Machine Learning Interpretability

Abdelraouf KESKES
Aurélia DOLO

January 21, 2020

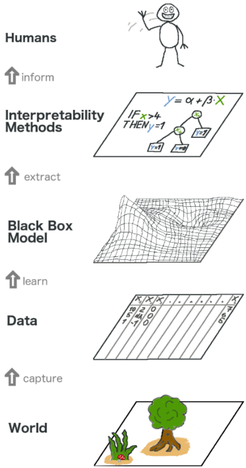
Outline

SCIENCES SORBONNE UNIVERSITÉ

Problem Setting

Problem

- Metrics such as Accuracy, F1-score,...:
  * they are not very informatives and reliable/trustful to take decisions
  * they don't answer the "How" and "Why" questions for the returned prediction
- **Question** : How to get more reliable predictions ? => **machine learning interpretability/explainability** ?



Fundamental definition

ML Explanation definition

- It must be human understandable, what ever his/her background is .
- It should give us an insightful justification for a given prediction
- It is often instance/example based
- the terms "Interpretability" and "Explainability" are used interchangeably in Machine learning

Objectifs

- Build low risk applications (sensitive domains such as health).
- Advanced debugging and therefore performances improvement for ML researchers and engineers.
- Build robust and reliable models.
- AI fairness.

Fundamental definitions

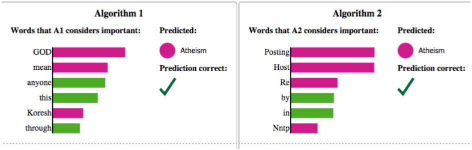
General notions

- There are some AI system that don't require an interpretability !
- We have 2 models (**"ML"** and **"EXplainability"**)
- The features are not necessary the same for both the ML model and the EX model, for instance in NLP we have words embeddings for **ML** and one hot encoding for **EX**

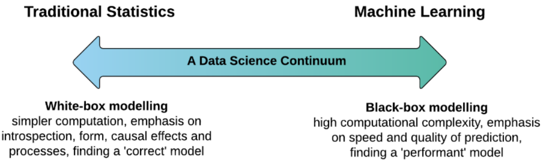
ML models are splitted into 2 categories

- **White box** :Models that are interpretable by essence, but they are less expressive and less performant
- **Black box** : Models that are more performant, more complex, and more expressive, **But** their decisions are impossible to understand, hence the **Explainability**

Fundamental definitions



Source : (LIME paper)



Source : (Applied AI)

Fundamental definitions

Explainability models are splitted into 2 categories

- **Model-Agnostic** : a model that works for all ML models considering it as a "black box"
- **Specific** : a model which explains a specific family of models by definition, for example : "Neural Networks", "Tree-Based", ...

The interpretability scope : Global vs Local

- **Local** : explain a certain prediction for a specific instance
- **Global** : explain the global behaviour of the ML model

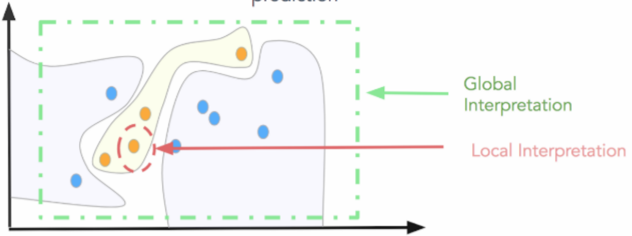
Fundamental definitions

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



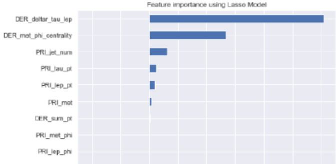
Source : (Datascience.com)

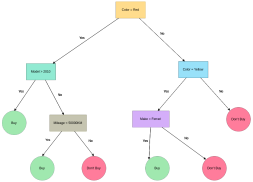
Interpretable models

White box models

the ML model and EX model are both the same model :

- Linear models (**)
- Decision Trees(**)
- Others(*): Decision Rules, Naive Bayes, KNN, ...



