Global model-agnostic explanations and surrogate models

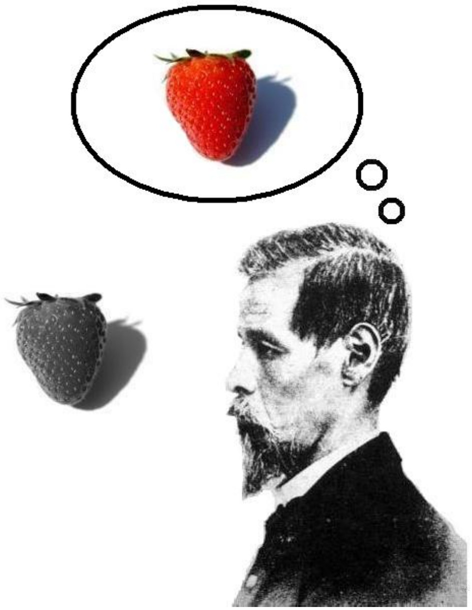
Szymon Bobek

Jagiellonian University
2024

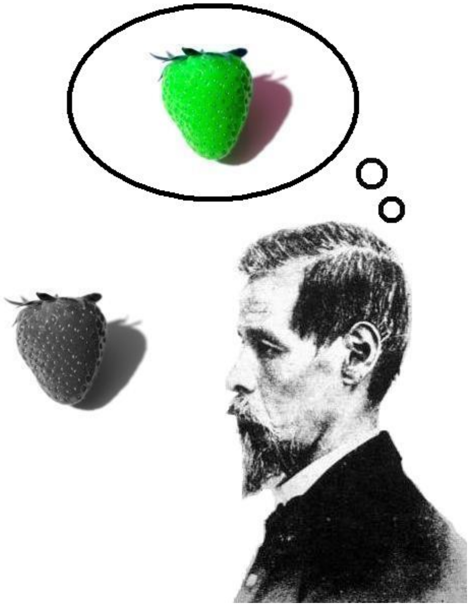
JAGIELLONIAN UNIVERSITY
IN KRAKÓW

https://geist.re

Fun with XAI – The inverted spectrum



It's gray!

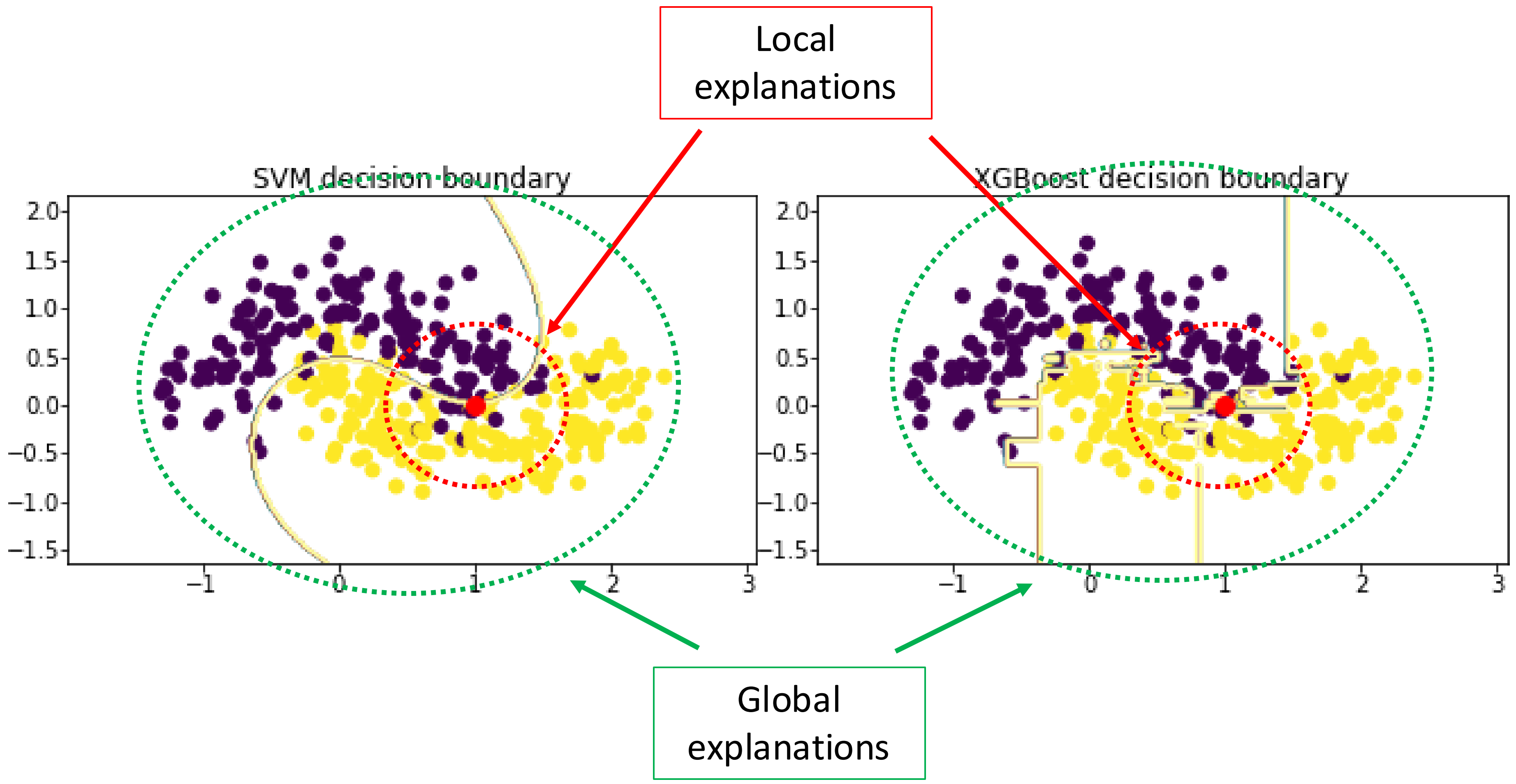


It's gray!

