Explainable AI- Counterfactual Explanations

Lecture at

"48th Winter Conference in Statistics"

Hemavan, March 12th, 2024

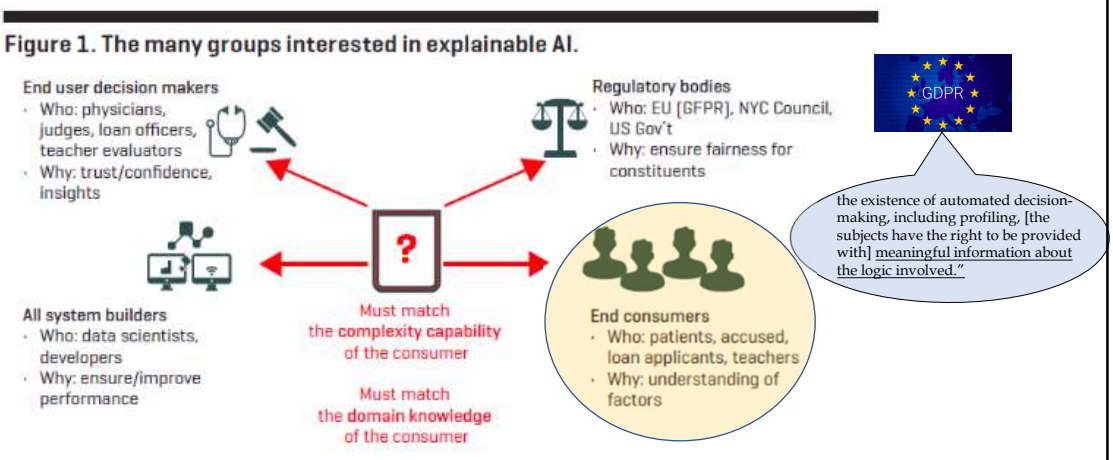


Figure 1. The many groups interested in explainable AI (from Hind, 2019)

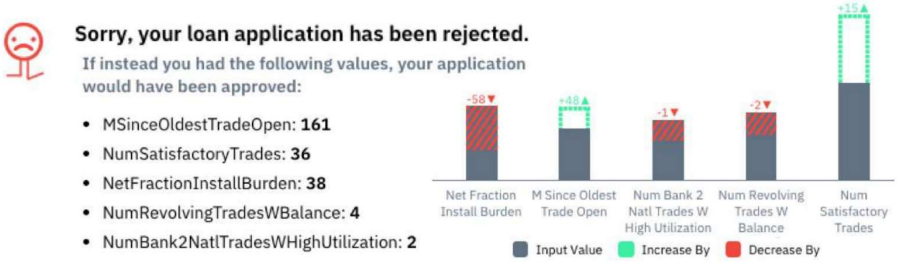
2

Counterfactual explanations

- Which features should be altered to obtain a different decision?
- Example:
  - Peter applies for a loan and gets rejected by the ML-method the bank uses for credit scoring.
  - He wonders why his application is rejected and how he might improve his chances to get a loan.
  - This question may be formulated as a counterfactual:
    "What is the smallest change to the features (e.g. income, age, number of credit cards) that would change the prediction from rejected to approved?"

Source: finbucket.com

Example



What is a good counterfactual explanation?

1. **The explanation should produce the predefined prediction as closely as possible.**
   - Ex: Assume that Peter's current probability of default is 5% and that he gets a loan if the predicted probability of default is less than 1%. Then, the question is: *"What are the minimal changes in Peter's features so that his probability becomes as close to 1% as possible".*
2. **We should change the feature vector as little as possible.**
3. **We should change as few features as possible.**

What is a good counterfactual explanation?

4. **The explanation should have feature values that are likely**
   - It makes no sense to tell Peter that he should change the number of transactions on his checking every month account to zero, and at the same time keep his current monthly transaction amount or that he should change his sex.

"I'd like to be a woman, please."

Source: cartoonstock.com

Multiple diverse explanations

It might be desirable to generate multiple diverse explanations

- Some explanations are not possible or convenient
- Ex: One explanation might be for Peter to double his income, while another might be to move to another city.

