentatives et faire un LIME sur ces dernières pour essayer d'approximer le modèle **globalement**

Algorithm 2 Submodular pick (SP) algorithm

Require: Instances X, Budget B
  for all $x_i \in X$ **do**
    $\mathcal{W}_i \leftarrow$ **explain**(x_i, x_i') ▷ Using Algorithm 1
  end for
  for $j \in \{1 \dots d'\}$ **do**
    $I_j \leftarrow \sqrt{\sum_{i=1}^{n} |\mathcal{W}_{ij}|}$ ▷ Compute feature importances
  end for
  $V \leftarrow \{\}$
  while $|V| < B$ **do** ▷ Greedy optimization of Eq (4)
    $V \leftarrow V \cup \text{argmax}_i\, c(V \cup \{i\}, \mathcal{W}, I)$
  end while
  return V



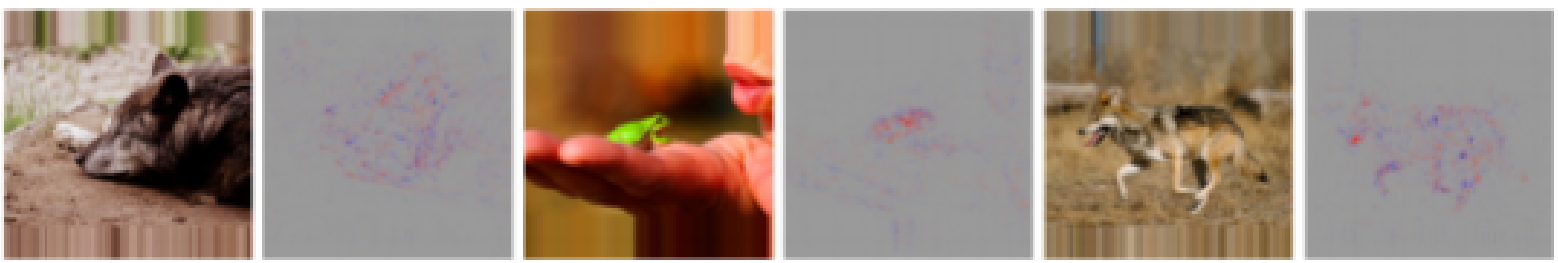
LRP

LRP (Layer-wise Relevance Propagation)

Idée : Backpropager les signaux de sortie d'un réseau de neurones jusqu'à la couche des inputs

Framework

- la sortie aj d'un neurone j est la non-linéarité g : $a_j = g(\sum_i w_{ij} * a_i + b)$
- une instance x, un modèle f, une couche l, une dimension p, un score de relevance R_p^l, tel que $f(x) \approx \sum_p R_p^{(1)}$ (la somme des contributions de chaque feature d'entrée)
- features p avec $R_p^{(1)} < 0$ sont contre la présence de la sortie et inversement ($R_p^{(1)} > 0$)



Source : (LRP with Local Renormalization Layers paper)

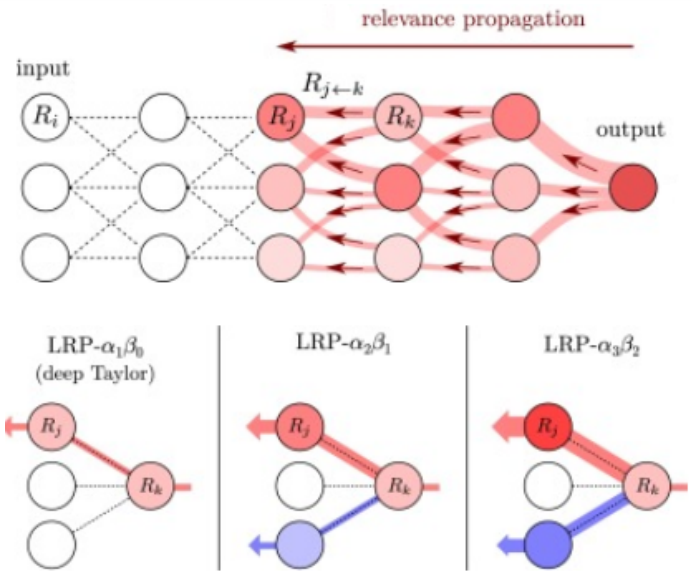
LRP

Détails ...