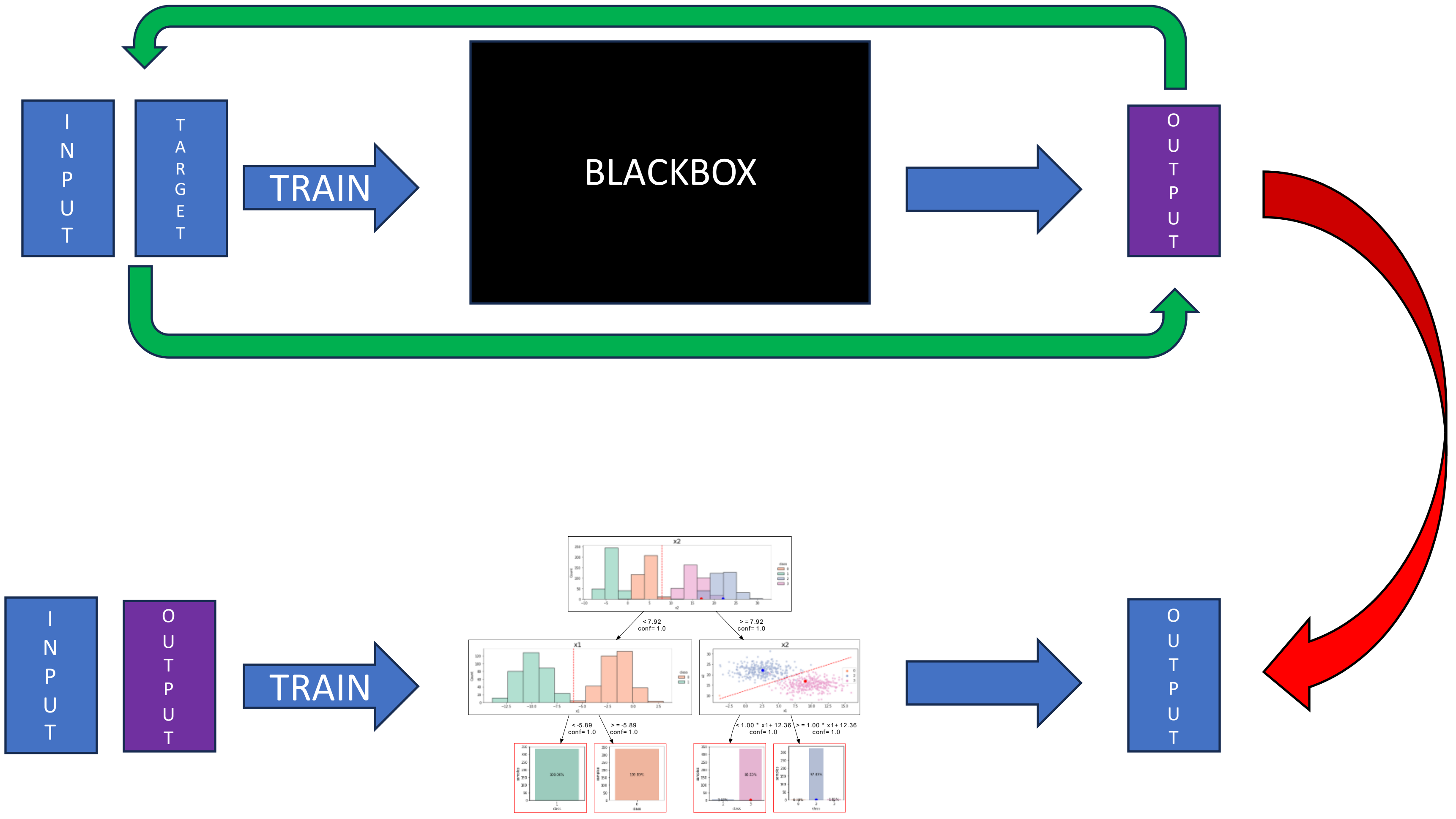
- John Locke's hypothetical concept
- Envisions a scenario where two people perceive colors differently (one sees red as blue and vice versa) but behave identically.
- It raises questions about subjective experiences and if they can be explained purely by physical processes.
- Does DNN "think" the same was as we do? Does it matter for XAI that they do not?

Global model-agnostic explanations

Local vs Global explanations



Global surrogate models



- Train blackbox model on training data
- Make predictions for the data and use predicted labels/values as target values for surrogate model
- Train interpretable model such as linear regression, decision tree, etc.
- Use surrogate model to explain blackbox model

Partial dependence plots

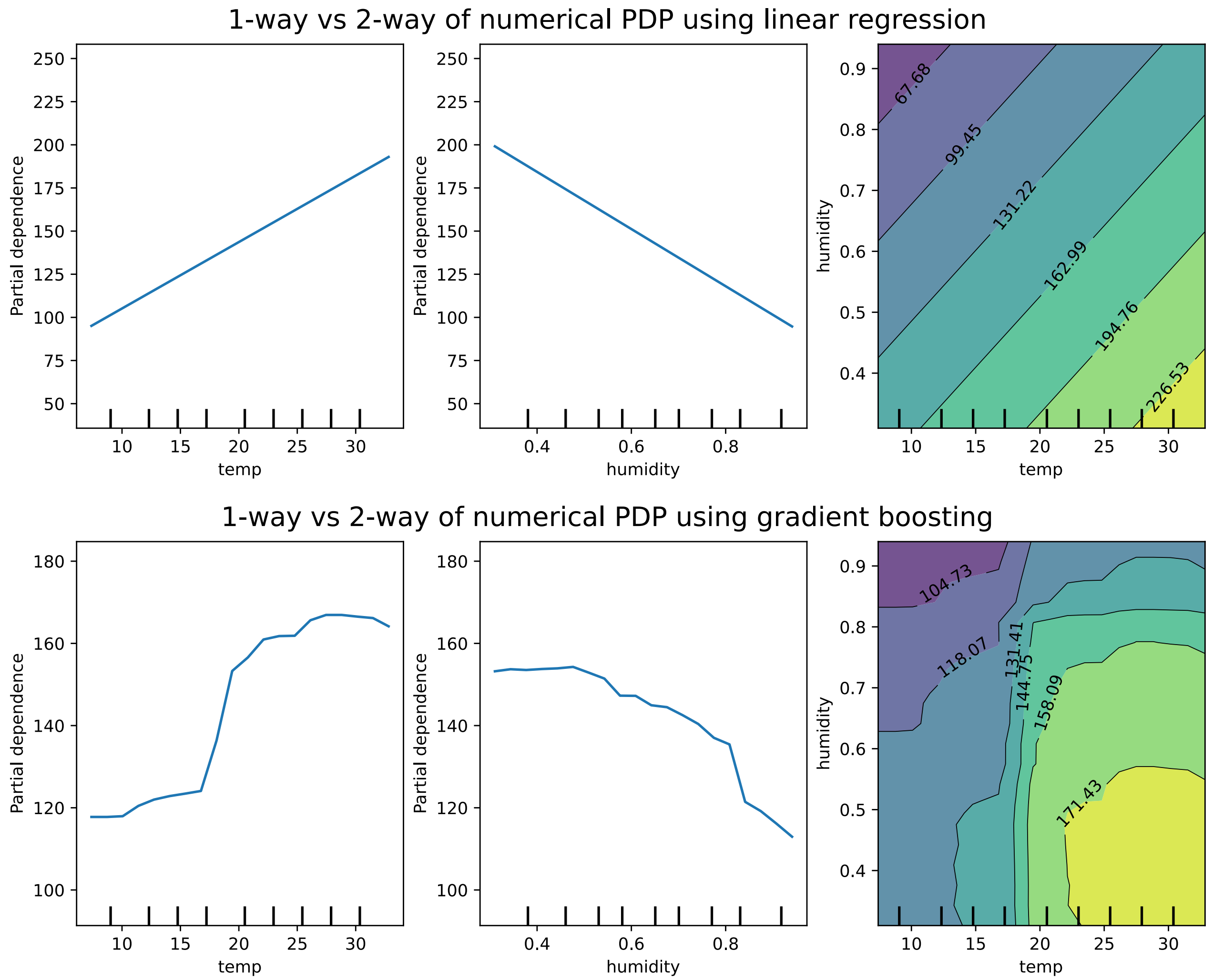
- **Feature interactions cannot be captured by all types of models**

Average over all instances

$$\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n}\hat{f}(x_S, x_C^{(i)})$$

Features for which we estimate partial dependence. Cannot be correlated with other features

Features values from dataset we are not interested in



How to measure feature interaction?

