Explainable Machine Learning

Global Model-Agnostic Methods

Shim Jaewoong

jaewoong@seoultech.ac.kr

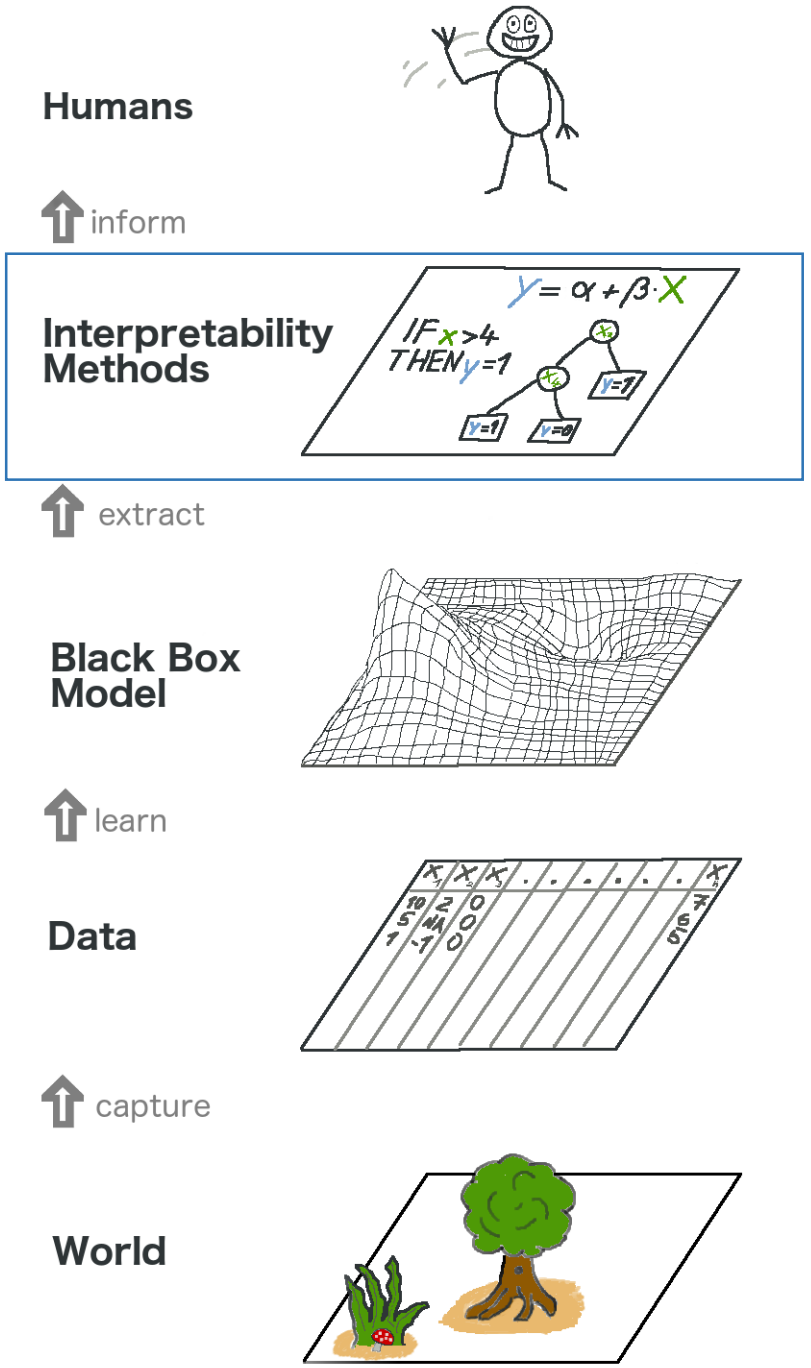
Model-Agnostic Methods

- Advantages of **Model-Agnostic methods**
  - free to use any machine learning model they like when the interpretation methods can be applied to any model
  - comparing models in terms of interpretability, it is easier to work with model-agnostic explanations, because the same method can be used for any type of model.

- Alternatives
  - **Using Interpretable models**
    - Predictive performance may be lost
  - **Model-specific methods**
    - Bind to one model type

Model-Agnostic Methods

- Big picture of model-agnostic interpretability
  - Interpretability is just another layer!



The real world goes through many layers before it reaches the human in the form of explanations.

Model-Agnostic Methods

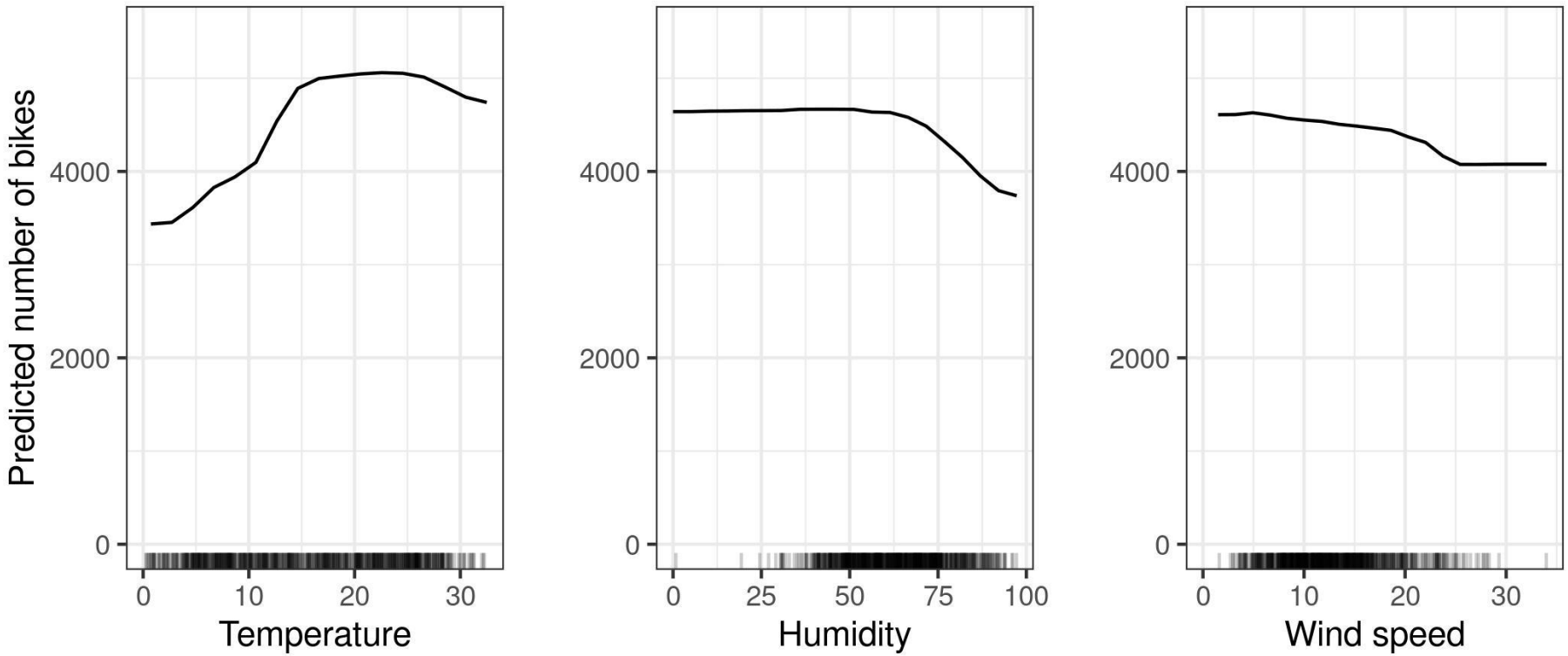
- Global model-agnostic
  - describe how features affect the prediction **on average**
  - Understand the general mechanisms

