THALES
Building a future we can all trust

Comment examiner un modèle ?

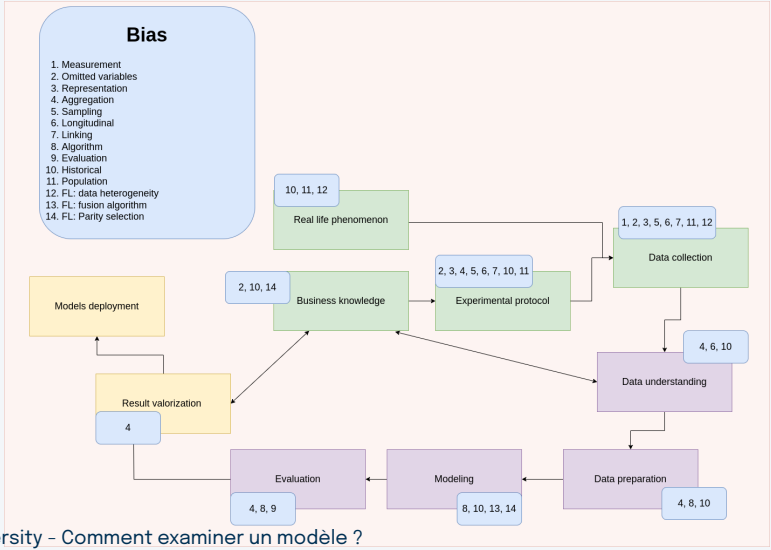
Comment examiner un modèle ?

Alice HELIOU and Vincent THOUVENOT
Laboratoire de Data Science de ThereSIS

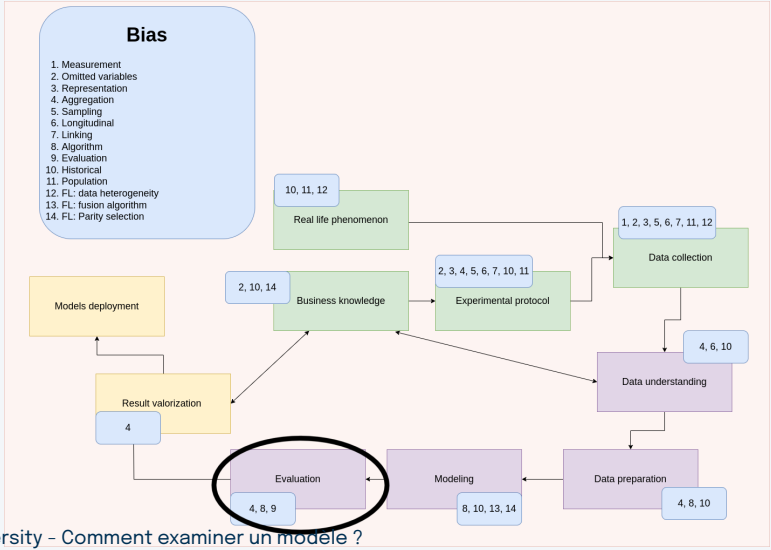
Content

Introduction

Position du cours



Position du cours



Rappel : Base rate métriques

Elles n'utilisent que la prédiction et les caractéristiques.

- Disparate impact

$$\frac{P(\hat{Y} = 1|Z = 0)}{P(\hat{Y} = 1|Z = 0)}$$

- Statistical parity (demographic parity) difference

$$P(\hat{Y} = 1|Z = 0) - P(\hat{Y} = 1|Z = 1)$$

avec $\hat{Y}$ la prédiction et Z le groupe (0 : non-privilégié, 1 : privilégié)
On peut s'en servir pour:

- comparer deux modèles
- comparer un modèle aux données (en utilisant alors le label au lieu de la prédiction)

Audit de modèle

Audit?

"Tous les modèles sont faux, mais certains sont utiles", Box et Draper, 1987

- Processus ou système d'outils pour décrire et valider les résultats d'un modèle appris à base d'IA
- Système à base d'IA souvent trop complexe pour être facilement validé
- Degré de complexité rend compliqué à anticiper les erreurs du modèle
- Audit nécessaire pour valider un modèle

Audit boite blanche ou boite noire

- **Boite blanche:** Implique que nous ayons accès au code source et aux poids du modèle, etc.
  - Par exemple, pour un modèle de régression linéaire, nous avons accès aux coefficients du modèle
  - Ses approches sont utilisées en interne, pour évaluer le modèle, le valider avant ou pendant son utilisation
- **Boite noire:** Nous pouvons seulement interrogé le modèle, mais pas avoir accès à ses informations internes
  - Par exemple, lors de l'utilisation d'un service de ML en ligne qui fournit des prédictions en fonction des entrée qu'on lui donne
  - Approches utilisées dans les audits externes où l'on ne connait pas l'ensemble du procédé d'apprentissage, et on n'a accès qu'aux prédictions du modèle.