- Partial dependence function
- **H-statistic**

$$PD_s(x_s) = \hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$$

$$PD_{jk}(x_j, x_k) = PD_j(x_j) + PD_k(x_k)$$

$$\hat{f}(x) = PD_j(x_j) + PD_{-j}(x_{-j})$$

$$H_{jk}^2 = \frac{\sum_{i=1}^{n}\left[PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)})\right]^2}{\sum_{i=1}^{n} PD_{jk}^2(x_j^{(i)}, x_k^{(i)})}$$

$$H_j^2 = \frac{\sum_{i=1}^{n}\left[\hat{f}(x^{(i)}) - PD_j(x_j^{(i)}) - PD_{-j}(x_{-j}^{(i)})\right]^2}{\sum_{i=1}^{n} \hat{f}^2(x^{(i)})}$$

If we **assume centered** (mean zero) prediction and PD functions, the two-way PD functions can be decomposed into sum of one-way PD functios if there are no interactions

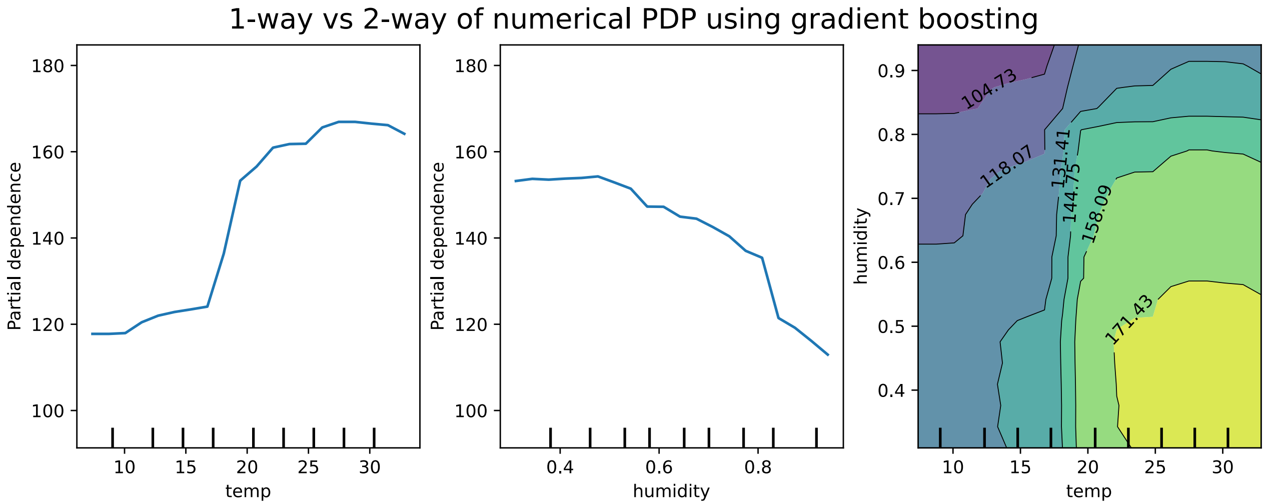
In such a case the prediciton function can be decomposed into the following term

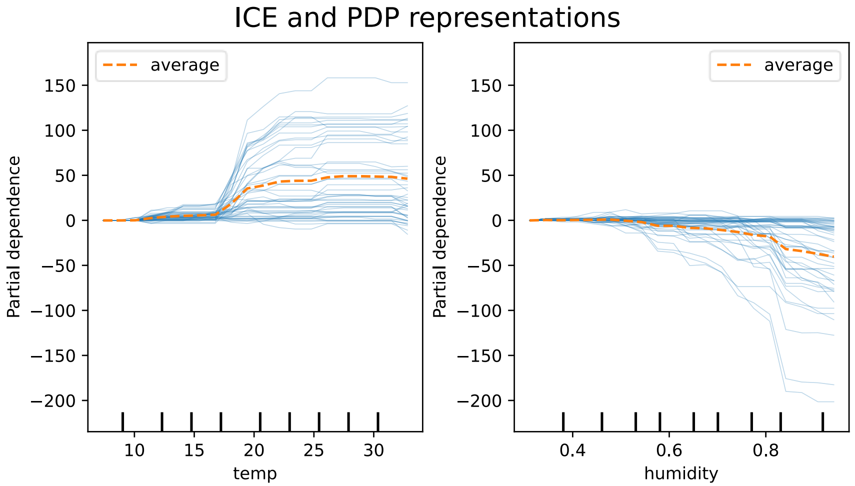
We can use the above to estimate the strength of dependence of two features:
It can be 0 -> No interactions
It can be 1 -> Effect comes only through interaciton

We can use above to estimate the strength of dependence of a features and all other features

Individual Conditional Expectation

- (ICE) plots display one line per instance that shows how the instance's prediction changes when a feature changes.
- For convenience we start from 0 by subtracting from all plots the prediction of the lower value of the feature of consideration
- The average of ICE curves from the PDP
- It is even easier to spot if there are interactions captured by model. If the ICE curves are not parallel, there are some interactions
- They give more insight into data, as average may cancel out some opposite effects





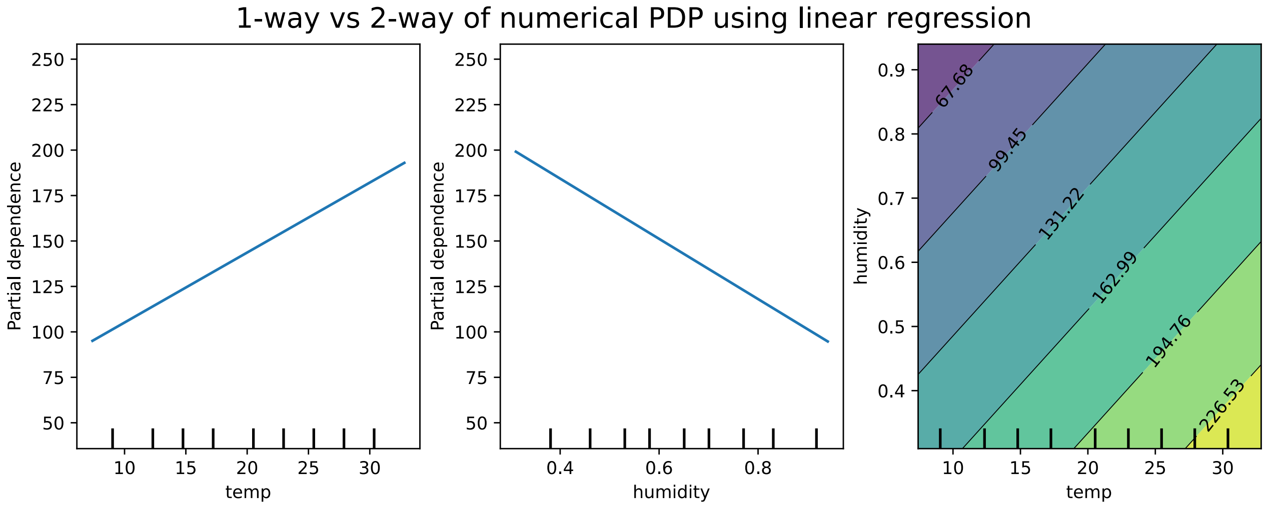
$$\hat{f}^{i}{}_{S}(x_S) = \hat{f}(x_S, \underbrace{x_C^{(i)}}_{fixed})$$

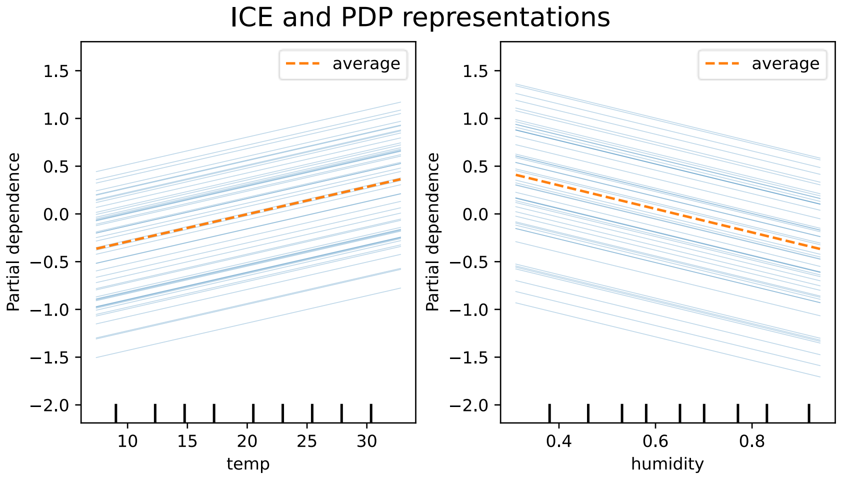
$$\hat{f}^{(i)}_{cent}(x_S) = \hat{f}^{(i)} - \hat{f}(x^a, x_C^{(i)})$$