- le score de relevance pour :
  - les neurones de **sortie** $l = L$ est : $R_o^{(L)} = f(x)$ et de **l'entrée** est : $R_p^{(L)}$
  - les neurones **intermédiaires** (back-prop) : $R_i^{(l)} = \sum_{j \in (l+1)} R_{i \leftarrow j}^{(l,l+1)}$
- toutes les variations de ce modèle EX son basées sur le calcul de $R_{i \leftarrow j}^{(l,l+1)}$ ...

Name	Formula	Usage	DTD
LRP-0 [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{0,j} a_j w_{jk}} R_k$	upper layers	✓
LRP-ϵ [7]	$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{0,j} a_j w_{jk}} R_k$	middle layers	✓
LRP-γ	$R_j = \sum_k \frac{a_j (w_{jk} + \gamma w_{jk}^+)}{\sum_{0,j} a_j (w_{jk} + \gamma w_{jk}^+)} R_k$	lower layers	✓
LRP-$\alpha\beta$ [7]	$R_j = \sum_k \left(\alpha \frac{(a_j w_{jk})^+}{\sum_{0,j} (a_j w_{jk})^+} - \beta \frac{(a_j w_{jk})^-}{\sum_{0,j} (a_j w_{jk})^-} \right) R_k$	lower layers	×⋆
flat [30]	$R_j = \sum_k \frac{1}{\sum_j 1} R_k$	lower layers	×
w^2-rule [36]	$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j$	first layer ($\mathbb{R}^d$)	✓
$z^\mathcal{B}$-rule [36]	$R_i = \sum_j \frac{x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_i x_i w_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-} R_j$	first layer (pixels)	✓

(⋆ DTD interpretation only for the case $\alpha = 1, \beta = 0$.)



DeepLIFT

DeepLIFT (Deep Learning Important FeaTures)

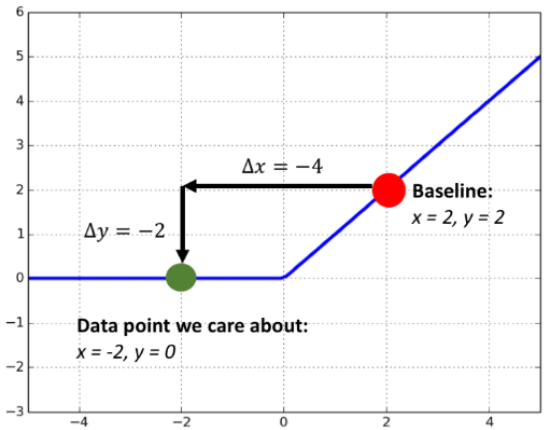
Idée : Remplacer la back-prop **des dérivées** par celle **des différences**

Détails

- 2 problèmes majeurs du gradient : **Saturation, ReLU(virer les Negatifs, Discontinuité)**
- On s'intéresse pas au gradient**(comment** y **change lorsque** x **change infinitésimalement)**
- On s'intéresse à la pente **(comment** y **change quand** x **diffère de sa référence** x_{ref}**)**
- $gradient = \frac{\partial y}{\partial x} \Rightarrow slope = \frac{y - y_{ref}}{x - x_{ref}} = \frac{\Delta y}{\Delta x}$
- maintien de la "chain rule" : $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial z} * \frac{\partial z}{\partial x} \Rightarrow \frac{\Delta y}{\Delta x} = \frac{\Delta y}{\Delta z} * \frac{\Delta z}{\Delta x}$
- $Feature_i$ Importance : $x_i \times \frac{\partial y}{\partial x_i} \Rightarrow (x_i - x_i^{ref}) \times \frac{\Delta y}{\Delta x_i}$
- Choisir une référence ?
  - les neurones de l'entrée : **expertise du domaine** (Ex: MNIST => images de 0)
  - les neurones intérmediaires et de sortie , on **forward** les entrées de références