Source: europemoving.eu

Source: worldonyou.com

Methods

- Simple trial and error
- A method by Wachter et al. (2017)
- A method by Dandl et al. (2020)
- A method by Redelmeier et. al (2023)

Wachter et. al (2018)

- Optimization problem:

$$\arg\min_{x'} \max_{\lambda} \lambda(f_w(x') - y')^2 + d(x_i, x')$$

Any suitable optimization algorithm can be used to minimize the loss function, e.g. Nelder-Mead.

- y' is the desired model outcome
- x_i is the current vector of covariates for person i
- f_w is the Black-box model
- d() is given by

$$d(x, x') = \sum_{j=1}^{p} \frac{|x_j - x'_j|}{MAD_j} \qquad MAD_j = \text{median}_{i\in\{1,\ldots,n\}}(|x_{i,j} - \text{median}_{l\in\{1,\ldots,n\}}(x_{l,j})|)$$

- λ balances the two terms.

9

Optimization

- To minimize this loss function, any suitable optimization algorithm can be used, such as Nelder-Mead.
- If you have access to the gradients of the machine learning model, you can use gradient-based methods like ADAM.
- Start with a low initial value for λ and a random initial x'.
- Find the x' which minimizes the loss function using this value of λ.
- While $|\hat{f}(x') - y'| > \epsilon$ where ϵ is a tolerance parameter
  - Increase λ and find the x' which minimizes the loss function.

Dandl et. al. (2020)

- Multi-objective optimization problem:

$$L(x, x', y', X^{obs}) = \left(o_1(\hat{f}(x'), y'), o_2(x, x'), o_3(x, x'), o_4(x', X^{obs})\right)$$

- Each objective corresponds to one of the 4 requirements.

Objectives

- Objective 1: $\hat{f}(x')$ should be as similar as possible to y'

$$o_1(\hat{f}(x'), y') = \begin{cases} 0 & \text{if } \hat{f}(x') \in y' \\ \inf\limits_{y' \in y'} |\hat{f}(x') - y'| & \text{else} \end{cases}$$

- Objective 2: x' should be as similar as possible to x

$$o_2(x, x') = \frac{1}{p} \sum_{j=1}^{p} \delta_G(x_j, x'_j)$$

Gower distance:

$$\delta_G(x_j, x'_j) = \begin{cases} \frac{1}{\hat{R}_j} |x_j - x'_j| & \text{if } x_j \text{ numerical} \\ \mathbb{I}_{x_j \neq x'_j} & \text{if } x_j \text{ categorical} \end{cases}$$

$\hat{R}_j$ is the observed value range of feature j

Objectives

- Objective 3: We should change as few features as possible

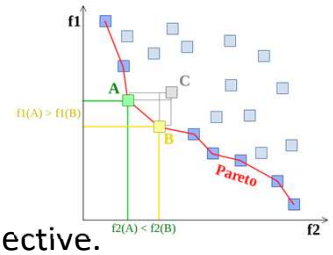
$$o_3(x, x') = ||x - x'||_0 = \sum_{j=1}^{p} \mathbb{I}_{x'_j \neq x_j}.$$

- Objective 4: The explanation should be likely

$$o_4(x', \mathbf{X}^{obs}) = \frac{1}{p} \sum_{j=1}^{p} \delta_G(x'_j, x_j^{[1]})$$

Compute the distance between the explanation and the nearest observed data point.

Multi-objective optimisation



- For a multi-objective optimization problem, no single solution exists that simultaneously optimizes each objective.
- In that case, the objective functions are said to be conflicting, and there exists a (possibly infinite) number of equally good Pareto optimal solutions.
- A solution is called Pareto optimal if none of the objective functions can be improved in value without degrading some of the other objective values.
- Use the Nondominated Sorting Genetic Algorithm (NSGA-II) to determine the Pareto frontier.

Disadvantages with optimisation-based methods

- Quite slow
- Usually restrict the black-box model to be differentiable, meaning that they do not work for tree-based classifiers like XGBoost or random forest.
- Do not properly handle fixed features (e.g. age, sex and race)
- Do not produce realistic counterfactuals (e.g. properly modelling the correlation between the variables).
- Do not handle categorical variables with more than two levels.

MCCE (Redelmeier et. al, 2023*)

- MCCE: Monte Carlo sampling of valid and realistic Counterfactual Explanations for tabular data
- Three steps:
  - Fits the joint distribution of the features and the decision with an autoregressive generative model where the conditionals are estimated using decision trees.
  - Samples a large set of observations from this model
  - Removes the samples that do not obey certain criteria.