- Local model-agnostic
  - explain **individual predictions**

Partial Dependence Plot (PDP)

Partial Dependence Plot (PDP)

- Partial Dependence Plot
  - It shows the marginal effect one or two features have on the predicted outcome
  - It can show whether the relationship between the target and a feature is linear, monotonic or more complex



Partial Dependence Plot (PDP)

- The partial dependence function

$$\hat{f}_S(x_S) = E_{X_C}\left[\hat{f}(x_S, X_C)\right] = \int \hat{f}(x_S, X_C) d\mathbb{P}(X_C)$$

x_s : feature of interest
X_C : other features

By marginalizing over the other features,
we get a function that depends only on features in S

Practically, estimated by calculating averages in the training data

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

(Monte Carlo method)

- For classification problem:
  - (the machine learning model outputs probabilities) The partial dependence plot displays the probability for a certain class given different values for feature(s) in S. An easy way to deal with multiple classes is to draw one line or plot per class.
- For categorical features:
  - For each of the categories, we get a PDP estimate by forcing all data instances to have the same category.

Partial Dependence Plot (PDP)

- PDP-based Feature Importance
  - Motivation : flat PDP indicates that the feature is not important
  - Numerical features
    - the deviation of each unique feature value from the average curve

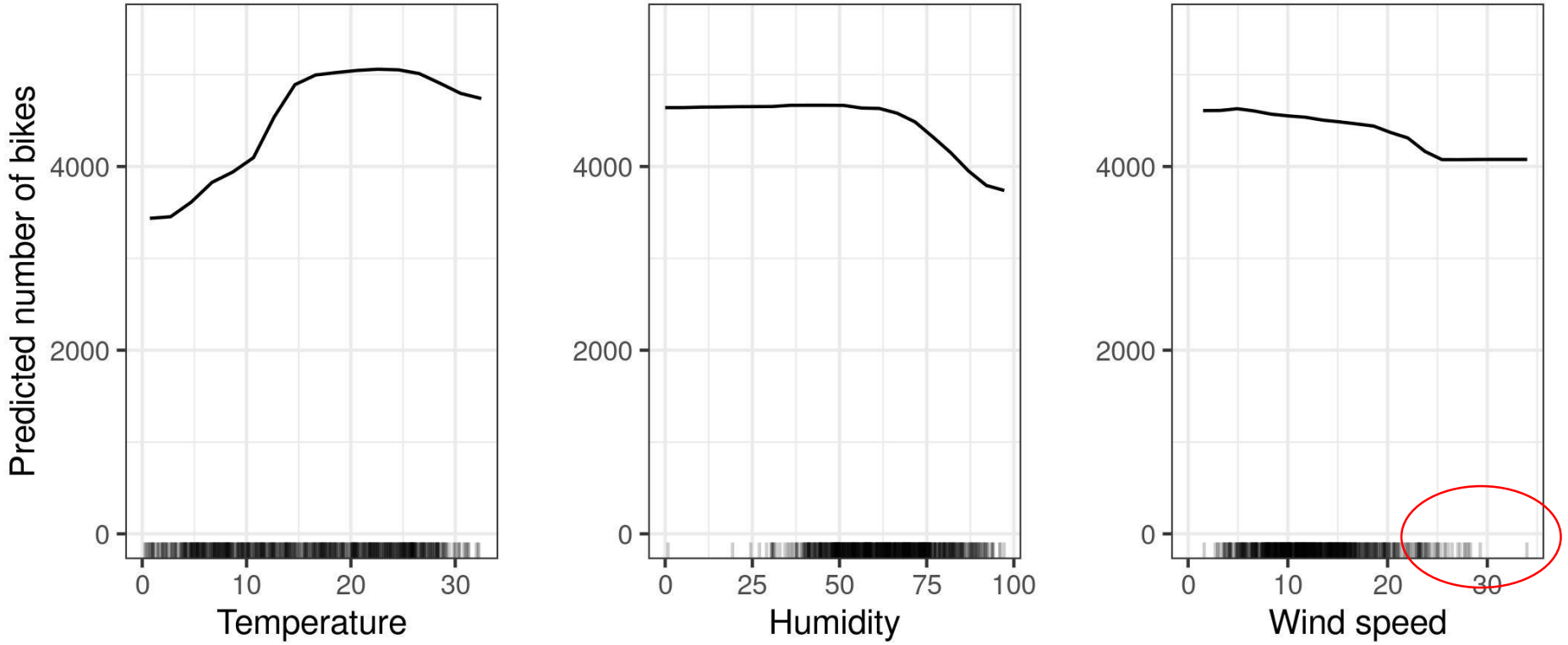
$$I(x_S) = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(\hat{f}_S(x_S^{(k)}) - \frac{1}{K}\sum_{k=1}^{K}\hat{f}_S(x_S^{(k)}))^2}$$

  - Categorical features
    - a rough estimate for the deviation when you only know the range.

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

Partial Dependence Plot (PDP)