DeepLIFT

$$y = ReLU(x) = max(0, x)$$



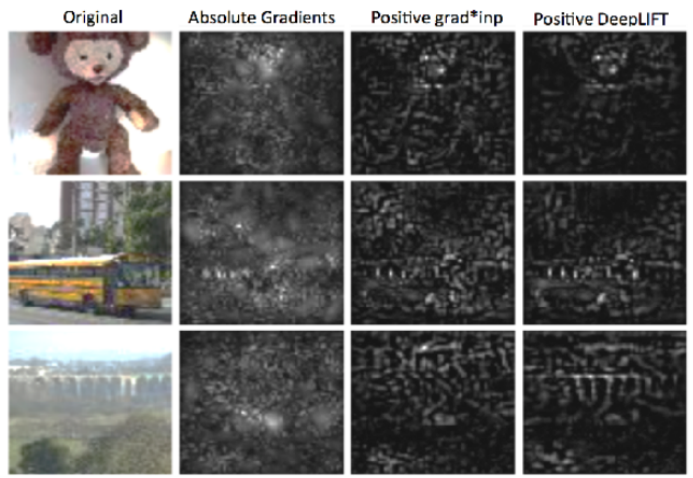
1. Calculating the slope

$$\frac{\Delta y}{\Delta x} = \frac{-2}{-4} = 0.5$$

2. Finding the feature importance

$$\Delta x \times \frac{\Delta y}{\Delta x} = -4 \times 0.5 = -2$$

Source : (Gabriel Tseng Medium Blogs)



Source : (DeepLIFT paper)

SHAP

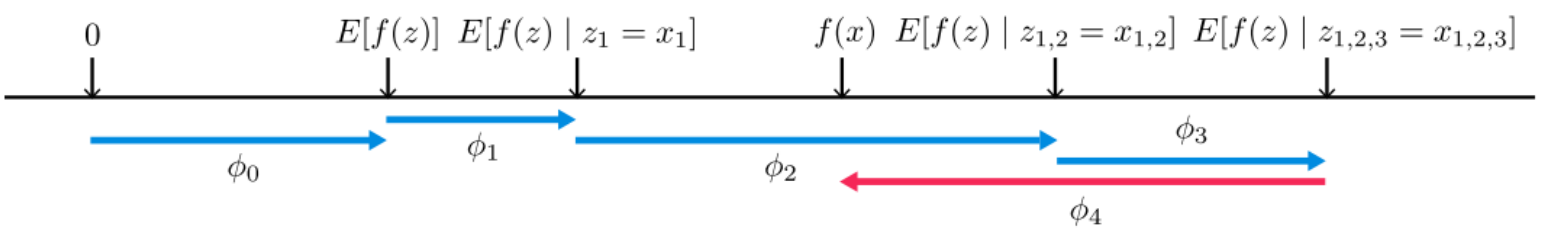
SHAP (SHapley Additive exPlanations)

Idée : Unifier toutes les approches précédentes grâce à la théorie des jeux

Framework

- la **shapley value** est une méthode pour attribuer les gains individuels aux joueurs en fonction de leur contribution au gain total .
- les joueurs sont **les valeurs de nos features** , le gain total pour une instance z est : $f(z) - E(f(z))$
- la shapley value est la contribution marginale moyenne d'une valeur de feature sur toutes les coalitions possibles :

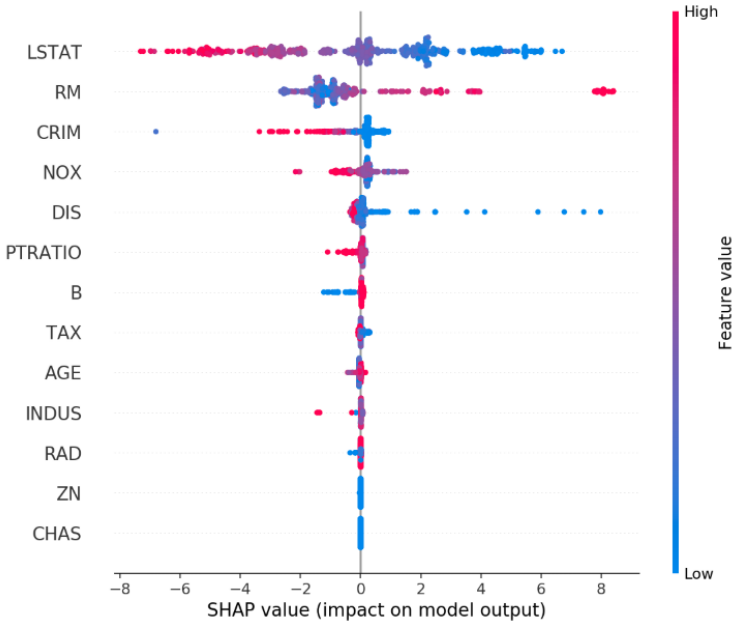
$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!\,(p-|S|-1)!}{p!} \left(val\left(S \cup \{x_j\}\right) - val(S)\right)$$