Anchor point, usually the lower value of a feature we are plotting

Individual Conditional Expectation

- (ICE) plots display one line per instance that shows how the instance's prediction changes when a feature changes.
- For convenience we start from 0 by subtracting from all plots the prediction of the lower value of the feature of consideration
- The average of ICE curves from the PDP
- It is even easier to spot if there are interactions captured by model. If the ICE curves are not parallel, there are some interactions
- They give more insight into data, as average may cancel out some opposite effects





$$\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$$

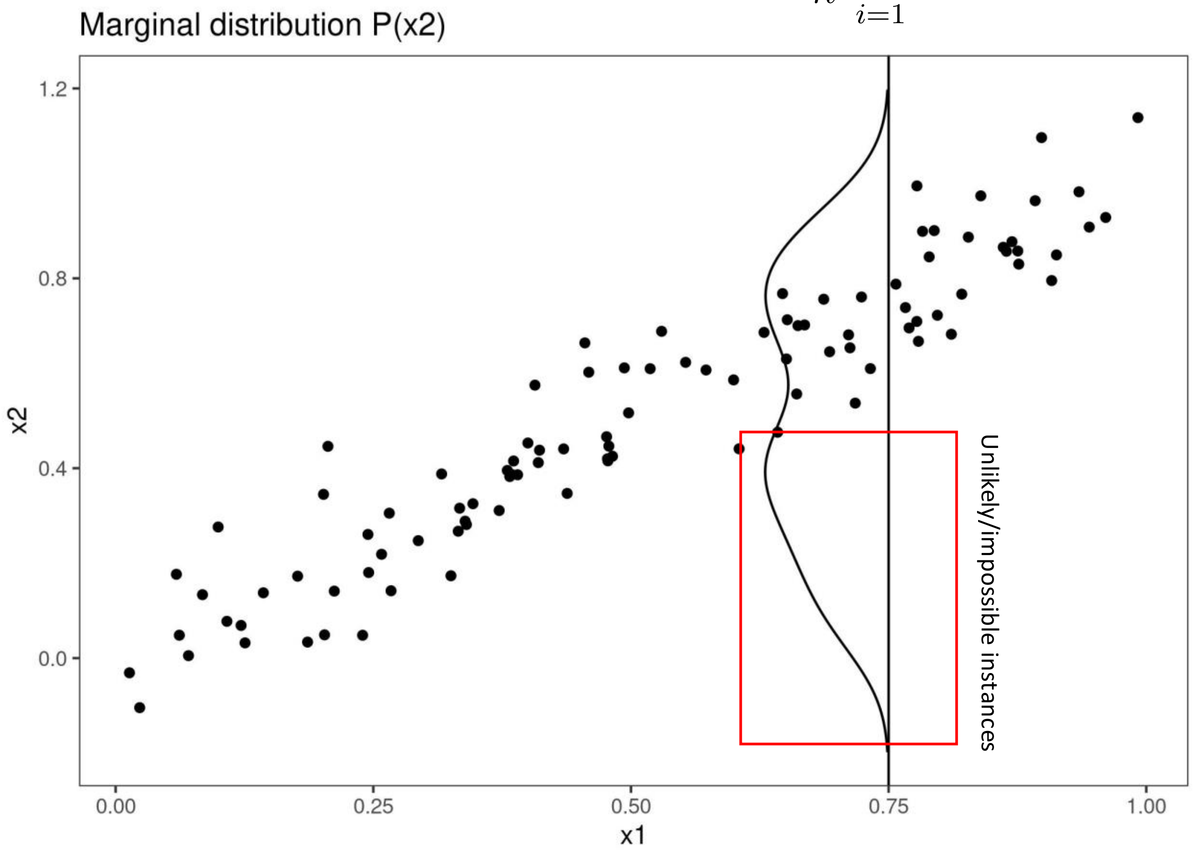
$$\hat{f}^{(i)}_{cent}(x_S) = \hat{f}^{(i)} - \hat{f}(x^a, x_C^{(i)})$$

Anchor point, usually the lower value of a feature we are plotting

Accumulated local effects

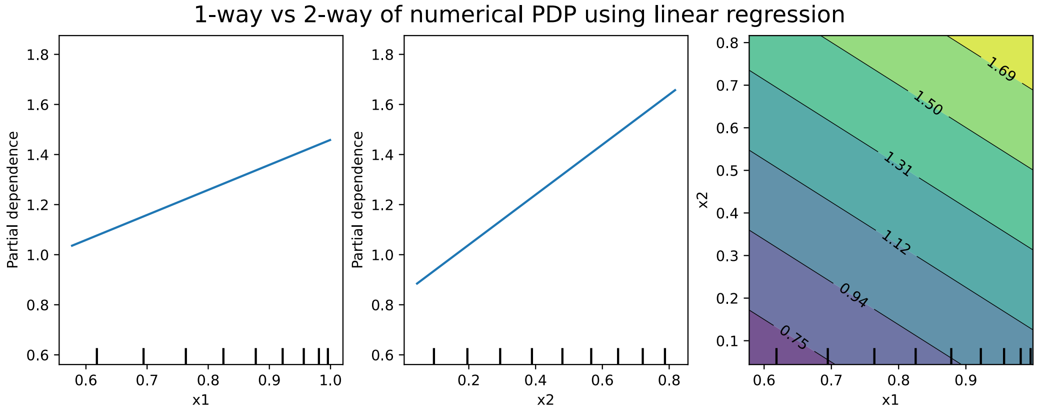
- PDP average effect of each feature value over all dataset, meaning that it also substitutes to the equation highly unlikely combinantions
- For instance for the x_1 value on the left, the 0.0 value of x_2 does not exist in the data, but will be used to calculate PDP

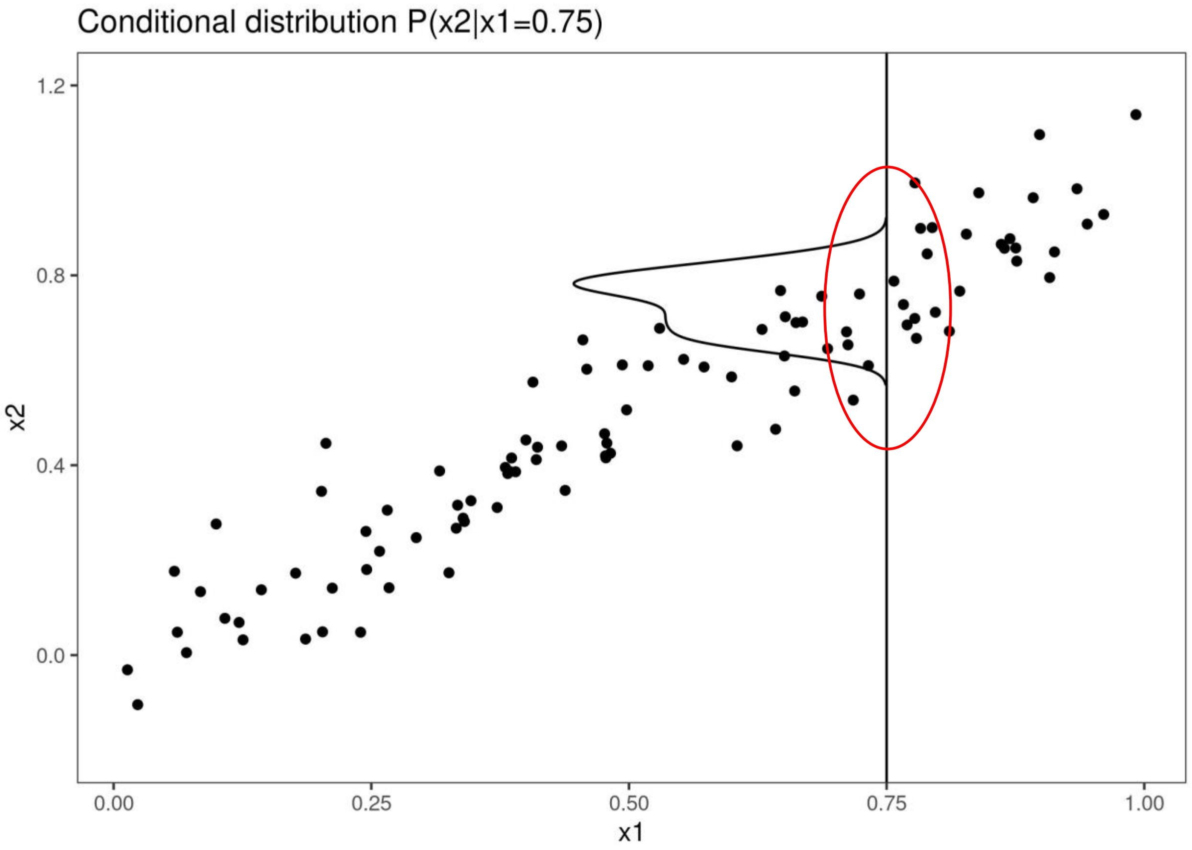
$$\hat{f}_S(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$$