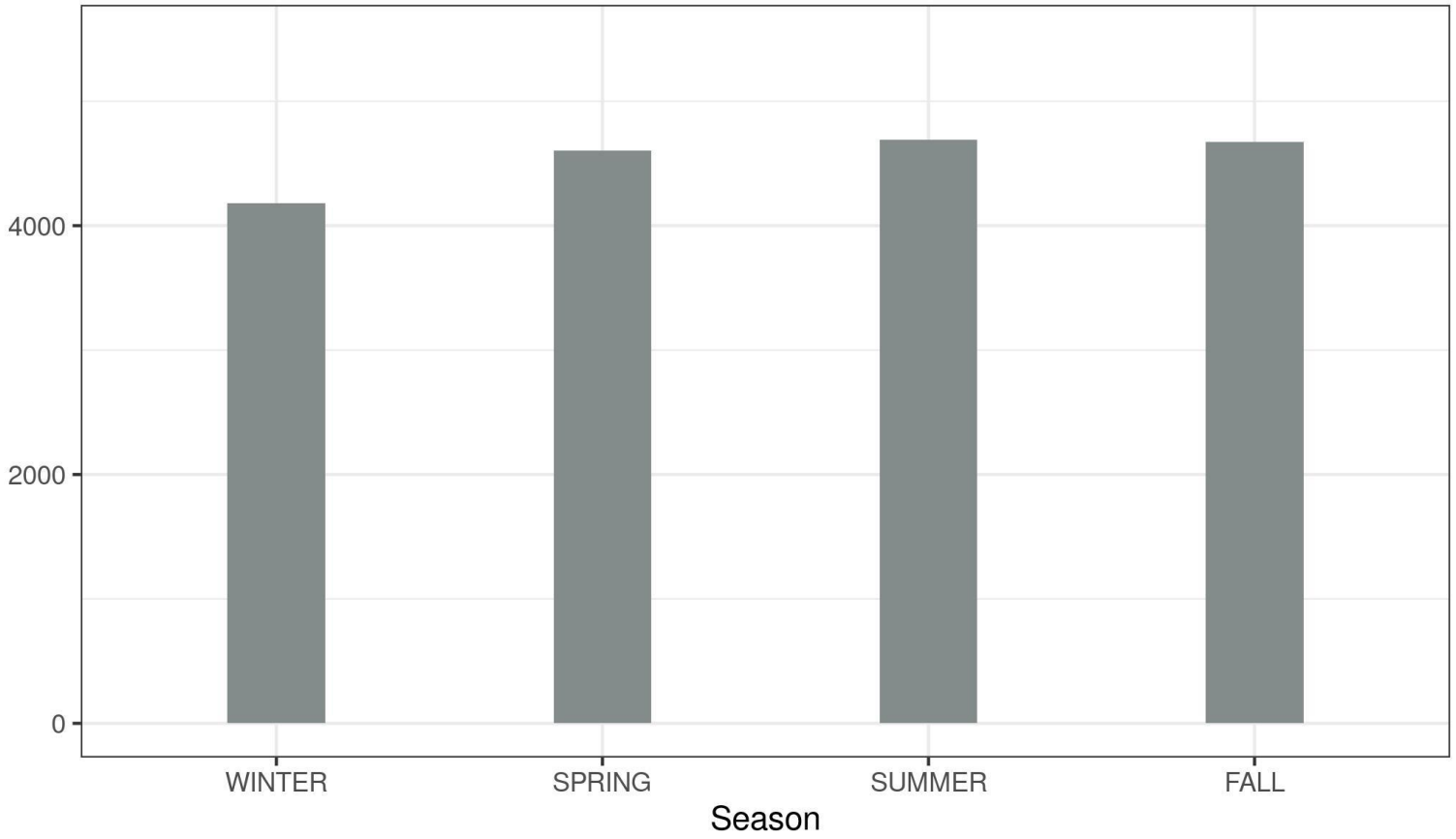
- Examples: bike rental (regression)
  - fitted a random forest to predict the number of bicycles and use the partial dependence plot to visualize the relationships the model has learned.



The predicted number of bike rentals does not fall when wind speed increases from 25 to 35 km/h, but there is not much training data, so the machine learning model could probably not learn a meaningful prediction for this range.

Partial Dependence Plot (PDP)

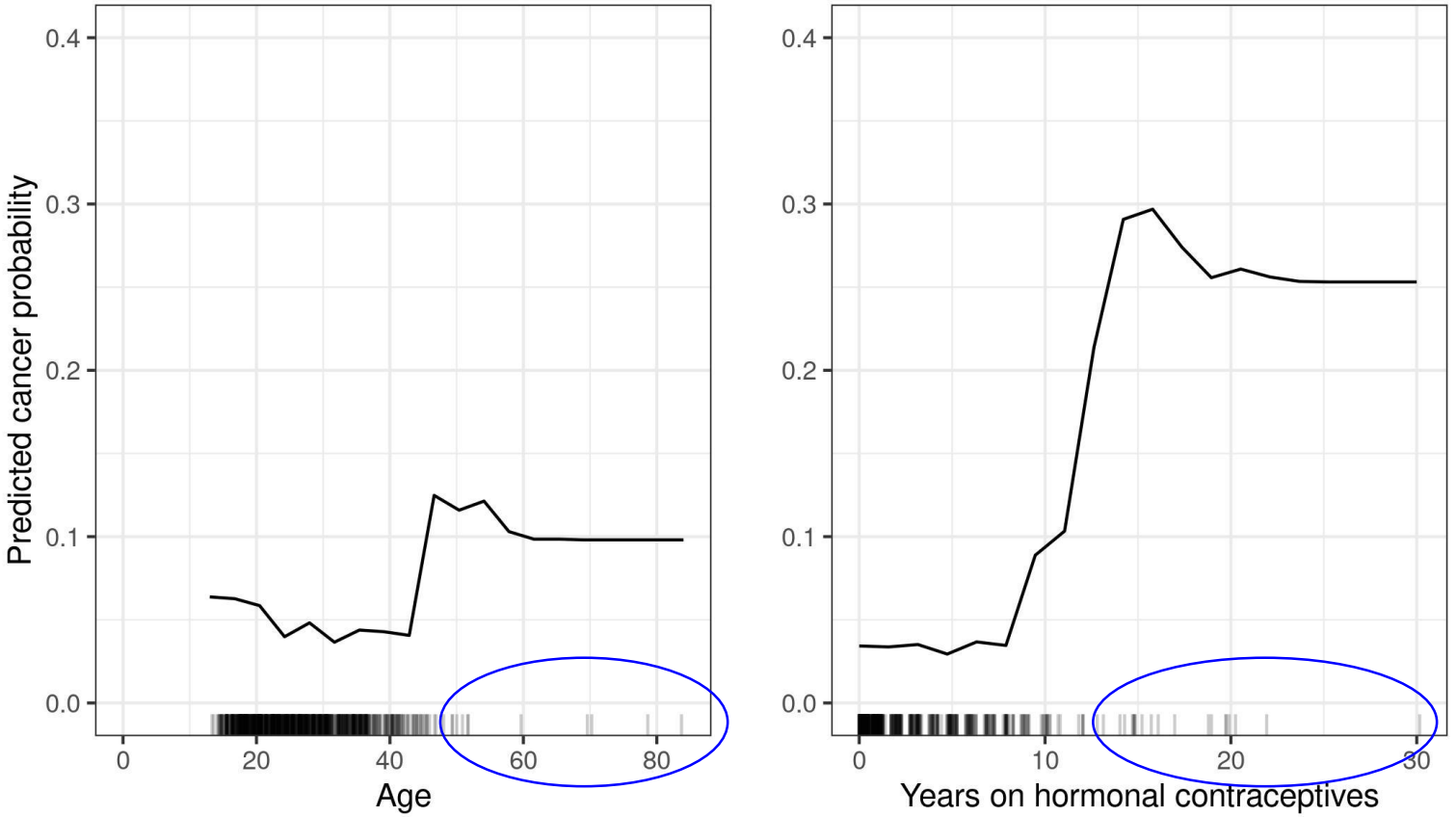
- Examples
  - Categorical feature (season)



all seasons show similar effect on the model predictions,
only for winter the model predicts fewer bicycle rentals