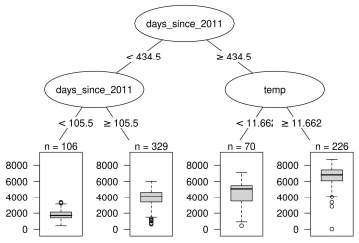
*Accepted for publication in *Data Mining and Knowledge Discovery*, January 2024.

Step 1: Autoregressive generative model

- Decompose the distribution of the data **X** into products of conditional probability distributions as follows:

$$p(\boldsymbol{X}) = p(X_1) \times \prod_{i=2}^{q} p(X_i | X_1, \ldots, X_{i-1}).$$

- Fit a classification tree (CART) to each conditional distribution.



From https://christophm.github.io/interpretable-ml-book/tree.html

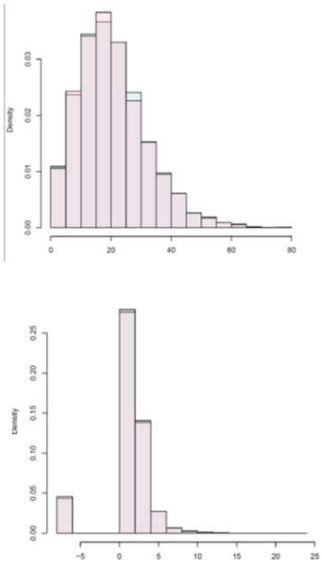
Classification trees can be fitted using the R-package **rpart.**

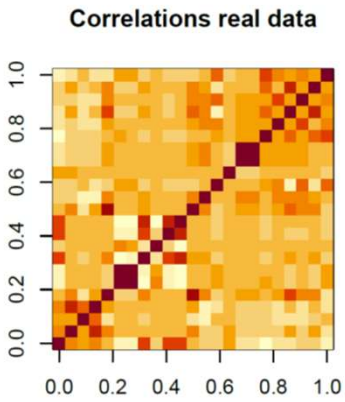
Step 2: Generation

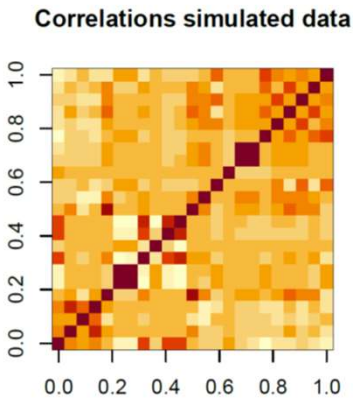
- Step 2 consists of generating a K x q dimensional data set **D**, by sequentially sampling from the conditional distributions.

1. Generate K simulations $D_{1,1}, \ldots D_{K,1}$ by sampling with replacement from the values of variable X_1 in the training data set.
2. For $j = 2, \ldots q$
   - For $k = 1, \ldots K$
     - Find the end node in the tree T_j to which the sample $D_{k,1}, .., D_{k,j-1}$ belongs
     - Select $D_{k,j}$ by randomly sampling one observation from this node.

Example







Fixed features

- Features like age, sex and race are usually assumed to be fixed
- This can easily be taken into account by replacing step 1 in the generation procedure by
  - For $j = 1, \ldots p$
    - For $k = 1, \ldots K$
      - $D_{j,k} = x_j^*$
- where *p* is the number of fixed variables and x_j^* is the fixed value of variable j.
- In addition, step 2 starts at j=p+1 instead of j=2.

Step 3: Postprocessing

- The last step removes the rows of **D** that do not satisfy certain criteria:
  - First, the prediction should be lower than a prespecified limit c.
  - Second, the number of features changed should be as small as possible.
  - Third, the Gower distance between the observation and the counterfactual should be as small as possible.

$$\text{Gower distance} = \frac{1}{q}\sum_{j=1}^{q}\delta_G(d_j, x_j) \in [0,1],$$

where

$$\delta_G(d_j, x_j) = \begin{cases} \frac{1}{R_j}|d_j - x_j| & \text{if } x_j \text{ is numerical,} \\ \mathbb{1}_{d_j \neq x_j} & \text{if } x_j \text{ is discrete, categorical, or ordinal,} \end{cases}$$

MCCE: Advantages and disadvantages

- **ADVANTAGES:**
  - Does not restrict the black-box model to be differentiable.
  - Does properly handle fixed features.
  - Does produce realistic counterfactuals
  - Does handle categorical variables with more than two levels.
  - Breaks up the task of generating counterfactuals into independent steps that can easily be altered without affecting the others.