M-Plots as a partial solution

- One solution might be to narrow down the calculation of PDP only to closest neigbourhood of the feature value for which we average the effect
- We deal with unlikely features, but still mix the effect of correlated features



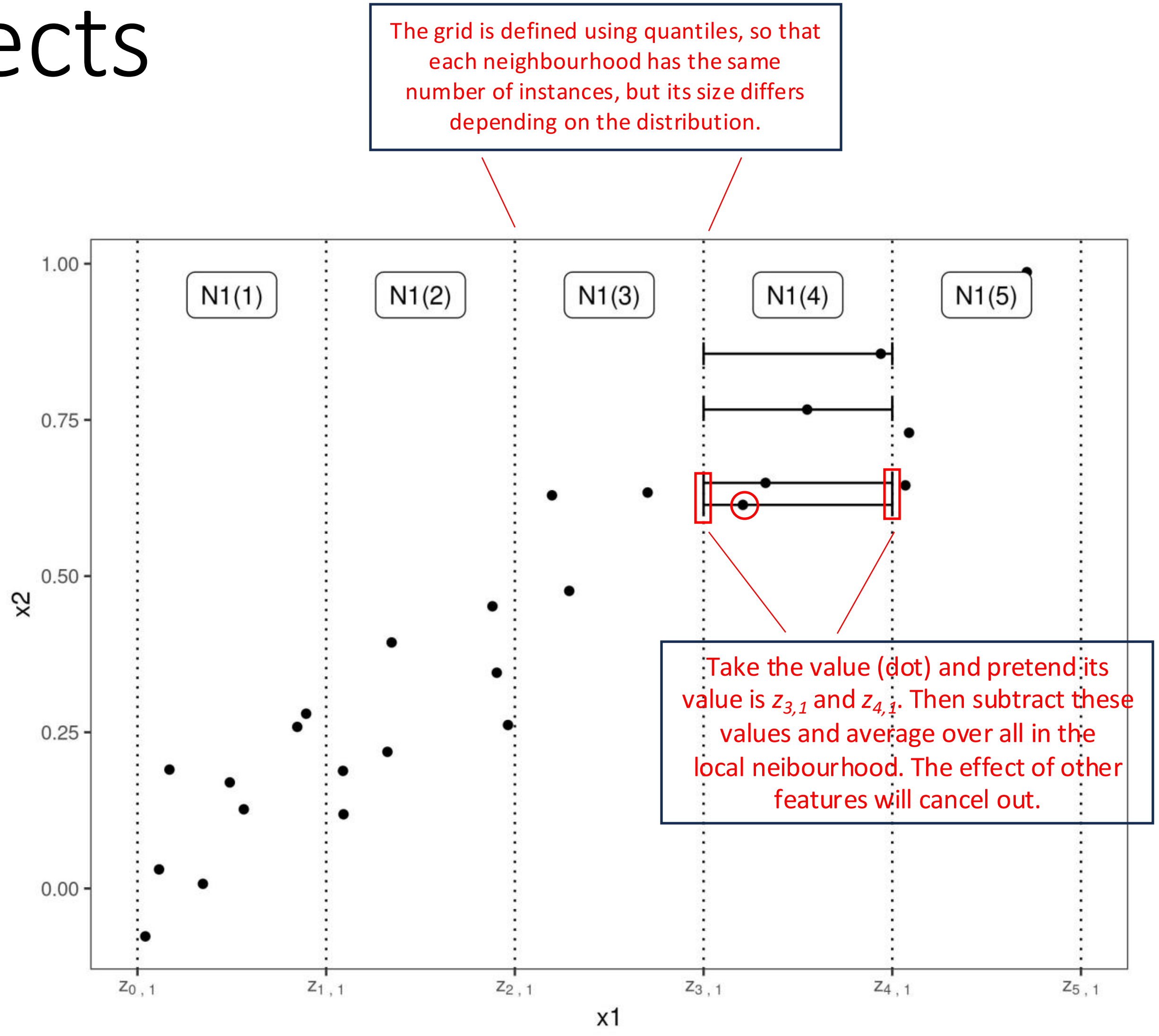


Accumulated local effects

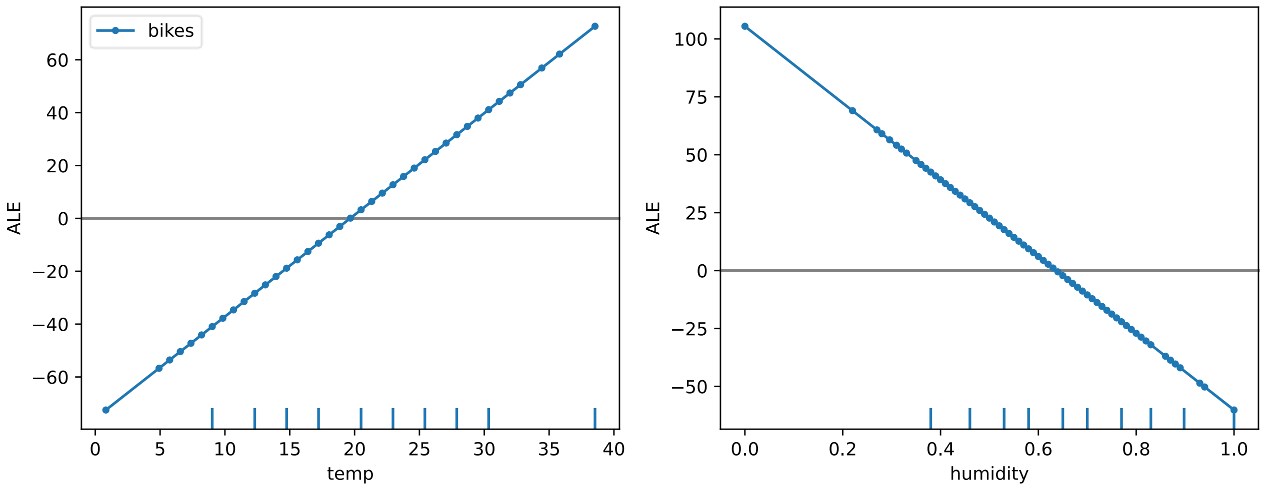
- ALE plots focus on changes in prediction for defined intervals
- ALE plots do not mix effects of correlated features
- It is like calculating partial derivative with respect to x_j *and averaging this effect*