Partial Dependence Plot (PDP)

- Example: cancer classification
  - fit a random forest to predict whether a woman might get cervical cancer based on risk factors



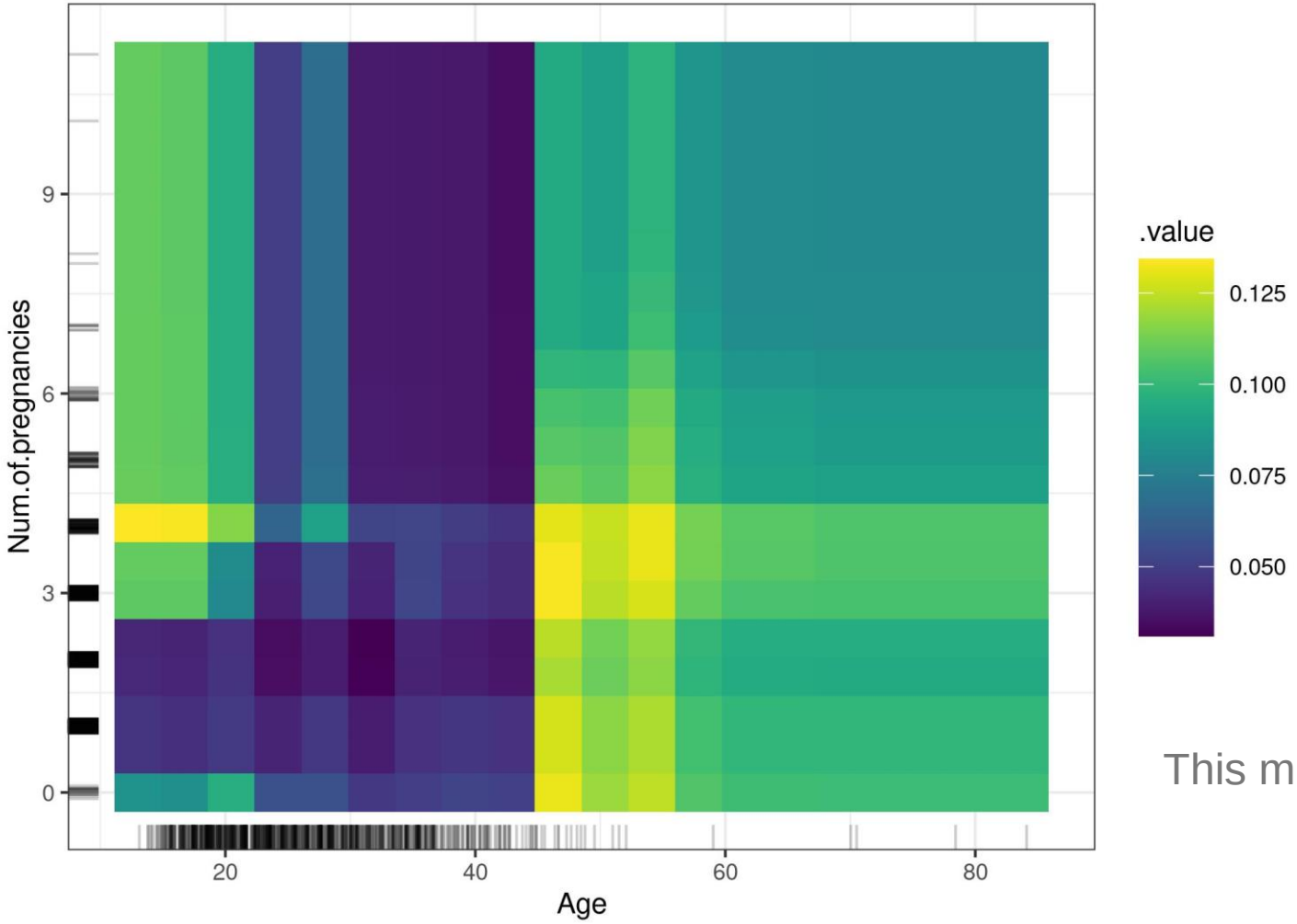
PD estimates are less reliable

the probability is low until 40 and increases after

The more years on hormonal contraceptives the higher the predicted cancer risk, especially after 10 years

Partial Dependence Plot (PDP)

- Example: cancer classification
  - the partial dependence of two features (interactions)



This might just be a correlation and not causal

Partial Dependence Plot (PDP)

- Advantages
  - The computation of partial dependence plots is **intuitive**
  - Partial dependence plots are **easy to implement**

- Disadvantages
  - The realistic **maximum number of features** in a partial dependence function is two.
  - You need to consider the feature distribution. you might overinterpret regions with almost no data.
  - The **assumption of independence :** computed are not correlated with other features
    - When the features are correlated, we create new data points in areas of the feature distribution where the actual probability is very low (for example it is unlikely that someone is 2 meters tall but weighs less than 50 kg).
    - Solution ? ALE plot !
  - **Heterogeneous effects might be hidden** because PD plots only show the average marginal effects.
    - Interactions

Accumulated Local Effects (ALE) Plot

Accumulated Local Effects (ALE) Plot

- Accumulated Local Effects (ALE) Plot
  - share the same goal with PDPs: how a feature affects the prediction on average
  - partial dependence plots have a **serious problem when the features are correlated.**
  - ALE plots are a faster and unbiased alternative