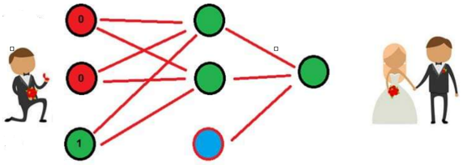


Complex Models

Very performant models but black box

We need a separate model to explain them

- Neural networks
- Bagging, Boosting, Random forests and others



XGBoost
RandomForests
ConvolutionalNets
RecurrentNeuralNetwork
FeedForwardNets
LightGBM
GANs

LIME

LIME (Local Interpretable Model-agnostic Explanations)

Idea : for a specific instance, we approximate the ML model locally

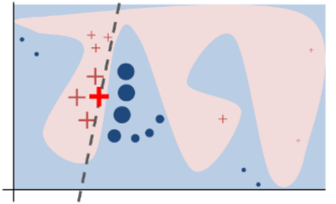
Framework

- an instance $x \in R^d$ and its EX representation $x' \in \{0,1\}^{d'}$
- The ML Model $f : R^d \mapsto R$ and the EX Model $g : \{0,1\}^d \mapsto R$
- Sampled instances $z_i \in R^d$,their representation z_i', and a proximity kernel $\pi_x(z)$
- Optimization problem : $\xi(x) = \arg\min_{g \in G} [\mathcal{L}(f, g, \pi_x) + \Omega(g)]$
- Custom MSE (weighted by a 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

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z_i' \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z_i', f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z_i' as features, $f(z)$ as target
return w



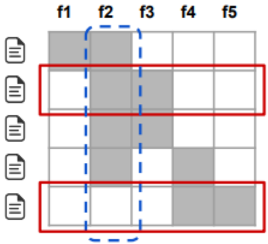
SP-LIME : Global extension of LIME

SP-LIME (Sub-modular Pick LIME)

Idea : locate representative instances and learn a LIME on each of them in order to approximate the model behavior **globally**

Algorithm 2 Submodular pick (SP) algorithm

Require: Instances X, Budget B
  for all $x_i \in X$ **do**
    $\mathcal{W}_i \leftarrow \textbf{explain}(x_i, x'_i)$ ▷ Using Algorithm 1
  end for
  for $j \in \{1 \ldots d'\}$ **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



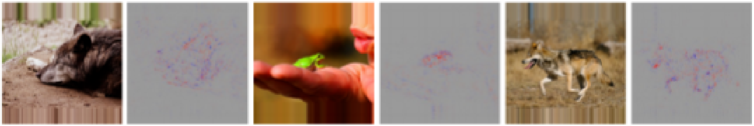
LRP

LRP (Layer-wise Relevance Propagation)

Idea : Backprop the outputs signals to the input layer in Neural Nets

Framework

- *aj* is the neuron *j* output which is defined as the non-linearity g : $a_j = g(\sum_i w_{ij} * a_i + b)$
- an instance x, an ML model f, a layer l, a dimension p, a relevance score R_p^l, as $f(x) \approx \sum_p R_p^{(1)}$ (features contribution is summed)
- features p with $R_p^{(1)} < 0$ contribute negatively to activate the output neuron and reversely ($R_p^{(1)} > 0$)



Source : (LRP with Local Renormalization Layers paper)

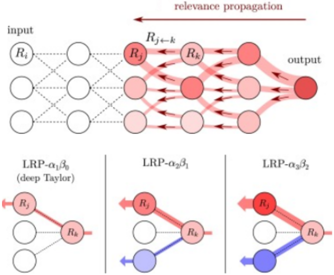
LRP

Details ...