Paramètres d'une audit

- **Parties prenantes:** Inclure au maximum les parties prenantes différentes, réprésentatif à la fois de la population et des minorités
- **Dépendances techniques:** Définir le système qui va être audité et sur lequel l'audit peut agir
- **Temps:**
  - Décalage coonnu entre la culture de la technologie, qui met l'accent sur l'innovation et la rapidité d'exécution et la culture de responsabilité de l'entreprise responsable de l'audit
  - Une audit peut être longue, notamment lorsqu'elle implique des interactions avec les parties prenantes
- **Objectifs:** Doivent être défnis à la fois au niveau conceptuel et concrêtement

Exemple d'audit - Cas de Facebook

- Audit sur les droits civiques à Facebook à partir de 2018, sous l'égide d'un avocat spécialis dans les droits civiques
- Déroulement:
  - Entretien avec plus de 70 organisations de défense de droit civiques
  - Problématique mise en avant:
    - Discriminations dans la publicité pour le logement, le crdit et l'emploi
    - Utilisation de Facebook pour diffuser des messages et organiser des événements suprématistes
  - Audit fournit un retour d'expérience sur les points sur lesquels Facebook doit mettre en place
    - Détection de fake news
    - Identification de contenus inappropriés
    - Détection de publication encourageant à ne pas voter

Google framework

- Proposé par des chercheur.se.s de Google en 2021
  - Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing, Raji, Smart, White, Mitchell, Gebru, Hutchinson, Smith-Loud, Theron, Barnes, 2021
- Part de la constatation que des outils individuels peut être utilisés à la fois pour l'audit et la recherche proactive de fairness
- Inclut des systèmes encourageant le reporting de métriques de performance d'un modèle et les attributs utilisés par celui-ci
- Audit sert à caractériser les aspects fondamentaux de l'ensemble de la chaine de traitement globale du système à base d'IA

Google framework

Scoping	Mapping	Artifact collection	Testing	Reflection	Post-audit
Define audit scope	Stakeholder buy-in	Audit checklist	Review documentation	Remedial plan	Go/no-go decision
Product requirements document (PRD)	Conduct interviews	Model cards	Adversarial testing	Design history file (ADHF)	Design mitigations
AI principles	Stakeholder map	Datasheets	Ethical risk analysis chart		Track implementation
Use case ethics review	Interview transcripts			Summary report	
Social impact assessment	Failure modes and effects analysis (FMEA)				

Google framework

- **Scoping:**
  - Fixation des objectifs, si possible avec des groupes variés
  - Définition de l'impact prévu et des possiblités de changer le système
- **Mapping:**
  - Compréhension du système/des moyens d'action/ des resources disponibles par les auditeur.rice.s
  - Identification des parties internes pouvant contribuer
- **Artifact collection:**
  - Documenter l'état du système tel qu'il a été rencontré au cours de l'audit
  - Centralisation de la documentation importante

Google framework

- **Testing:**
  - Tests techniques, mesures de fairness
  - Doivent avoir été spécifiés lors des étapes précédentes
  - Ne doivent pas être uniquement quantitatifs, mais également avec des retours humains
- **Reflection:**
  - Réservé à l'examen des tests et à leur mise en relation avec les objectifs et les champs d'application du projet
  - Développement de guides, de mesures à prendre et des évaluations des risques

Audit boîte noire

- Va dépendre de ce qu'on audit
- Quelques outils (parmi d'autres):
  - Examples contrefactuels
  - Construction d'un méta-modèle
  - Auditer un modèle pour les influences indirectes

What is a good counterfactual example?

- Individual with a forecast close to the target value

What is a good counterfactual example?

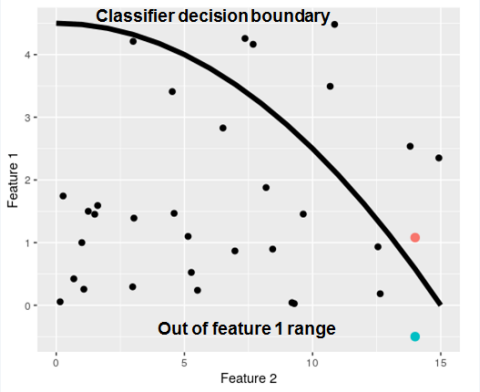
- Individual with a forecast close to the target value
- Individual close to the observation to be explained

What is a good counterfactual example?

- Individual with a forecast close to the target value
- Individual close to the observation to be explained
- As few features as possible should be modified

What is a good counterfactual example?

- Individual with a forecast close to the target value
- Individual close to the observation to be explained
- As few features as possible should be modified
- The individual must be realistic



Guideline for counterfactual examples quality evaluation I

- Computation time
- Counterfactual examples candidate can not reach the target prediction with some approaches
  - Success rate of counterfactual examples
- Proximity between the original instance and the counterfactual example
  - Combination of L2, L1 and L0 Norm between the two instances
  - Gower distance

$$D_{Gower}(\mathbf{x}, \mathbf{y}) = 1 - \frac{1}{p} \sum_{j=1}^{p} s_j(\mathbf{x}, \mathbf{y}),$$

where

- If j is a quantitative feature, than $s_j(\mathbf{x}, \mathbf{y}) = 1 - \frac{|x_j - y_j|}{R_j}$ where R_j is the range of feature k
- If j is qualitative, $s_j(\mathbf{x}, \mathbf{y}) = \mathbf{1}_{x_j = y_j}$.

Guideline for counterfactual examples quality evaluation II

- Feasibility of counterfactual examples
  - Number of time where a counterfactual example violates constraints
  - Evaluation of realism of counterfactual example with an anomaly detection algorithm
- Evaluate how much a feature change was necessary to change the model prediction
  - Successively flipping one value of the counterfactual example after another back to original instance
  - Compute when the prediction flipped from the target prediction to the original prediction
- Diversity of counterfactual examples

Guideline for counterfactual examples quality evaluation III

- Uncertainties of prediction in the neighborhoods around the counterfactual example
  - Consider some instances such that $H = \{\mathbf{x} \in X | f(\mathbf{x}) \leq \theta\}$ with $\theta \in [0, 1]$ and denote by $\tilde{H}$ the counterfactual instances corresponding to the set H:

$$yNN = 1 - \frac{1}{|\tilde{H}|k} \sum_{i \in \tilde{H}} \sum_{j \in kNN(\mathbf{c}_{\mathbf{x}_i})} |1_{f(\mathbf{c}_{\mathbf{x}_i}) > \theta} - 1_{f(\mathbf{x}_j) > \theta}|,$$

where kNN denotes the k nearest neighbors.