Accumulated Local Effects (ALE) Plot

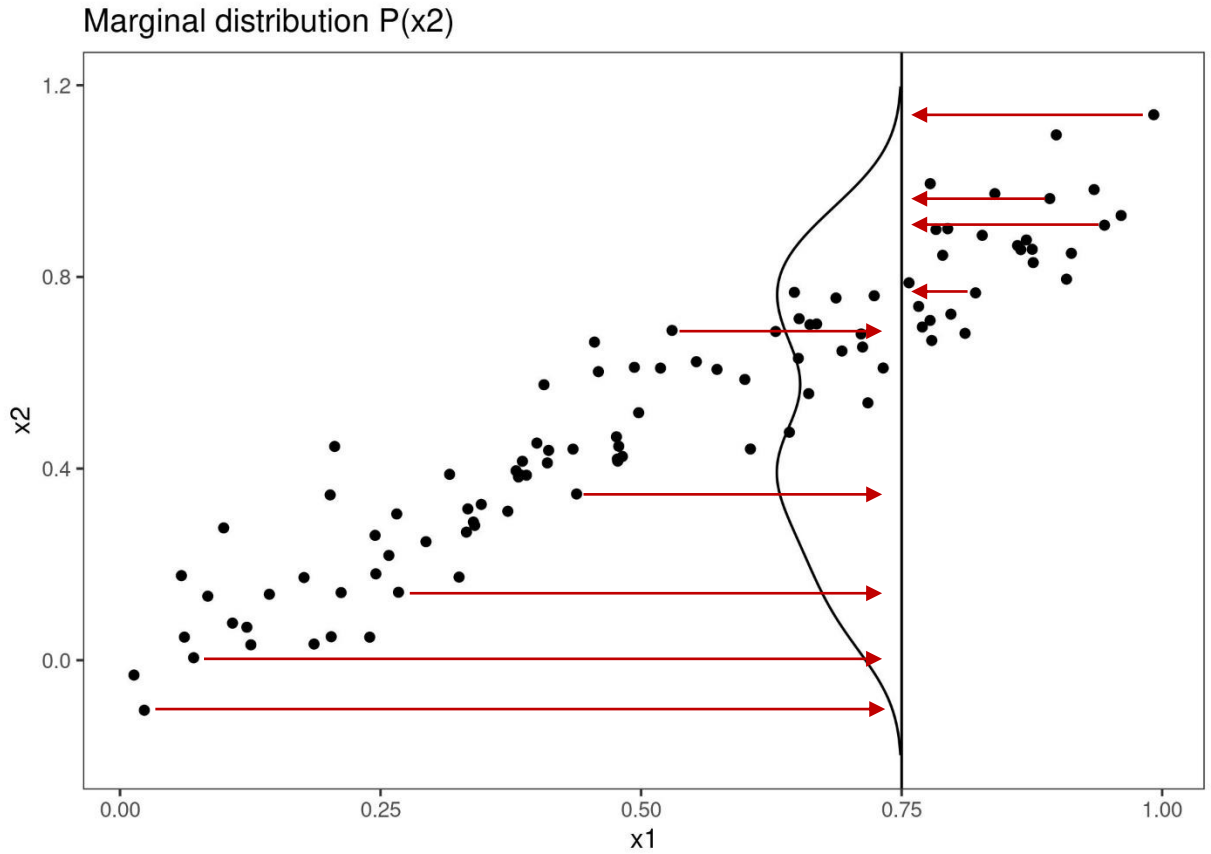
- Motivation from PDPs
  - Example: predicts **the value of a house** depending on **the number of rooms** and **the size of the living area**.

  - **Procedure PDPs**
    - 1) Select feature.
    - 2) Define grid.
    - 3) Per grid value:
      - a) Replace feature with grid value
      - b) average predictions.
    - 4) Draw curve.

  - For the calculation of the first grid value of the PDP – say 30 m^2 – we replace the living area for **all** instances by 30 m^2, even for houses with 10 rooms. → unrealistic!

Accumulated Local Effects (ALE) Plot

- Motivation and Intuition



Strongly correlated features x1 and x2. To calculate the feature effect of x1 at 0.75, the PDP replaces x1 of all instances with 0.75, falsely assuming that the distribution of x2 at x1 = 0.75 is the same as the marginal distribution of x2 (vertical line). This results in unlikely combinations of x1 and x2 (e.g. x2=0.2 at x1=0.75), which the PDP uses for the calculation of the average effect.

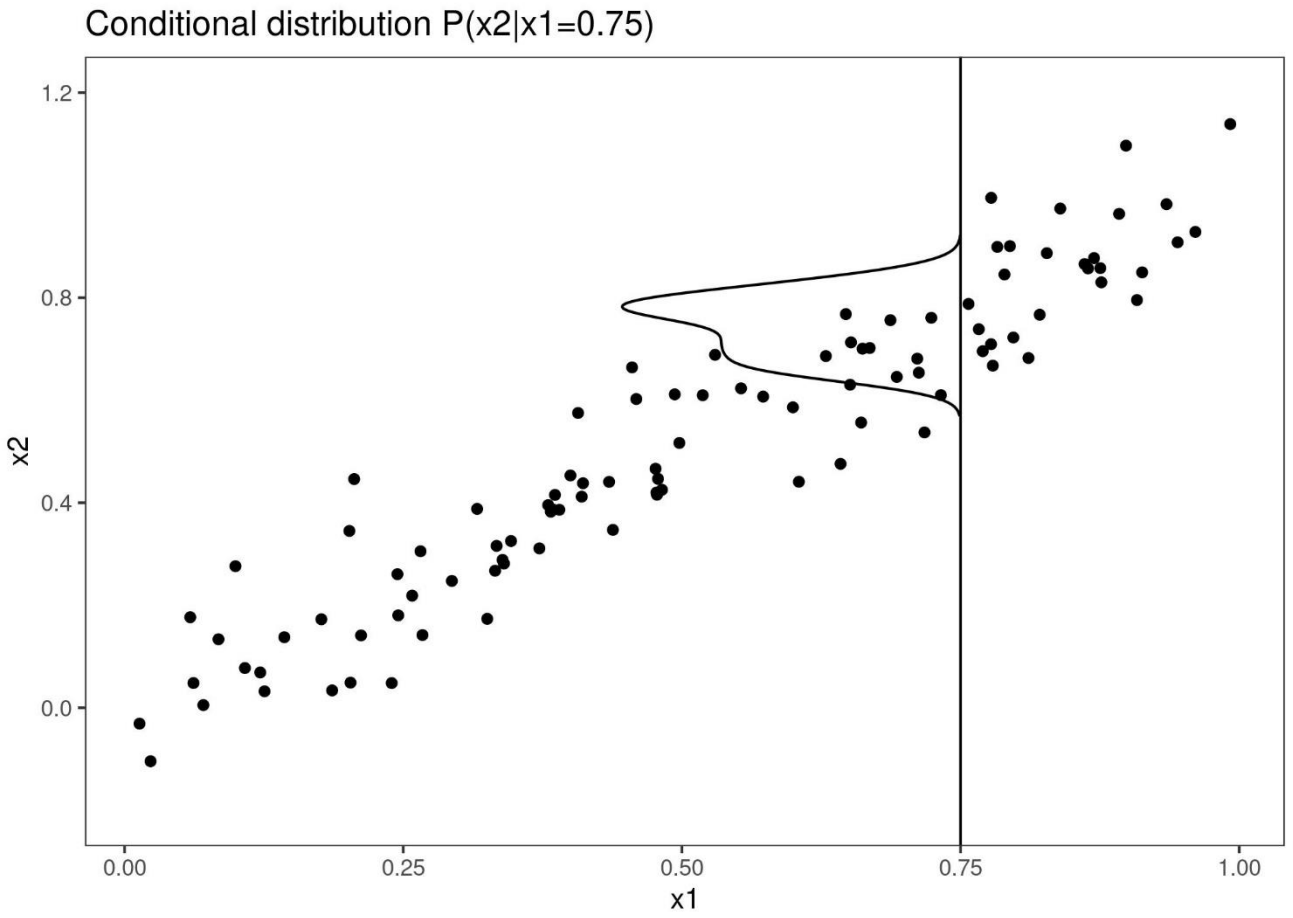
Accumulated Local Effects (ALE) Plot

- M-plots (Marginal plots)
  - Conditional distribution instead of marginal distribution
  - PDPs

$$\hat{f}_{S,PDP}(x) = E_{X_C}\left[\hat{f}(x_S, X_C)\right] = \int_{X_C} \hat{f}(x_S, X_C) d\mathbb{P}(X_C)$$