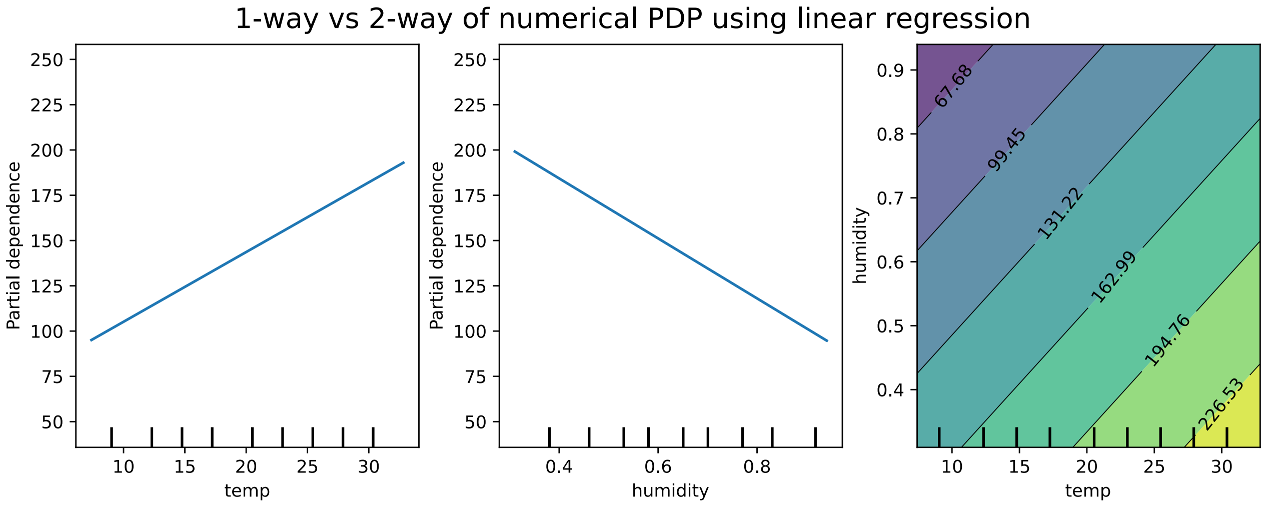
$$\hat{\hat{f}}_{j,ALE}(x) = \sum_{k=1}^{k_j(x)} \frac{1}{n_j(k)} \sum_{i:x_j^{(i)} \in N_j(k)} \left[\hat{f}(z_{k,j}, x_{-j}^{(i)}) - \hat{f}(z_{k-1,j}, x_{-j}^{(i)})\right]$$

$$\hat{f}_{j,ALE}(x) = \hat{\hat{f}}_{j,ALE}(x) - \frac{1}{n}\sum_{i=1}^{n} \hat{\hat{f}}_{j,ALE}(x_j^{(i)})$$