Three approaches families

- Independence-based approaches
  - Assume that the input feature of the model are independent
  - Use combinatorial solvers, evolutionary algorithms, gradient based optimization to minimize a loss with feasibility and diversity constraints
- Causality-based approaches
  - Use of Pearl's causal modeling
  - Assume the knowledge of the system of causal equations or the causal graph
- Dependence-based
  - Use some generative models
  - Allow to take into account dependence between features

Independence-based approach: A first method

$$CF(\mathbf{x}) = \arg\min_{\mathbf{z}} d(\mathbf{x}, \mathbf{z}) + \lambda y_{loss}(f(\mathbf{z}), y'),$$

where

- y' is the target prediction for the counterfactual examples
- y_{loss} could be the Hinge Loss $y_{loss}(f(\mathbf{z}), y') = \max(0, 1 - t \times logit(f(\mathbf{z}))$, where $t = 2y' - 1$ and $logit$: $x \to \log\left(\frac{x}{1-x}\right)$
- $d(\mathbf{x}, \mathbf{z}) = \sum_{j \in F_{cat}} \mathbb{1}_{x^j \neq z^j} + \sum_{j \in F_{con}} \frac{|x^j - z^j|}{median_{i \in \{1,\dots,n\}} |x_i^j - median_{i \in \{1,\dots,n\}}(x_i^j)|}$

Wachter et al. Counterfactual Explanations without Opening the Black Box, 2018: Automated Decisions and the GDPR.

Independence-based approach: Enforce diversity of counterfactual examples

$$\mathbf{CF}(\mathbf{x}) = \arg\min_{\mathbf{z_1},\ldots,\mathbf{z_k}} \frac{1}{k}\sum_{i=1}^{k} d(\mathbf{x},\mathbf{z_i}) + \frac{\lambda_1}{k}\sum_{i=1}^{k} y_{loss}(f(\mathbf{z_i}),y') - \lambda_2 \mathit{diversity}(\mathbf{z_1},\ldots,\mathbf{z_k}),$$

where

- k is the number of counterfactual examples targeted
- *diversity* is a diversity metric. For instance

$$\mathit{diversity}(\mathbf{z_1},\ldots,\mathbf{z_k}) = \mathit{det}(\mathbf{K}),$$

  where $\mathbf{K_{i,j}} = \frac{1}{1+d(\mathbf{z_i},\mathbf{z_j})}$.
- λ_1 and λ_2 are some hyperparameters

Mothilal et al. Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations.

Independence-based approach: Solve a multi-criterion problem

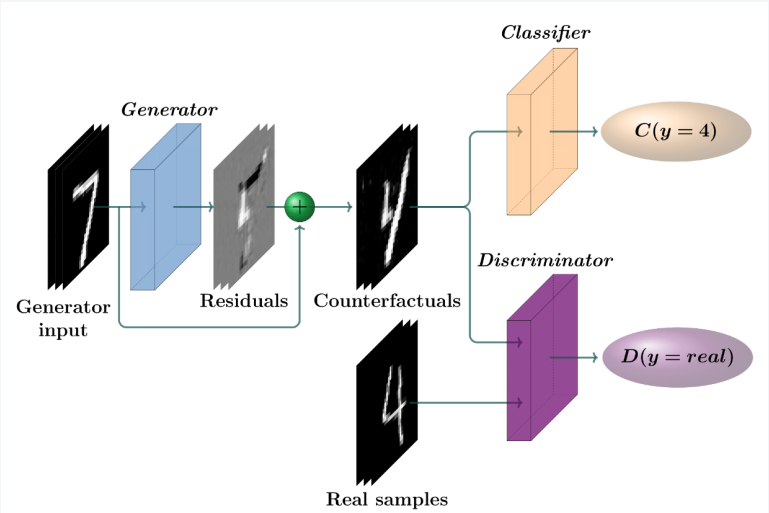
$$CF(\mathbf{x}) = \arg\min_{\mathbf{z}} \left(o_1(f(\mathbf{z}), Y'), o_2(\mathbf{x}, \mathbf{z}), o_3(\mathbf{x}, \mathbf{z}), o_4(\mathbf{z}, X) \right),$$

where

- Y' is the set of the targeted predictions
- $o_1(f(\mathbf{z}), Y') = \inf_{y' \in Y'} |f(\mathbf{z}) - y'| \mathbb{1}_{f(\mathbf{z}) \notin Y'}$
- $o_2(\mathbf{z}, \mathbf{x}) = \frac{1}{p} \sum_{j=1}^{p} \delta_G(z_j, x_j)$, where $\delta_G(z_j, x_j) = \frac{1}{R_j}|z_j - x_j|$ if the feature j is numerical, with R_j is the range value of the feature. If j is categorical, then $\delta_G(z_j, x_j) = \mathbb{1}_{x_j \neq z_j}$
- $o_3(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{p} \mathbb{1}_{x_j \neq z_j}$
- $o_4(\mathbf{z}, X) = \frac{1}{p} \sum_{i=1}^{q} w_i \sum_{j=1}^{p} \delta_G(z_j, x_j)$, where $\sum_{i=1}^{q} w_i = 1$