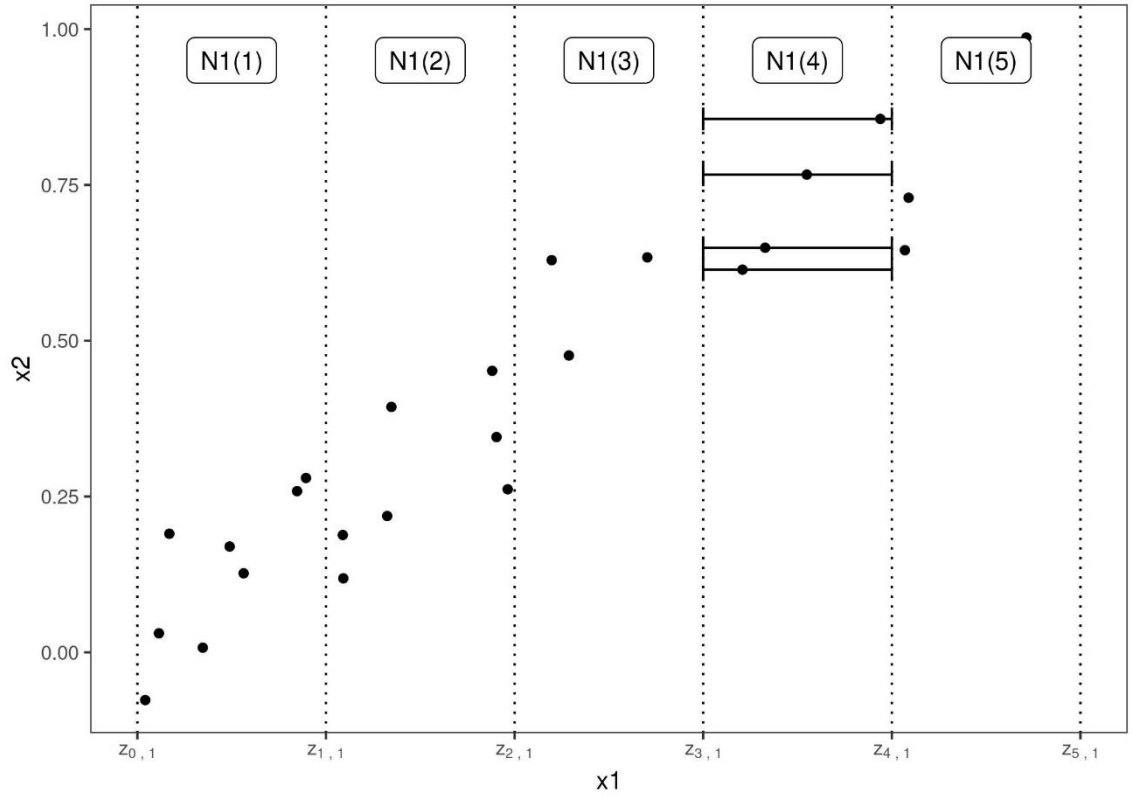
  - M-plots

$$\hat{f}_{S,M}(x_S) = E_{X_C|X_S}\left[\hat{f}(X_S, X_C)|X_S = x_s\right] = \int_{X_C} \hat{f}(x_S, X_C) d\mathbb{P}(X_C|X_S = x_S)$$



Accumulated Local Effects (ALE) Plot

- ALE (Accumulated Local Effects plots)
  - **Average the changes of predictions, not the prediction iteself**



Calculation of ALE for feature x1, which is correlated with x2. First, we divide the feature into intervals (vertical lines). For the data instances (points) in an interval, we calculate the difference in the prediction when we replace the feature with the upper and lower limit of the interval (horizontal lines). These differences are later accumulated and centered, resulting in the ALE curve.

Accumulated Local Effects (ALE) Plot

- ALE (Accumulated Local Effects plots)
  - divide the feature into many intervals and compute the differences in the predictions

Grid value z

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

Difference in prediction

Average of all instances within an interval

Accumulate the average effects across all intervals.

Effects

Local

Accumulated

This effect is centered so that the mean effect is zero.

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

The value of the ALE can be interpreted as the main effect of the feature at a certain value compared to the average prediction of the data. For example, an ALE estimate of -2 at $x_j = 3$ means that when the j-th feature has value 3, then the prediction is lower by 2 compared to the average prediction.

Accumulated Local Effects (ALE) Plot

- ALE (Accumulated Local Effects plots)

$$\hat{f}_{S,ALE}(x_S) = \int_{z_{0,S}}^{x_S} E_{X_C|X_S=x_S}\left[\hat{f}^S(X_s, X_c)|X_S = z_S\right] dz_S - \text{constant}$$