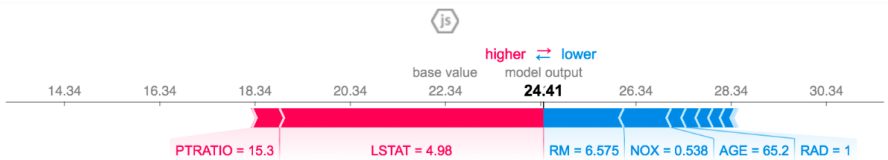


SHAP

Modèles engendrés

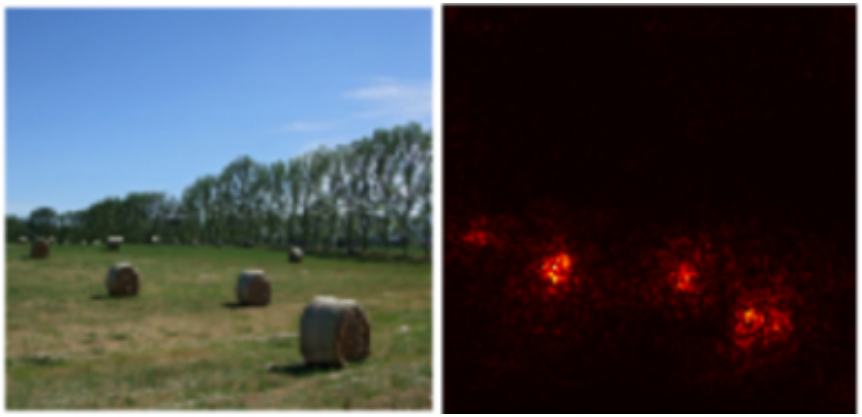
- **KernelSHAP (LIME+Shapley values)** : Lime utilise des heuristiques pour les kernels , SHAP montre que le meilleur unique kernel est : $\pi_x(z') = \frac{M-1}{\binom{M}{|z'|} * |z'| * (M-|z'|)}$
- **DeepSHAP (DeepLIFT+Shapley values)** : spécifique aux réseaux de neurones
- **TreeSHAP(Decision Tree+Shapley values)**: variante pour les modèles Tree-based