Dandl, S.,et al. Multi-Objective Counterfactual Explanations, 2020

Dependence-based approach: Use of a GAN



Dependence-based approach: Use of a GAN

$$\min_G \max_D V_{CounteRGAN}(D, G) = V_{S-RGAN}(D, G) + V_{CF}(G, f, y') + Reg(G(\mathbf{x})),$$

where

- $V_{S-RGAN}(D, G) = E_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left(\log(D(\mathbf{x}))\right) + E_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left(\log(1 - D(\mathbf{x} + G(\mathbf{x})))\right)$
- $V_{CF}(G, f, y') = E_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left(1 - 1_{\mathbb{1}_{f(\mathbf{x}+G(\mathbf{x}))>threshold} = y'}\right)$
- $Reg(G(\mathbf{x}))$: combination of l_1 and l_2 norms, help to control the sparsity and the amplitude of the perturbation of the original instance

Nemirovsky, et al., CounteRGAN: Generating Realistic Counterfactuals with Residual Generative Adversarial Nets. 2021

Dependence-based approach: Use of a VAE

Loss function:

$L_{VAE}(\mathbf{x}, \mathbf{c_x}) = E_{Q(\mathbf{z}|\mathbf{x},y')}\left(d(\mathbf{x}, \mathbf{c_x}) + \lambda y_{loss,\beta}(f(\mathbf{c_x}), y') + KL(Q(\mathbf{z}|\mathbf{x}, y')||P(\mathbf{z}|\mathbf{x}, y'))\right)$, where $Q(\mathbf{z}|\mathbf{x}, y')$: latent space encoder

Possibility to add a feedback from the user:

$$\min \sum_{i=1}^{q} L_{VAE}(\mathbf{x}, \mathbf{c_x}) + \lambda_o ||o_i - sim(\mathbf{x_i}, \mathbf{c_{xi}})||_2^2,$$

where

- $(\mathbf{x_i}, \mathbf{c_{xi}}, o_i)_{i \in \{1,\dots,q\}}$: $o_i \in \{0, 1\}$ user feedback
- *sim*: similarity between the original instance and the counterfactual example and λ_o a hyperparemeter

Mahajan, et al. Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers. 2020

Dependence-based approach: Use of RL

- Train a generative model that works for all the instance
- Four advantages
  - Model-agnostic
  - Generating target-conditional counterfactual instances
  - Flexible feature range constraints for numerical and categorical attributes
  - Easily extended to other data modalities such as images
- Introduction of a conditioning vector to take into account univary and bivary constraints
- Encode an instance in a latent space, trains two separate network: an actor and a critic, and finally decode the counterfactual example in the original space

Samoilescu, et al., Model-agnostic and Scalable Counterfactual Explanations via Reinforcement Learning, 2021

Dependence-based approach: Use of RL

Load a pre-trained encoder *enc*. and decoder *dec*. Randomly initialize an actor $\mu(.;\theta_\mu)$ and the critic $Q(.;\theta_Q)$. Intialize a replay buffer D_B. Define a reward function *reward* and a post-processing function *pp*. λ_S and λ_C are some loss hyperparameters.

Repeat:

- Sample a batch of input $\mathbf{x}$;
- Construct random target $\mathbf{y}'$ and conditioning vector $\mathbf{c}$;
- Compute $y = f(\mathbf{x})$, $\mathbf{z} = enc(\mathbf{x})$ and $\mathbf{z_{CF}} = \mu(\mathbf{z}, y, y', \mathbf{c}; \theta_\mu)$;
- Select $\mathbf{z}_{\tilde{\mathbf{CF}}} = clip(\mathbf{z_{CF}} + \varepsilon, -1, 1), \varepsilon \sim N(0, 0.1)$.
- Decode $\tilde{\mathbf{c_x}} = pp(dec(\mathbf{z}_{\tilde{\mathbf{CF}}}), \mathbf{c})$;
- Observe $R = reward(f(\mathbf{c_x}), y')$
- Add $(\mathbf{x}, \mathbf{z}, y', y, \mathbf{c}, \mathbf{z}_{\tilde{\mathbf{CF}}}, R)$ in D_B;
- If step of update the actor and the critic, repeat for many updates:

Dependence-based approach: Use of RL

- If step of update the actor and the critic, repeat for many updates:
  - Sample uniformly a batch of B experiences: $\mathbb{B} = \{(\mathbf{x}^b, \mathbf{z_b}, y'_b, y_b, \mathbf{c_b}, \mathbf{z}^{\tilde{\mathbf{b}}}_{\mathbf{CF}}, R_b)\}$ from D;
  - Update the critic by one step gradient decent using $\nabla_{\theta_Q} \frac{1}{|\mathbb{B}|} \sum_{\mathbb{B}} \left(Q(\mathbf{z_b}, y'_b, y_b, \mathbf{c_b}, \mathbf{z}^{\tilde{\mathbf{b}}}_{\mathbf{CF}}) - R \right)^2$;
  - Compute for each b
    - $\mathbf{z}^{\mathbf{b}}_{\mathbf{CF}} = \mu(\mathbf{z_b}, y'_b, y_b, \mathbf{c_b}; \theta_\mu)$;
    - $\mathbf{c_b} = dec(\mathbf{z}^{\mathbf{b}}_{\mathbf{CF}})$;
    - $L_{max} = -\frac{1}{|\mathbb{B}|} \sum_{\mathbb{B}} Q(\mathbf{z_b}, y'_b, y_b, \mathbf{c_b}, \mathbf{z}^{\tilde{\mathbf{b}}}_{\mathbf{CF}})$;
    - $L_{sparsity} = \frac{1}{|\mathbb{B}|} \sum_{\mathbb{B}} \left(L_1(\mathbf{x}^{\mathbf{b}}, \mathbf{C_b}) + L_0(\mathbf{x}^{\mathbf{b}}, \mathbf{C_b}) \right)$;
    - $L_{consist} \frac{1}{|\mathbb{B}|} \sum_{\mathbb{B}} \left(enc(pp(\mathbf{C_b}, \mathbf{c_b})) - \mathbf{z}^{\mathbf{b}}_{\mathbf{CF}} \right)^2$;
  - Update actor by one-step gradient descent using: $\nabla_{\theta_\mu} (L_{max} + L_{sparsity} + L_{consist})$