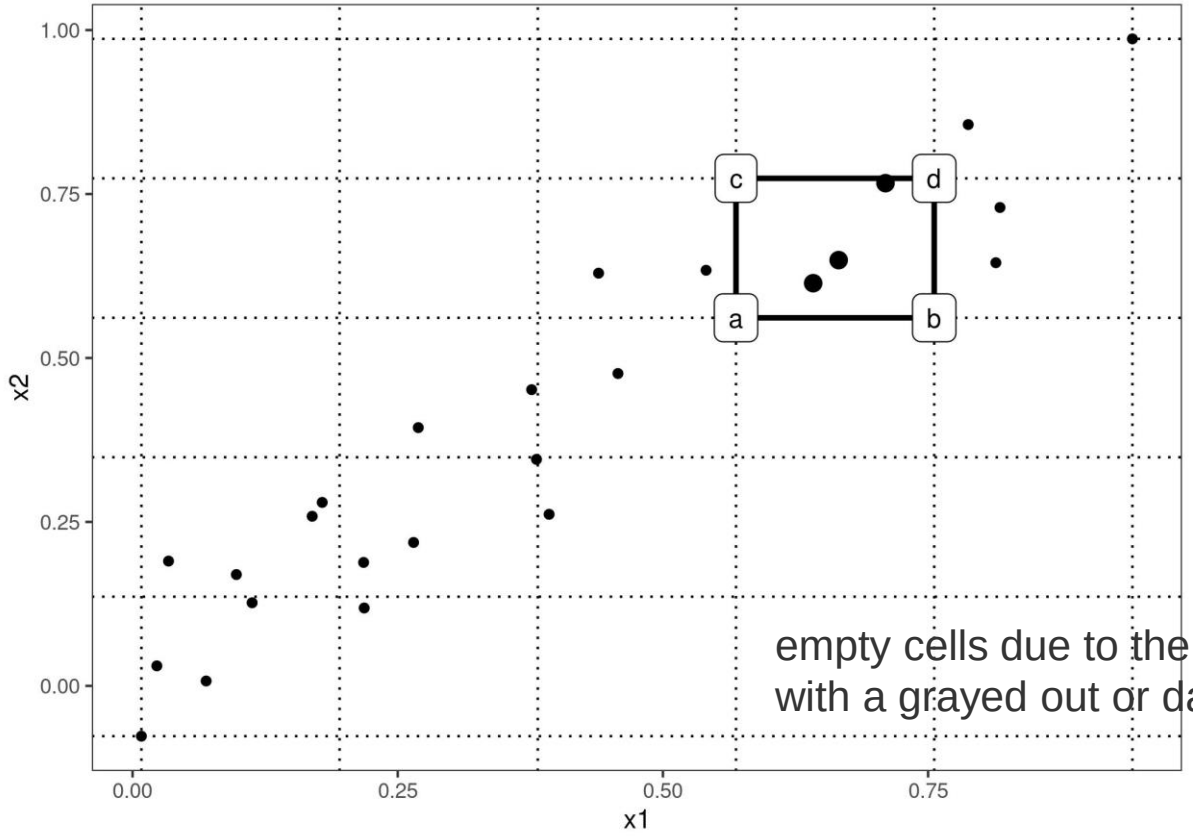
$$= \int_{z_{0,S}}^{x_S} \left(\int_{x_C} \hat{f}^S(z_s, X_c) d\mathbb{P}(X_C|X_S = z_S) d\right) dz_S - \text{constant}$$

changes in predictions

$$\hat{f}^S(x_s, x_c) = \frac{\partial \hat{f}(x_S, x_C)}{\partial x_S}$$

Accumulated Local Effects (ALE) Plot

- ALE plots for the interaction of two features
  - **only shows the additional interaction effect** of the two features, not main effect
  - The calculation principles are the same as for a single feature (rectangular cells instead of intervals)



We first replace values of x1 and x2 with the values from the cell corners. If a, b, c and d represent the "corner"-predictions of a manipulated instance, then the 2nd-order difference is (d - c) - (b - a).

empty cells due to the correlation can be visualized with a grayed out or darkened box.

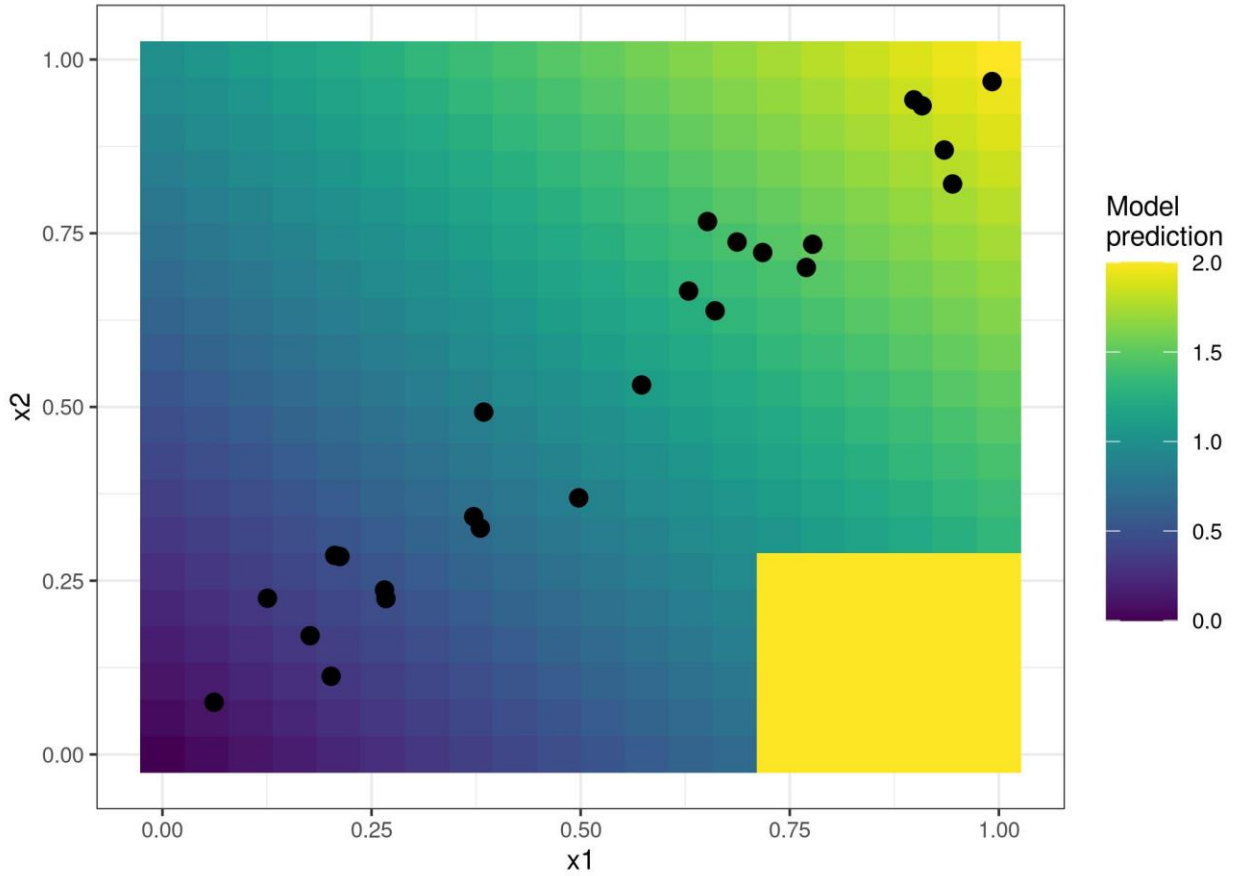
Accumulated Local Effects (ALE) Plot

- ALE plots for the interaction of two features
  - Suppose **two features do not interact**, but each has a linear effect on the predicted outcome
  - **In the 1D ALE** plot for each feature, we would see a straight line as the estimated ALE curve.
  - **In the 2D ALE** estimates, they should be close to zero, because the second-order effect is only the additional effect of the interaction.

PDPs always show the total effect,
ALE plots show the first- or second-order effect.