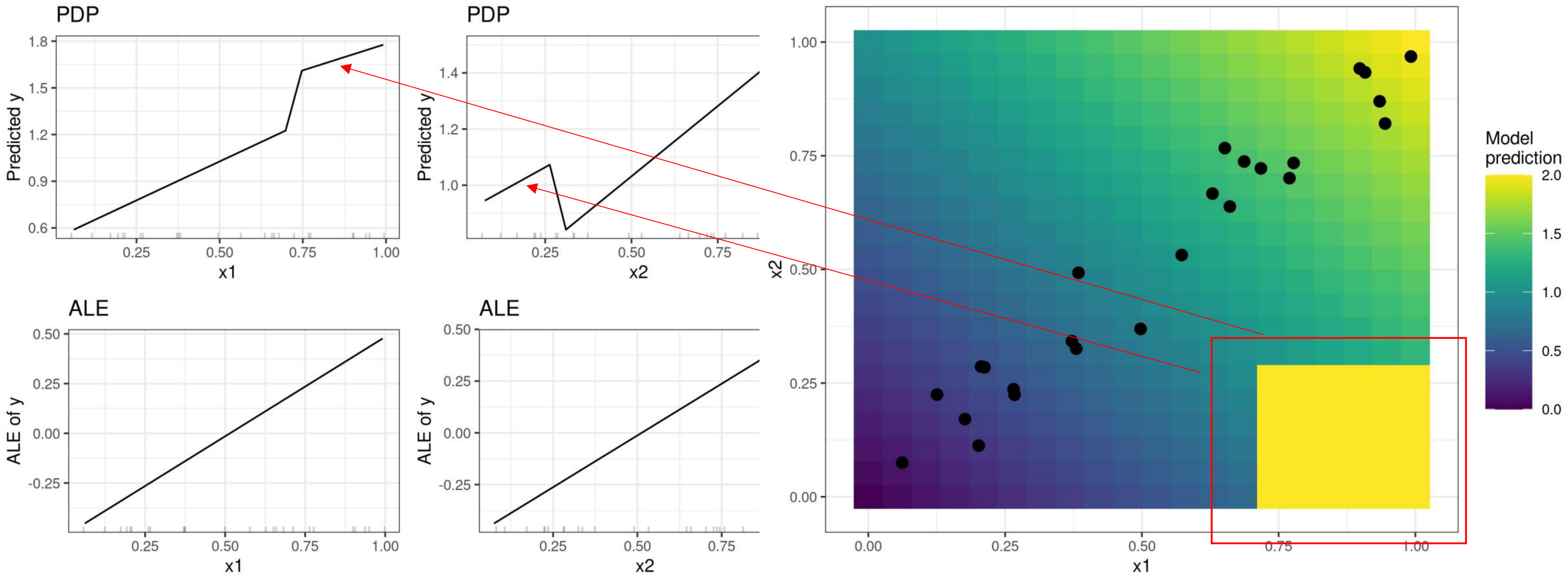
Problem of correlated features solved



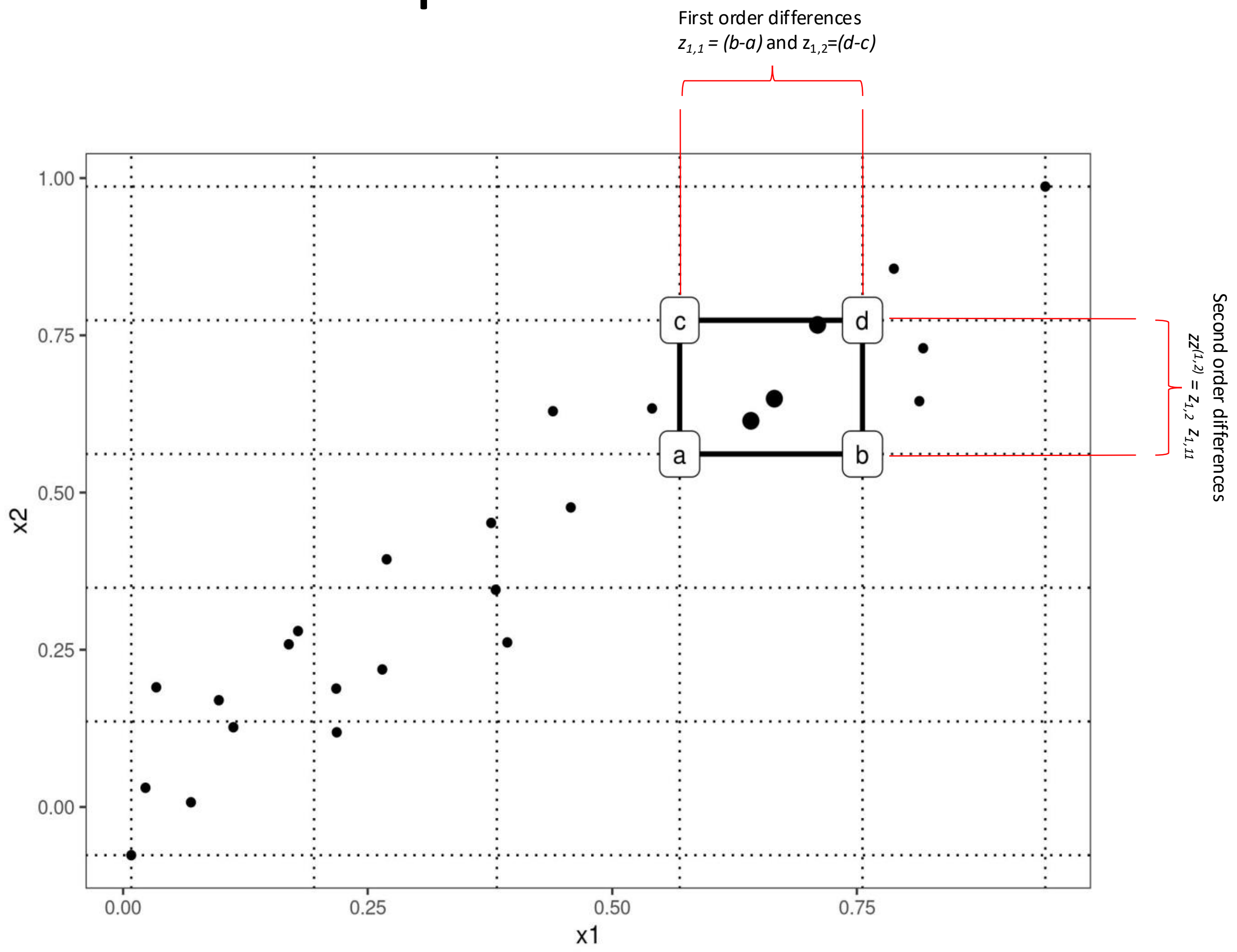


- The centered ALE at x_i shows how effect of x differs from the average effect of x.
- For additive models, a prediciton function can therefore be approximated as: $f(x) = \mathbb{E}(f(x)) + \sum_{i=1}^{d} \mathrm{ALE}(x_i)$
- Additionally by looking at the slope of ALE we can see how changing a x will change the prediction
- For instance, the slope for *temp* is around 3.5, which means that by increasing *temp* by one, 3.5 more bicycles will be predicted

Problem of unlikely instances solved

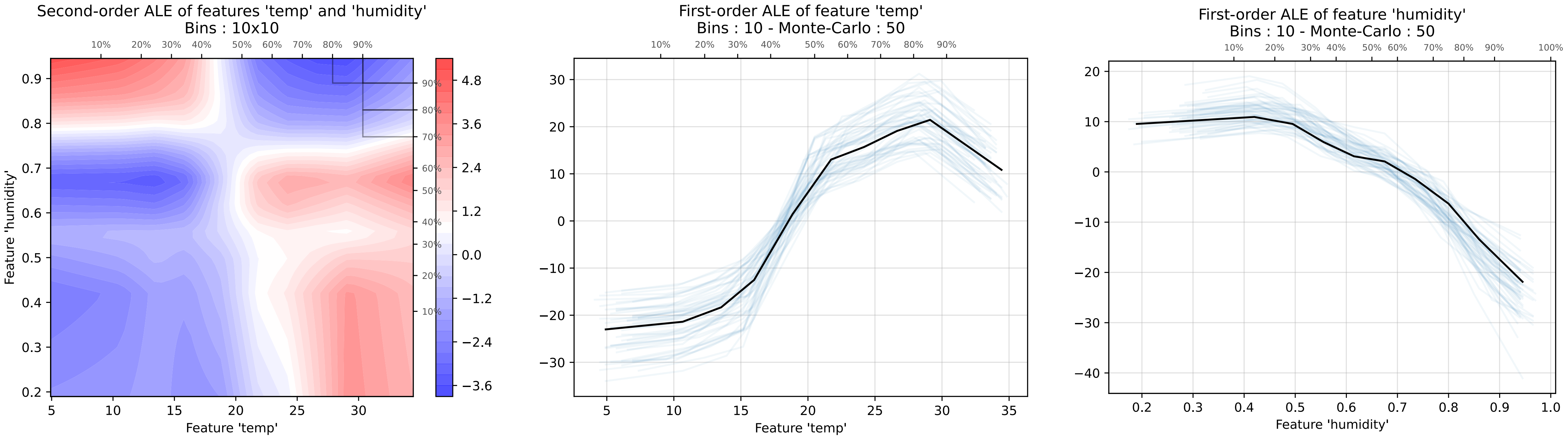


2D ALE plots for interaction effect



- 2D ALE plots exhibit second-order effect of two features
- The plot shows only additional effect of interaction between features
- Hence, no interaction results in constant zero 2D ALE plot

First order and second order effects



ALE for categorical variables

Sky	Temperature	Humidity	Rain
overcast	12	80	0
overcast	13	60	1
overcast	11	40	0
sunny	16	30	0
…	…	…	…
sunny	17	20	0
sunny	16	50	0
cloudy	11	80	1
cloudy	20	90	1
cloudy	12	70	0

- We need at least ordinal features to calculate ALE
- In case of categorical features we calculate distance matrix between subpopulations defined by different values of categorical variable
- The difference between these subpopulations can be Kolmogorov-Smirnof (similarity of 1D probability distributions) test for continuous and probability difference for categorical features
- Distance between subpopulations is sum of the above distances
- Such a distance matrix is then reduced with multidimensional scaling and we obtain order

Kol-Smir stat. for t_o and t_c

	Overcast	Sunny	Cloudy
Overcast	0	…	$\boxed{(t_o \sim t_c)}+(h_o \sim h_c)+(P(r_o)-P(r_c))$
Clear	…	0	…
Cloudy	…	…	0

We reduce this matrix to 1D

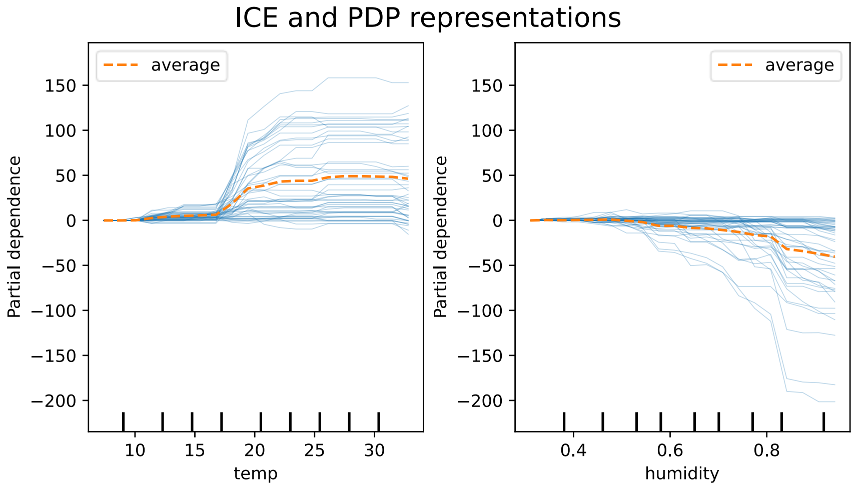
Dependence plots summary

- Partial Dependence Plots: “Let me show you what the model predicts on average when each data instance has the value v for that feature. I ignore whether the value v makes sense for all data instances.”
- M-Plots: “Let me show you what the model predicts on average for data instances that have values close to v for that feature. The effect could be due to that feature, but also due to correlated features.”
- ALE plots: “Let me show you how the model predictions change in a small”window” of the feature around v for data instances in that window.”