Output: the train actor network μ used for counterfactual generation

Open Source and Thales tools

Module	Type	License	Language	Comments
DiCE	Open Source	MIT	Python	Mainly independence-based approaches, with one approach using Variational Auto-Encoder
Alibi	Open Source	Apache 2.0.	Python	RL, Prototype and independence-based approach. Include others approaches of xAI
CARLA	Open Source	MIT	Python	Evaluation of counterfactual examples. Use existing implementation of various independence and dependence based approaches
growingsphere	Open Source	MIT	Python	Growing sphere approaches
counterfactual	Open Source	No license	R	Independence based approach based on multi-objective opitmization
xAI TRT Canada	Thales	InnerSource	Python	Independence-based method (See Wachter). Include others approaches of xAI. Provide an API and an application
explainer	Thales	No license	Python	Independence based methods

Feedback on DiCE

- Proposed by researchers from Microsoft
- Content
  - Five approaches to find counterfactual examples
    - Model-agnostic method: random sampling, KD-Tree and genetic algorithms
    - Dedicated to differentiable model: explicit loss-based method (TensorFlow 1 and 2, PyTorch), VAE (only for PyTorch model)
  - Possibility to extract local and global features importance
  - Can be used for classification or regression tasks
- Roadmap
  - Use of DiCE for debugging ML models
  - Constructed English phrases and other ways to output the counterfactual examples
  - Evaluating feature attribution methods on necessity and sufficiency metrics using counterfactuals
  - Bayesian optimization and other algorithms for generating counterfactual explanations
  - Better feasibility constraints for counterfactual generation

Feedback on DiCE

```
import dice_ml
# Provide data structure
d = dice_ml.Data(dataframe=train_dataset, continuous_features=['age', 'hours_per_week'],
    outcome_name='income')
# Using sklearn backend
m = dice_ml.Model(model=model, backend="sklearn")
# Using method=random for generating CFs
exp1 = dice_ml.Dice(d, m, method="random")
e1 = exp1.generate_counterfactuals(x_test, total_CFs=1
                                   , desired_class="opposite")
```

Alibi

- Provided by researchers from Seldon Technologies Ltd
- Provide various methods for local and global explanations
  - Independence-based, Prototype and RL counterfactual examples approaches, ALE, Anchor, SHAP, etc.
- Roadmap
  - Integrate TensorFlow 2 for counterfactual examples backend
  - Enforce the control that the user can have on the constraints on features space change

Alibi

```
predictor = lambda x: clf.predict_proba(preprocessor.transform(x))

explainer = CounterfactualRLTabular(predictor=predictor,
                                    encoder=heae.encoder,
                                    decoder=heae.decoder,
                                    latent_dim=LATENT_DIM,
                                    encoder_preprocessor=heae_preprocessor,
                                    decoder_inv_preprocessor=heae_inv_preprocessor,
                                    coeff_sparsity=COEFF_SPARSITY,
                                    coeff_consistency=COEFF_CONSISTENCY,
                                    category_map=adult.category_map,
                                    feature_names=adult.feature_names,
                                    ranges=ranges,
                                    immutable_features=immutable_features,
                                    train_steps=TRAIN_STEPS,
                                    batch_size=BATCH_SIZE,
                                    backend="tensorflow")

explainer = explainer.fit(X=X_train)

explanation = explainer.explain(X, Y_t, C)
```

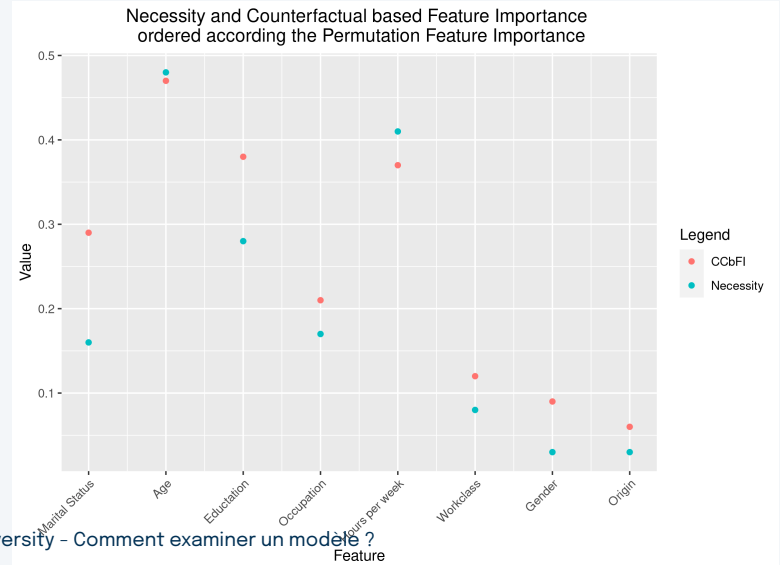
Counterfactual example as ML interpretability tool