- **DISADVANTAGES:**
  - May have problems with properly estimating the distribution of **X** when the dimension is high and the number of samples is low.

Example: Adult data

- Prediction task is to determine whether a person makes over USD 50K a year.
- Variables: Age, FNLWGT, Education, Capital.Gain, Capital.Loss, Hours.per.week, Marital.Status, Country, Occupation, Race, Relationship, Sex, Workclass.
- 30,718 persons, 24% makes over USD 50K a year.
- Split data in training (50%), validation (25%) and test (25%) sets.
- Fit a deep learning model.
- AUC for test set is 0.90.

Example

All competing methods, except FACE, are optimization-based.

Method	Age	FNLWGT	Edc.	Gain	Loss	Hr.	MS	Co.	Occ.	Race	Rel.	Sex	Work
Original	**20**	**266015**	**10**	**0**	**0**	**44**	**NM**	**US**	**O**	**NW**	**NH**	**M**	**P**
C-CHVAE	**20**	247240	**10**	652	1679	42	M	**US**	**O**	**NW**	H	**M**	**P**
CEM-VAE	23	190709	12	10296	**0**	52	**NM**	**US**	**O**	W	**NH**	**M**	NP
CLUE	11	398962	**10**	10725	-62	49	**NM**	**US**	**O**	W	**NH**	**M**	**P**
CRUDS	**20**	138021	21	4833	210	79	M	**US**	MS	W	H	M	**P**
FACE	46	220979	**10**	13550	**0**	40	**NM**	**US**	**O**	**NW**	**NH**	**M**	**P**
REViSE	**20**	76456	14	10	379	68	M	**US**	**O**	W	H	**M**	**P**
MCCE	**20**	348148	**10**	34095	**0**	48	**NM**	**US**	**O**	**NW**	**NH**	**M**	**P**

CEM-VAE, CLUE and FACE change the age which is regarded to be fixed
CEM-VAE, CLUE, CRUDS and REViSE change the race which is regarded to be fixed (“Not white”)

Binarized categorical features by partitioning them into the most frequent level and its counterpart.

Software

- Multi-object optimisation:
  - Both R and Python: https://github.com/susanne-207/moc
- MCCE:
  - Python: https://github.com/NorskRegnesentral/mccepy
  - R: https://github.com/NorskRegnesentral/mcceR

Break-out rooms

- Discuss the difference between LIME and Counterfactual explanations.

Old stuff

Nondominated Sorting Genetic Algorithm

- Use the Nondominated Sorting Genetic Algorithm (NSGA-II) to determine the Pareto frontier.
- NSGA-II is a nature-inspired algorithm that applies Darwin's law of the "survival of the fittest".
- The fitness of an explanation is its vector of objective values.
- The result of the optimisation is a diverse set of counterfactuals with different trade-offs between the four objectives.
- For more details, see Dandl et. al. (2020) or the "Interpretable Machine Learning Book".

Example: German credit card data

- Can be downloaded from https://www.kaggle.com/uciml/german-credit
- 522 complete observations and 9 features:
  - age: numeric
  - sex: female, male
  - job: 0 — unskilled and non-resident, 1 — unskilled and resident 2 — skilled, 3 — highly skilled
  - housing: own, rent, free
  - savings.account: little, moderate, rich
  - checking.account: little, moderate, rich
  - credit.amount: numeric (in DM)
  - duration: numeric (in months)
  - purpose: car, furniture, radio/tv, others
- Response:
  - risk good or bad

German credit card data ex. 1

- Use a Support Vector Machine (SVM) for prediction (AUC=0.63).
- All observations except for the one to explain is used for training.
- Want to generate contrafactuals for the following observation:

age	sex	job	housing	saving.accounts	checking.account	credit.amount	duration	purpose	PD
58	female	1	free	little	little	6143	48	car	72,3%

PD is probability of “bad risk”

- How to change the input features so that the PD is smaller than 50%?