Permutation feature importance

PDP-based feature importance

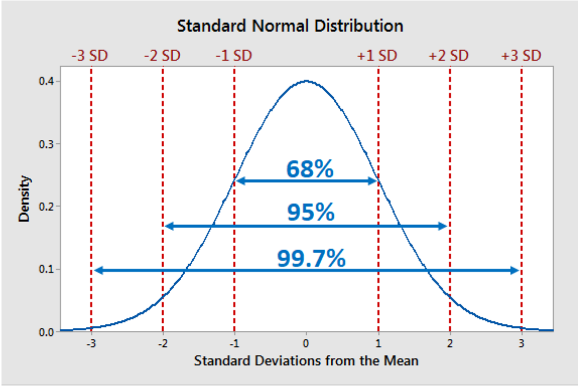
- Feature importance can be read from PDP/ICE plots (but, with caution)
- The less important feature, the flatter PDP is (i.e. constants are not important)



$$I(x_S) = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}\left(\hat{f}_S(x_S^{(k)}) - \frac{1}{K}\sum_{k=1}^{K}\hat{f}_S(x_S^{(k)})\right)^2}$$

$$I(x_S) = (max_k(\hat{f}_S(x_S^{(k)})) - min_k(\hat{f}_S(x_S^{(k)})))/4$$

The four comes from the fact that in normal distribution 95% of data is located between −2 and +2 stds
It is called the *range rule*

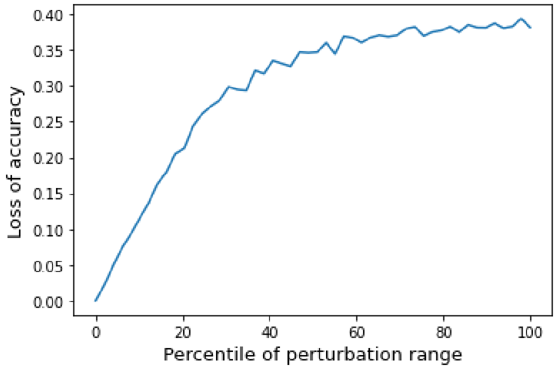


Permutation feature importance

- Estimate the model's error (e.g. MSE for regression)
- For each feature, perturb its value by shuffling it
- Observe how much the model's error increases (e.g. as a ration of original MSE to MSE after perturbation)
- Which data should we use to calculate feature improtance?

Temperature	Humidity	Wind	Sky
12	80	23	overcast
34	70	21	sunny
12	70	18	sunny
…	…	…	…
23	60	30	cloudy
4	50	4	cloudy

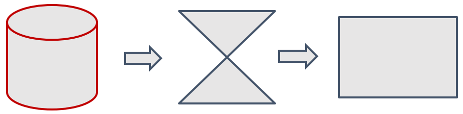
Alternatively you can add noise as perturbations and measure how the MSE increases with noise ratio



Prototype - critic

What in case of unsupervised learning?

- A prototype is a data instance that is representative of all the data.
- A criticism is a data instance that is not well-represented by the set of prototypes.
- You can build nearest-prototype predictor
- You can combine this with other XAI methods: find prototypes, predict with BBox, analyse with surrogate



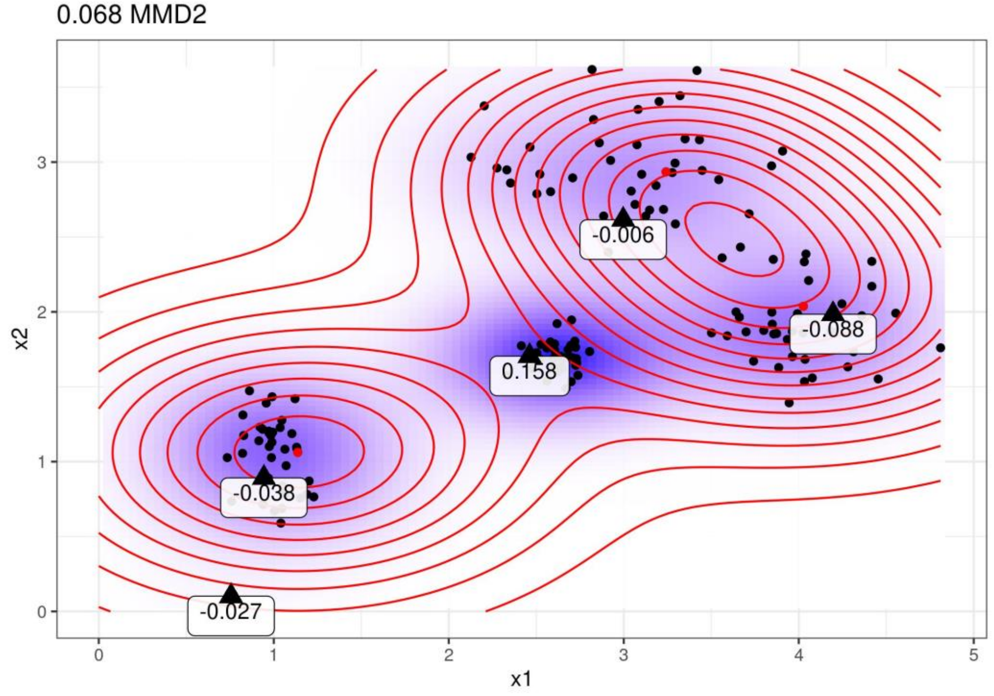
$$MMD^2 = \frac{1}{m^2}\sum_{i,j=1}^{m} k(z_i, z_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(z_i, x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j)$$

Minimize it

$$k(x, x') = exp\left(-\gamma ||x - x'||^2\right)$$

$$witness(x) = \frac{1}{n}\sum_{i=1}^{n} k(x, x_i) - \frac{1}{m}\sum_{j=1}^{m} k(x, z_j)$$

Maximize the absolute value of it



What in case of unsupervised learning?

- A prototype is a data instance that is representative of all the data.
- A criticism is a data instance that is not well-represented by the set of prototypes.
- You can build nearest-prototype predictor
- You can combine this with other XAI methods: find prototypes, predict with BBox, analyse with surrogate

$$MMD^2 = \frac{1}{m^2}\sum_{i,j=1}^{m} k(z_i, z_j) - \frac{2}{mn}\sum_{i,j=1}^{m,n} k(z_i, x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j)$$

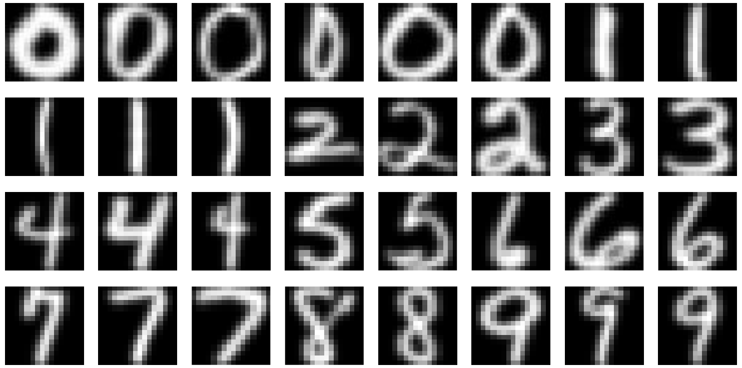
Minimize it

$$k(x, x') = exp\left(-\gamma||x - x'||^2\right)$$

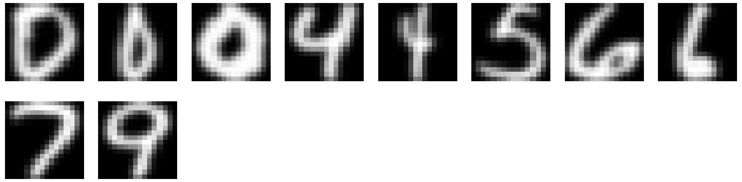
$$witness(x) = \frac{1}{n}\sum_{i=1}^{n} k(x, x_i) - \frac{1}{m}\sum_{j=1}^{m} k(x, z_j)$$

Maximize the absolute value of it

32 Prototypes



10 Criticisms



The prototypes are instances that well represent the distribution of data with respect to some kernel function. Criticisms are the opposite. It does not answer **what determines them – this is left for the data scientist**.

Thank you for your attention!

JAGIELLONIAN UNIVERSITY
IN KRAKÓW

https://geist.re