- Adult Dataset: Predict whether income exceeds $50K/yr based on census data
- Training of a Random Forest and use of DiCE to generate counterfactual examples
- Local explanation: for one observation, generate one or several counterfactual examples
- Global explanation
  - Generate counterfactual examples for several observations and extract two global features importance criteria
    - Necessity: Proportion of sucess for finding a counterfactual example when only the feature is allowed to change during the generation of the counterfactual example
    - Counterfactual examples based feature importance: Proportion of counterfactual examples where the given feature changes
  - Python implementation based on DiCE
  - Comparaison of results with classical Permutation Feature Importance

Counterfactual example as ML interpretability tool

Type of observation	age	workclass	education	marital status	occupation	race	gender	hours per week	pred
Original instance	29	Private	HS-grad	Married	Blue-Collar	White	Female	38	0
Counterfactual example	29	Private	**Assoc**	Married	**White-Collar**	White	Female	38	**1**

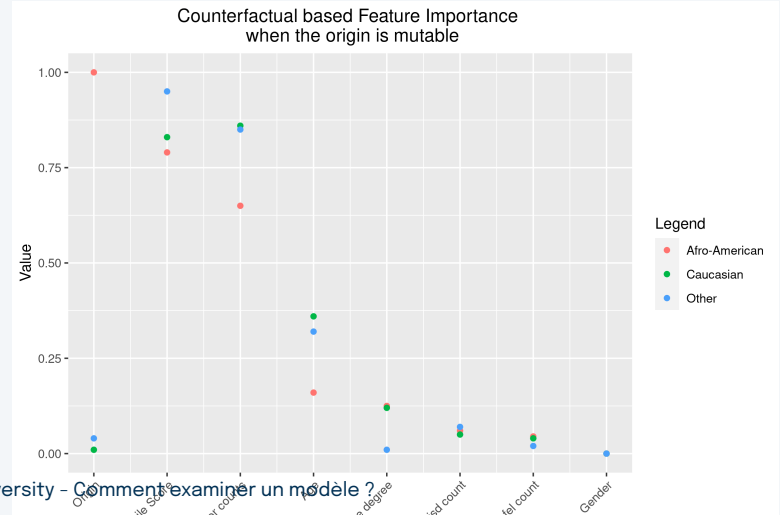
Counterfactual example as ML interpretability tool



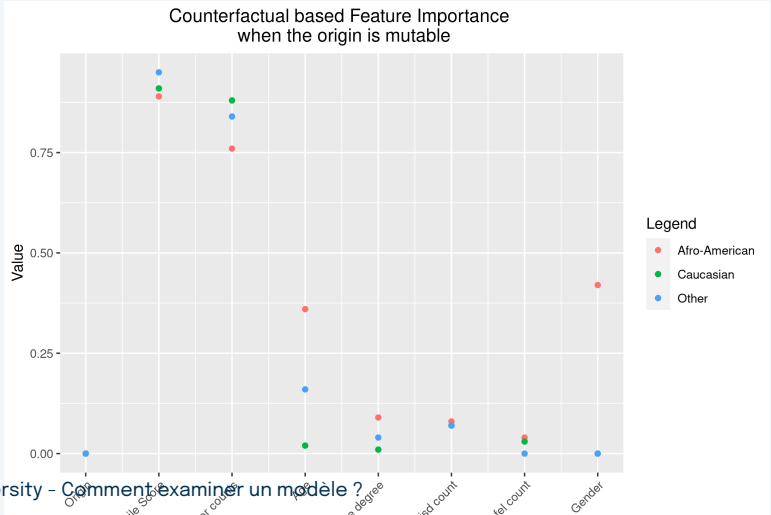
Counterfactual example as ML Fairness evaluation tool

- COMPAS dataset: Predict recidivism based on socio-demographic information and COMPAS score
- Training of a Logistic Regression and use of Alibi to generate counterfactual examples
- Fairness inspection based on Counterfactual example by focusing on individual predicted as recidivist on a testing set
  - Counterfactual example based feature importance
  - Average distance between the original instance and the counterfactual examples
  - Comparison of distance when the origin is allowed to change or is immutable

Counterfactual example as ML Fairness evaluation tool



Counterfactual example as ML Fairness evaluation tool



Counterfactual example as ML Fairness evaluation tool

Scenario	Population	Average distance
Immutable	Afro-American	6.47 (6.0)
Immutable	Caucasian	5.35 (4.8)
Immutable	Other	5.96 (4.4)
Mutable	Afro-American	5.95 (5.8)
Mutable	Caucasian	5.22 (4.8)
Mutable	Other	5.56 (4.3)

Construction d'un méta modèle

- Apprendre un modèle apprenant à prédire les sorties du modèle à auditer
- Plusieurs questions:
  - Choix d'un modèle facilement interprétable ou compliqué (mais potentiellement plus précis) ?
    - Attention, même si les prédictions du méta-modèle sont similaires à celles du modèle à auditer, il peut "raisonner" de manière complétement différente
  - Apprendre un modèle qui en fonction des sorties du modèle à auditer prédis les attributs sensibles ?