- the relevance score for :
  - the **output** neurons is : $R^{(L)} = f(x)$
  - the **intermediate** neurons (back-prop) : $R_i^{(l)} = \sum_{j \in (l+1)} R_{i \leftarrow j}^{(l,l+1)}$
  - the **input** neurons ($l = 1$) where they are considered as the last intermediate neurons $R_i^{(1)}$
- All the variations of the EX model are based on $R_{i \leftarrow j}^{(l,l+1)}$ formula

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	$\times^\star$
flat [30]	$R_j = \sum_k \frac{1}{\sum_j 1} R_k$	lower layers	$\times$
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

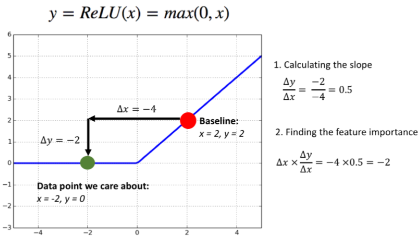
DeepLIFT (Deep Learning Important FeaTures)

Idea : Replace **gradients** by **differences** during the backprop

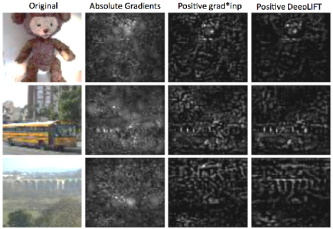
Détails

- Gradients raise 2 major problems : **Saturation, ReLU(virer les Negatifs, Discontinuité)**
- We are not interested in the gradient **(how y changes when x changes infinitesimally)**
- We are interested in the slope **(how y changes when x vary from its reference x_{ref})**
- $gradient = \frac{\partial y}{\partial x} \Rightarrow slope = \frac{y - y_{ref}}{x - x_{ref}} = \frac{\Delta y}{\Delta x}$
- We keep the "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 : $x_i \times \frac{\partial y}{\partial x_i} \Rightarrow (x_i - x_i^{ref}) \times \frac{\Delta y}{\Delta x_i}$
- **Problem** ? how to get the reference ?
  - inputs neurons : handcrafted by **domain experts** (Ex: MNIST => images initialized with 0)
  - intermediate and outputs neurons, we just need to **forward** the references inputs

DeepLIFT



Source : (Gabriel Tseng Medium Blogs)



Source : (DeepLIFT paper)

SHAP

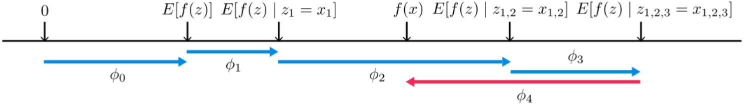
SHAP (SHapley Additive exPlanations)

Idea : Unify all the previous methods and many more under the game theory paradigm

Framework

- la **The shapley value** is a method of attributing individual rewards to players based on their contribution to the total reward .
- Players are **features values**,The total reward for an instance z is : $f(z) - E(f(z))$
- the shapley value is the average marginal contribution of a feature value for all possible coalitions.

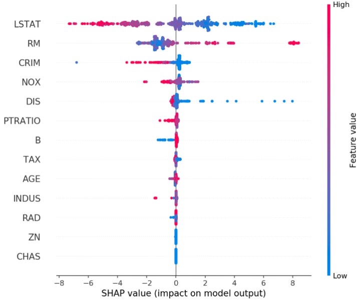
$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!\,(p-|S|-1)!}{p!} \left(val\left(S \cup \{x_j\}\right) - val(S)\right)$$

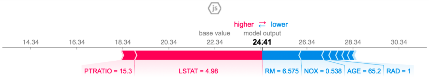


SHAP

SHAP EX Models

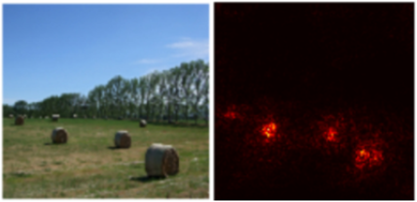
- **KernelSHAP (LIME+Shapley values)** : Lime uses heuristics to seek a kernels , SHAP demonstrates that the best kernel is unique and it is : $\pi_x(z') = \frac{M-1}{\binom{M}{|z'|} * |z'| * (M - |z'|)}$
- **DeepSHAP (DeepLIFT+Shapley values)** : Adapted for Neural Networks
- **TreeSHAP(Decision Tree+Shapley values)**: Adapted for Tree-based models