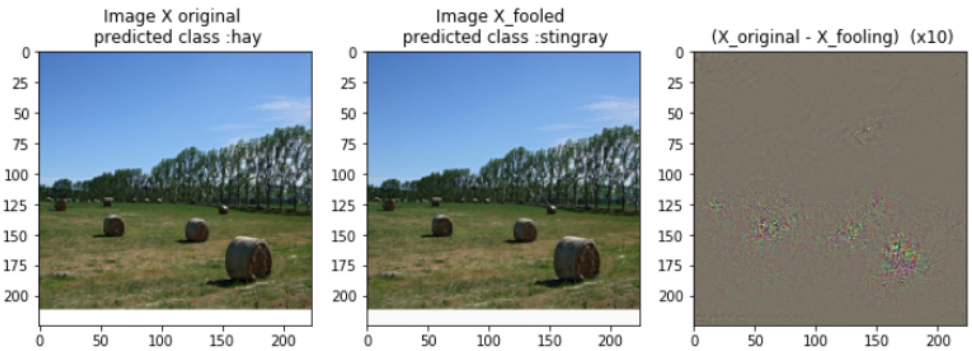


Example-based

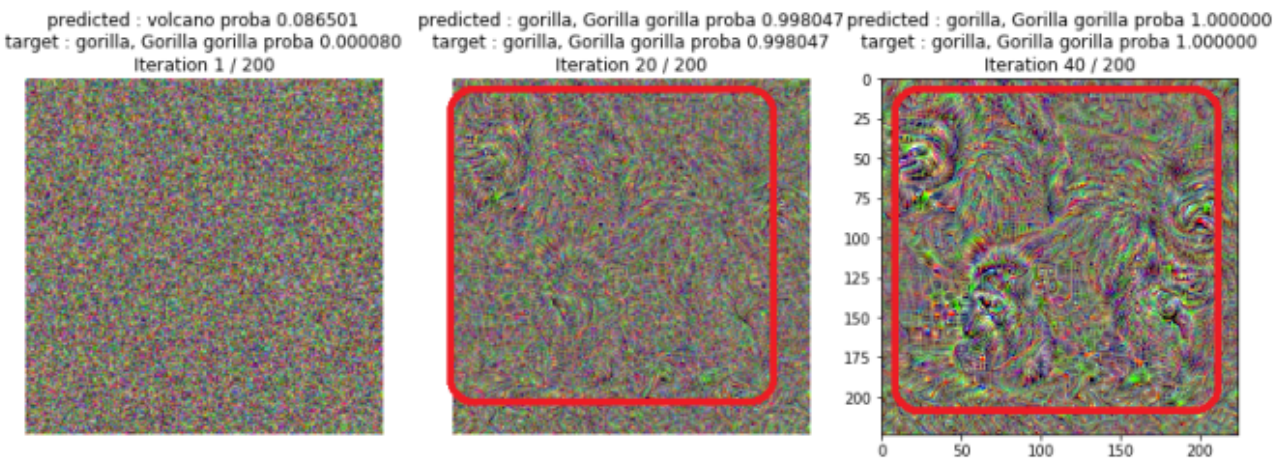
Saliency Maps (hay class)



Adversarial attacks(hay class)



Deep Visualization(Gorilla class)



Comparaison et limites

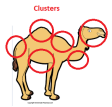
Models	LIME	LRP	DeepLift	Attacks	SHAP
Model-agnostic	✓				✓KernelSHAP
Model-specific		✓	✓	✓	✓DeepSHAP
local approximation	✓	✓	✓	✓	✓KernelSHAP ✓TreeSHAP
global approximation	✓SP-LIME				✓Feature Importance and others
gradient-based				✓	
backprop-based		✓	✓	✓	✓DeepSHAP
perturbation-based	✓				✓KernelSHAP

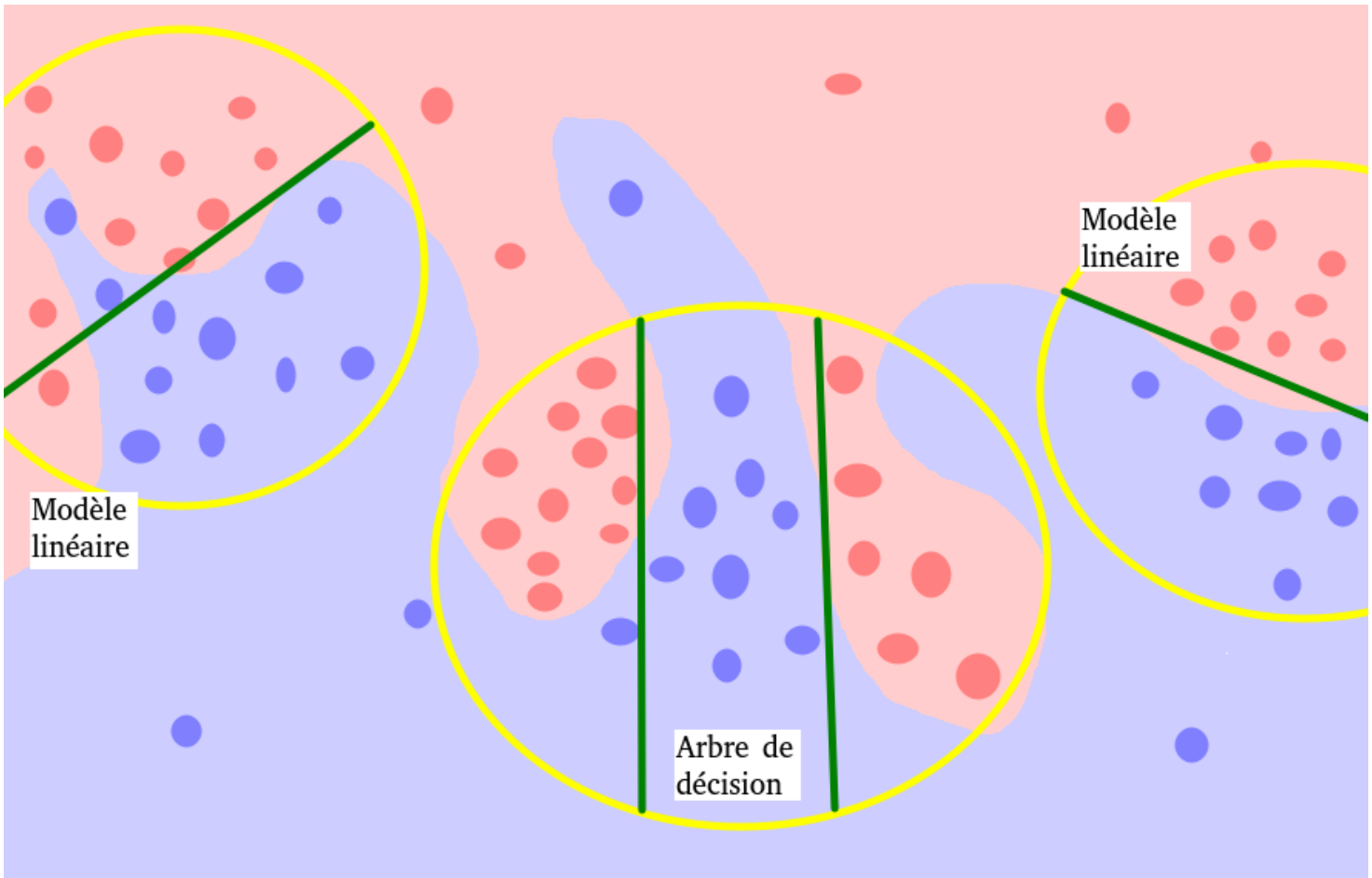
les principaux problèmes des précédentes approches ...