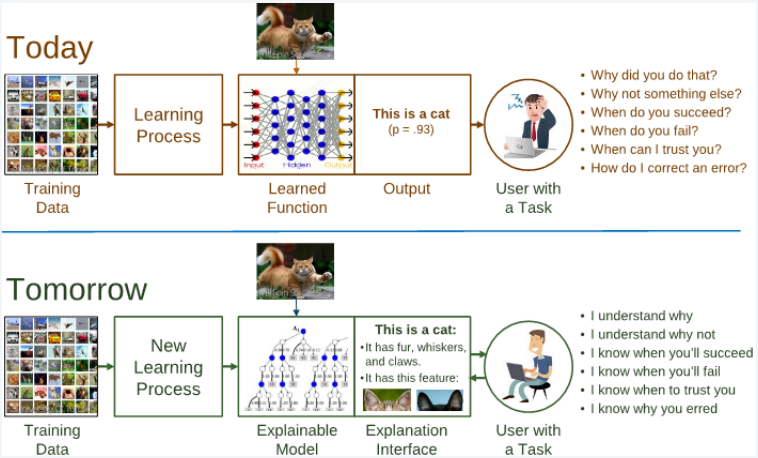
Auditer les modèles pour les influences indirectes

Auditing Black box models for indirect influence, Adler et al.

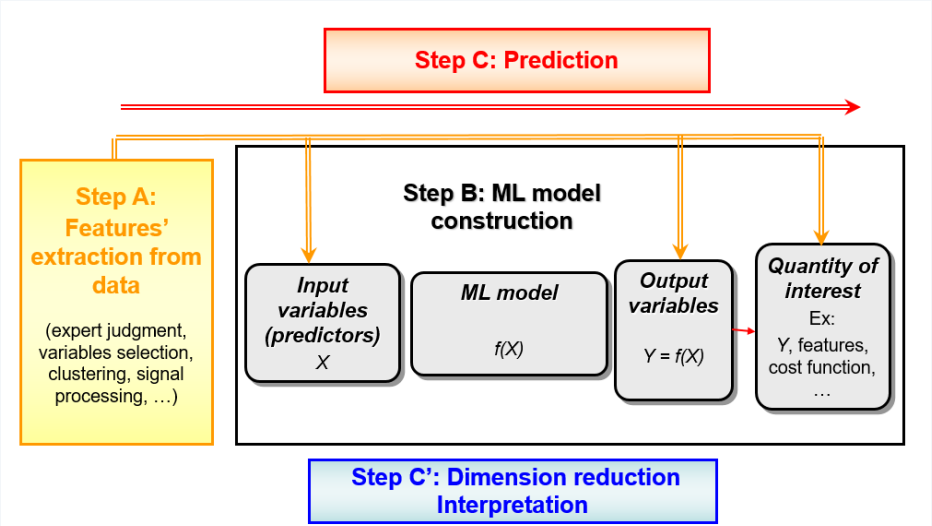
- Constat : Lorsque l'attribut sensible est obtenable à partir des autres caractéristiques, il risque d'avoir une influence indirecte sur le modèle apppris.
- Objectif : Evaluer la sensibilté du modèle (en boite noire) à l'influence indirecte de l'attribut sensible
- Méthodologie : Pour chaque caractéristique, on cherche à modifier minimimalement sa distribution. De sortes qu'au final l'attribut sensible ne puisse pas être déduit avec une accuracy supérieure à un certain seuil.
- La baisse de performance du modèle audité sur les données corrigées reflète l'importance des effets indirects de l'attribut sensible

Interpretable, explainable AI

Explicativité et interprétabilité dans le Machine Learning - DARPA's point of view



Context


Step C: Prediction
Step A: Features' extraction from data
(expert judgment, variables selection, clustering, signal processing, …)
Step B: ML model construction
Input variables (predictors)
X
ML model
f(X)
Output variables
Y = f(X)
Quantity of interest
Ex:
Y, features, cost function, …
Step C': Dimension reduction Interpretation

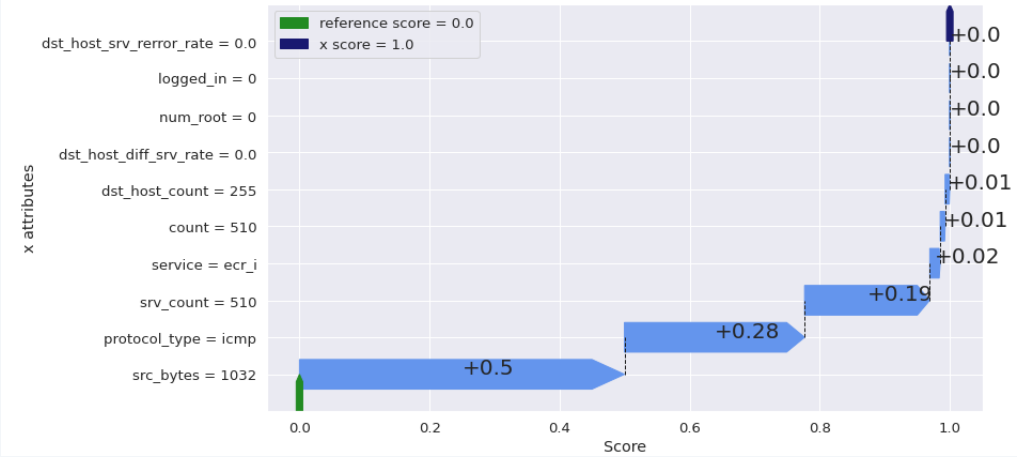
Interpretibility level

- Global behaviour
  - In Machine Learning, how find the features the more important on the model outputs ?
  - In Sensitivity Analysis, how deal with dependent features ?
- Local behaviour
  - In Machine Learning, how the features impact the model output for one given observation ?