German credit card data ex. 1

Get 12 counterfactuals which satisfy the target 0 <= PD < 50%

age	sex	job	housing	saving.accounts	checking.account	credit.amount	duration	purpose	PD
58	female	1	free	little	little	6143	48	car	72,3%

	age	sex	job	housing	saving.accounts	checking.account	credit.amount	duration	purpose	dist.target	dist.x.interest	nr.changed	dist.train	pred
1	58	female	1	free	little	rich	6143	48	car	0	0.1111111	1	0.170464418	0.49
2	58	female	2	free	little	little	6143	7	car	0	0.1060606	2	0.095124153	0.48
3	58	female	3	free	little	little	6143	22	car	0	0.1178451	2	0.032834591	0.48
4	57	female	2	free	little	little	6143	8	car	0	0.1063612	3	0.091456524	0.49
5	54	female	3	free	little	little	6143	21	car	0	0.1274651	3	0.026581585	0.49
6	50	female	3	free	little	little	6143	20	car	0	0.1370851	3	0.020328578	0.50
7	47	female	3	free	little	little	6143	18	car	0	0.1464045	3	0.017743201	0.49
8	46	female	3	free	little	little	6143	18	car	0	0.1483886	3	0.015759074	0.50
9	58	male	2	free	little	rich	6143	41	car	0	0.2710438	4	0.013898762	0.38
10	36	male	3	free	little	little	6143	21	car	0	0.2742905	4	0.010199527	0.41
11	36	male	3	free	little	little	6143	20	car	0	0.2759740	4	0.008516025	0.41
12	31	male	3	free	little	little	6143	25	car	0	0.2774772	4	0.008321165	0.45

o_1 o_2 o_3 o_4

German credit card data ex. 2

- How to change the input features so that the PD is smaller than 50%?

age	sex	job	housing	saving.accounts	checking.account	credit.amount	duration	purpose	PD
22	female	2	own	little	moderate	5951	48	radio/TV	63,0%

- 57 counterfactuals with PD smaller than 50%
  - 100% with «duration» changed to a lower value
  - 77% with «credit amount» changed to a lower value

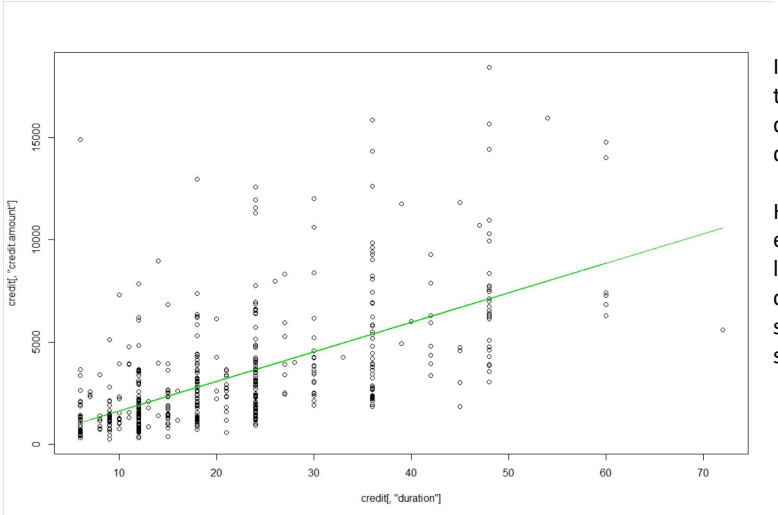
German credit card data ex. 2

PD as a function of “duration” and “credit.amount” when keeping the other features fixed.

	Dur = 10	Dur=20	Dur = 30	Dur = 40	Dur = 50
Credit.amount=1000	0.29	0.34	0.42	0.51	0.60
Credit.amount=2000	0.28	0.33	0.42	0.51	0.60
Credit.amount=3000	0.27	0.33	0.41	0.51	0.60
Credit.amount=4000	0.27	0.33	0.41	0.52	0.61
Credit.amount=5000	0.27	0.33	0.42	0.52	0.61

Why is credit.amount changed in 77% of the counterfactuals ?

German credit card data ex. 2



In the training data there is a tendency of credit.amount decreasing when duration is decreased.

Hence, for objective 4 (“the explanation should be likely”) to be satistified, credit.amount should be smaller when duration is smaller.