- Le problème du sampling et les exemples qui sont pas réalistes
- Le gradient et le problème de saturation, des discontinuités et des signaux négatifs
- Variation de l'interpretabilité : pour des petites perturbations sur un même point l'interpretabilité change drastiquement

Idée générale

CAMEL (Clustering bAsed Model-agnostic ExpLanations)

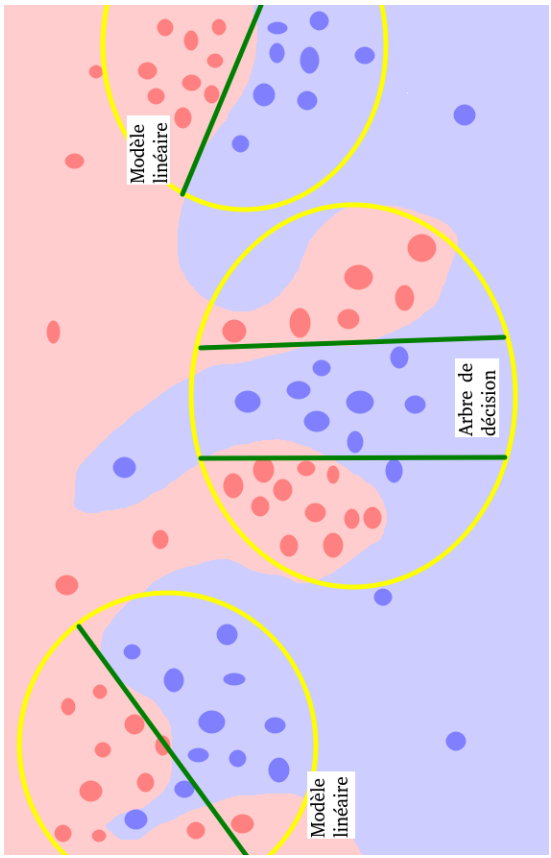




Idée

Ingrédients

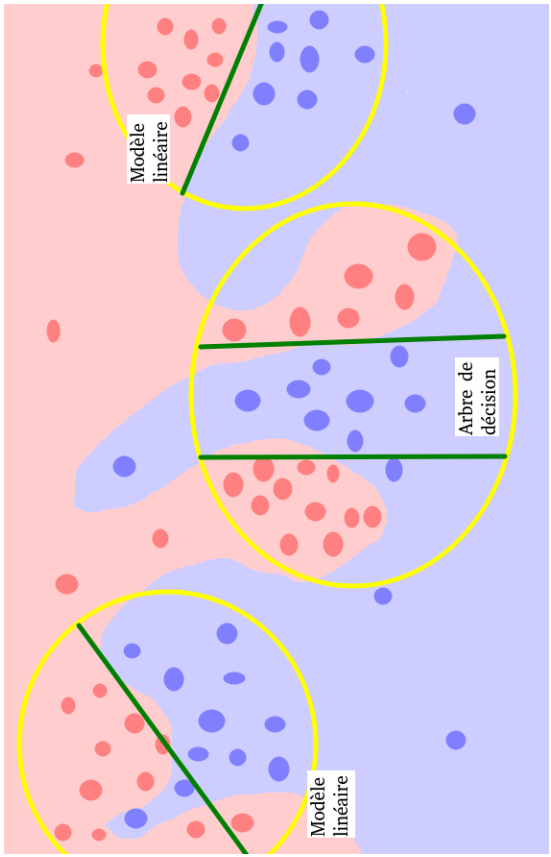
- cible : valeur prédite par la black box
- uniquement des point issus du dataset original
- Distance à la frontière (non représentée)
- Algorithme de clustering (en jaune)
- Algorithmes interprétables (en vert)