Post-hoc interpretability vs intrinsic interpretability

- Post-hoc interpretability
  - A fitted model is interpreted using external tools
  - Global interpretation tools: Partial dependence plot, permutation features importance, etc.
  - Local interpretation tools: LIME, SHAP, ShapKit, counterfactual example, saliency map, etc.
- Intrinsic interpretability
  - Simple model: linear model, decision tree, etc.
  - Generalized Additive Model, EBM, GAMI-Net, xNN
  - Rule based approaches: Skope-Rule, Rule-fit, SIRUS, etc.

Features attribution: overview

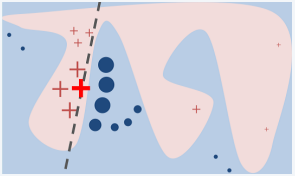


Features attribution : Local Interpretable Model-Agnostic Explanation (LIME)

Objective: Obtain an explanation about the prediction for one instance
Assumption:

- Model broadly complex
- Locally, can be approximated by a more understandable models (linear or logistic regression, tree, etc)

Features attribution : Local Interpretable Model-Agnostic Explanation (LIME)



1. Choose your instance of interest for which you want to have an explanation of its black box prediction
2. Perturb your dataset and get the black box predictions for these new points
3. Weight the new samples by their proximity to the instance of interest
4. Fit a weighted, interpretable model on the dataset with the variations
5. Explain prediction by interpreting the local model

$$\arg\min_{g} \sum_{z,z' \in Z \times Z'} \pi_x(z)(f(z) - g(z'))^2 + \Omega(g)$$

Features attributions for interpretation for machine learning predictions

Attribute importance for individual machine learning predictions.

How rank the features according the influence of their **attribute** on **model prediction** ?

Context of the study:

- **Local**: interpretation at instance level
- **Agnostic**: black box model
- **Contrastive**: comparison with reference(s)
- **Tabular data** with meaningful features

Features attribution: Shapley Value for attribute importance

Lloyd S Shapley. A value for n-person games. In Contributions to the Theory of Games. 1953.

- $\mathcal{M}$ a set of players of dimension d;
- $v : P(\mathcal{M}) \to R_v$ such as $v(\emptyset) = 0$. The range R_v can be $\Re$ or a subset of $\Re$. We shortly describe R_v in the case of Machine Learning next subsection. $P(\mathcal{M})$ is a family of sets over $\mathcal{M}$;
- If $S \subset \mathcal{M}$, $v(S)$ is the amount of wealth produced by coalition S when they cooperate.

The Shapley Value of a player j is a fair share of the global wealth $v(\mathcal{M})$ produced by all players together:

$$\phi_j(\mathcal{M}, v) = \sum_{S \subset \mathcal{M} \setminus \{j\}} \frac{(d - |S| - 1)!|S|!}{d!} \left(v(S \cup \{j\}) - v(S)\right),$$

with $|S| = \text{cardinal}(S)$, i.e. the number of players in coalition S. The Shapley Values are the only indices which respect the four following properties :

- Additivity: $\phi_j(\mathcal{M}, v + w) = \phi_j(\mathcal{M}, v) + \phi_j(\mathcal{M}, w)$ for all j, with $v : P(\mathcal{M}) \to R_v$ and $w : P(\mathcal{M}) \to R_v$;
- Null player: if $v(S \cup \{j\}) = v(S)$ for all $S \subset \mathcal{M} \setminus \{j\}$ then $\phi_j(\mathcal{M}, v) = 0$;
- Symmetry: $\phi_{\pi j}(\pi\mathcal{M}, \pi v) = \phi_j(\mathcal{M}, v)$ for every permutation π on $\mathcal{M}$;
- Efficiency: $\sum_{j \in \mathcal{M}} \phi_j(\mathcal{M}, v) = v(\mathcal{M})$.

Features attribution: Shapley Value for attribute importance

Inputs

- A trained model f (black box)
- An individual $\mathbf{x}^*$

Objective

Get importance $\phi_i \in \mathbb{R}$ of each attribute x_i^* of $\mathbf{x}^*$ such that:

$$\sum_{i=1}^{d} \phi_i = f(\mathbf{x}^*) - \phi_0$$

where ϕ_0 is a base value which can be the prediction of a chosen reference $\mathbf{r}$ ($\phi_0 = f(\mathbf{r})$).

Implemented in **ShapKit**, a Thales' Open Source Python module, and famous with **SHAP**.

SHAP, Shapkit, LIME, desméthodes intéressantes mais à utiliser avec précaution

- Basés sur des approximations avec une théorie fragile
- Rien n'assure la fidélité des explications
- D'ailleurs, les explications obtenues par LIME et SHAP peuvent être différentes, voire opposées…

Global surrogate model

1. Calculate the predictions for a dataset by the model to be explained
2. Learn an interpretable model (linear, tree, additive, etc.) using previous predictions as variables to explain
3. Evaluate the performance of the model
4. Interpret the results

Features summary

- Features importance: measure features impact on errors
  - Breiman, Random Forest, 2001
  - Fisher, Rudin, and Dominici, Model class reliance: Variables Importance Measure for any Machine Learning model class, from Rashomon perspective, 2018
- Partial Dependence plot: Marginal effect of one feature on prediction
  - Friedman, Greedy function approximation: a gradient boosting machine, 2000
- Accumulated Local Effects plot: Influence of one feature on prediction
  - Apley, Visualizing the effects of predictor variables in black box supervised learning models, 2016
- Implemented in **scikit-learn**

Features summary