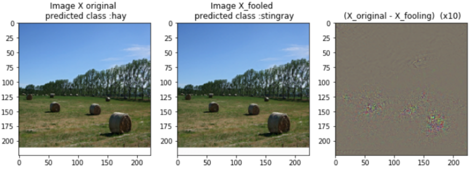


Example-based

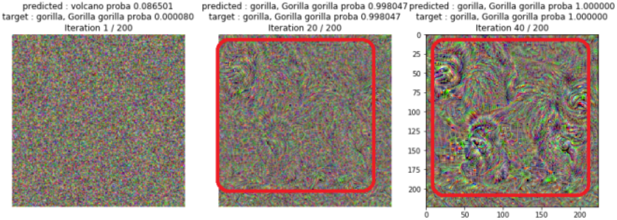
Saliency Maps (hay class)



Adversarial attacks(hay class)



Deep Visualization(Gorilla class)



Comparative summary

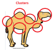
Models	LIME	LRP	DeepLift	Attacks	SHAP
Model-agnostic	✓				✓KernelSHAP
Model-specific		✓	✓	✓	✓DeepSHAP
local approximation	✓	✓	✓	✓	✓KernelSHAP ✓TreeSHAP
global approximation	✓SP-LIME				✓Feature Importance and others
gradient-based				✓	
backprop-based		✓	✓	✓	✓DeepSHAP
perturbation-based	✓				✓KernelSHAP

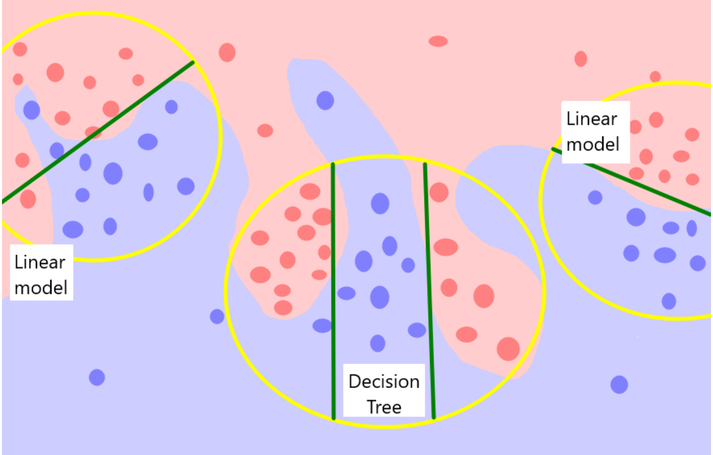
Common problems ...

- The sampling problem that could lead to unrealistic datapoints
- Gradient major problems : saturation, discontinuities, and negative signals backprop
- Explanations variability : for small changes on the same point, interpretability may change drastically

Our model in one slide !

CAMEL (Clustering bAsed Model-agnostic ExpLanations)

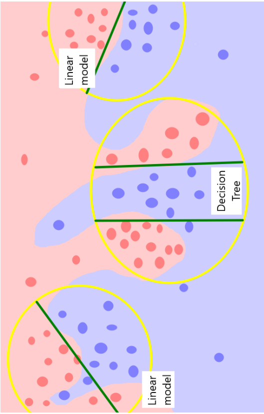




Architecture choisie

Idea and pseudo-algorithm

- No sampling we use only our datapoints, since the model learns a boundary from existing points !
- We apply a clustering algorithms based on the dataset distribution such as K-means or DBSCAN or ... (see clusters in Yellow)
- We learn an interpretable model locally in each cluster with a Kernel SHAP (see Green approximators)
- we could extend it to a global explanation



Evaluation

Protocole

- Dataset : Cervical Cancer (Risk Factors)
- As any other EX model cited previously we will need human volunteers to evaluate the quality of our explanations
- Each instance should be reviewed by different persons
- Then, we could build comparative tables and plots with other models

Hyper parameters to explore

- The clustering algorithm itself
- the number of clusters C
- $TreeDepth$ for Tree-based models and $K - Lasso$ for linear models
- others : for instance **weighted?** datapoints or not etc ...

Conclusion

Important points to keep in mind

- **Interpretability** vs **Performance**
- **Our approach** :

  combine both the global and local scope by essence
  No sampling
  No gradients back propagation
- **Cost** : similar to the other methods ...

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).