Architecture choisie

Tâche

- contexte de classification
- Cervical cancer (Risk Factors) Data Set [3]
- Foret aléatoire (n=50, profMax=5) entraînée
- dans un premier temps : pas de distance ie. on considère tous les points
- k-means (k=10)
- un arbre de décision par cluster (profMax=3, 5-fold cross validation)



Evaluation

Protocole

- expérimentation sur des humains
- volontaire / rémunération
- nombre d'exemples limités
- chaque exemple soumis à plusieurs personnes
- calcul du score
- comparaison avec LIME

Score

- pour chaque exemple : similarité cosinus
- on moyenne tous les exemple pour le modèle
- k-means non déterministe : plusieurs itérations du modèle

Questionnaire

- un exemple et la classification donnée par la black box
- sélectionner au plus 3 features pertinentes

Possibilité d'analyse

Paramètres à explorer

- $k \in [5, 10, 15]$: nombre de clusters
- $profMax \in [1, 2, 3]$: profondeur max pour les arbres
- $K - LASSO \in [5, 10, 15]$: pour les modèles linéaires (Regréssion L1)

Remarques

- 10 modèles à évaluer
- 10 exemples par modèles
- 10 évaluations par exemple par modèle
- soit 1000 questionnaires à remplir pour évaluer un modèle

Conclusion

Bilan

- **Interprétabilité** vs **Performance**
- **Notre approche** : combiner global et local, stabilité des interprétations, pas de sampling, pas de gradients ...
- **Coût de l'évaluation** : comme toutes les méthodes ...

Merci pour votre attention

Des questions?

Réferences I

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W.
On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation.
PLOS ONE 10, 7 (2015), 1–46.

Binder, A., Montavon, G., Lapuschkin, S., Müller, K.-R., and Samek, W.
Layer-Wise Relevance Propagation for Neural Networks with Local Renormalization Layers.
In *Artificial Neural Networks and Machine Learning – ICANN 2016* (Cham, 2016), A. E. Villa, P. Masulli, and A. J. Pons Rivero, Eds., Springer International Publishing, pp. 63–71.

Réferences II


Fernandes, K., Cardoso, J. S., and Fernandes, J.
Transfer learning with partial observability applied to cervical cancer screening.
In *Iberian conference on pattern recognition and image analysis* (2017), Springer, pp. 243–250.


Goodfellow, I. J., Shlens, J., and Szegedy, C.
Explaining and harnessing adversarial examples.
arXiv preprint arXiv:1412.6572 (2014).


Lundberg, S. M., and Lee, S.-I.
A unified approach to interpreting model predictions.
In *Advances in Neural Information Processing Systems* (2017), pp. 4765–4774.

Réferences III


Molnar, C.
Interpretable Machine Learning.
2019.
https://christophm.github.io/interpretable-ml-book/.


Montavon, G., Samek, W., and Müller, K.-R.
Methods for interpreting and understanding deep neural networks.
Digital Signal Processing 73 (2018), 1 – 15.


Shrikumar, A., Greenside, P., and Kundaje, A.
Learning important features through propagating activation differences.
In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 3145–3153.

Réferences IV


Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A.
Not just a black box: Learning important features through propagating activation differences.
ArXiv abs/1605.01713 (2016).


Simonyan, K., Vedaldi, A., and Zisserman, A.
Deep inside convolutional networks: Visualising image classification models and saliency maps.
arXiv preprint arXiv:1312.6034 (2013).


Tulio Ribeiro, M., Singh, S., and Guestrin, C.
" Why Should I Trust You?": Explaining the Predictions of Any Classifier.
arXiv preprint arXiv:1602.04938 (2016).

Réferences V

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H.
Understanding neural networks through deep visualization.
arXiv preprint arXiv:1506.06579 (2015).