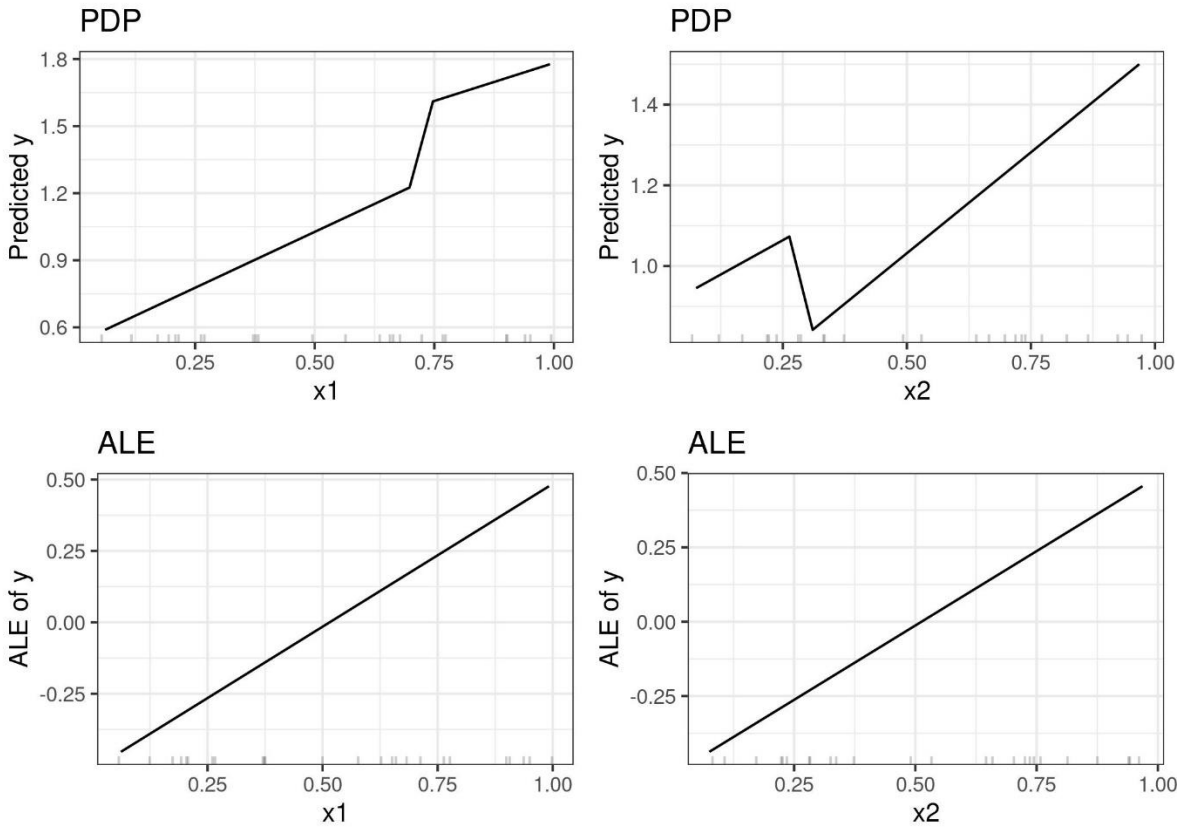
Accumulated Local Effects (ALE) Plot

- Examples: toy dataset
  - two strongly correlated features
  - a prediction model : mostly a linear regression model, but does something weird in which we have never observed instances.



Accumulated Local Effects (ALE) Plot

- Examples
  - PDP vs ALE

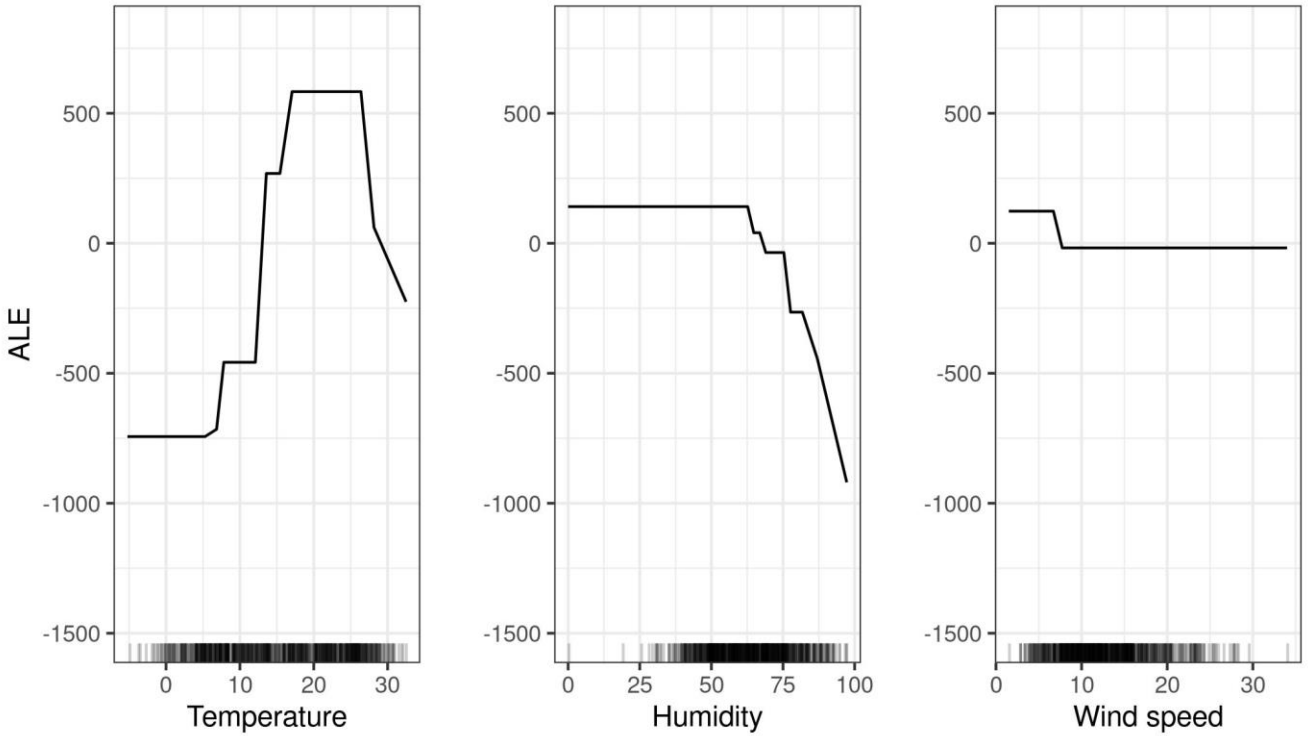


The PDP estimates are **influenced by the odd behavior** of the model outside the data distribution (steep jumps in the plots).

The ALE plots correctly identify that the machine learning model has **a linear relationship** between features and prediction, **ignoring areas without data**.

Accumulated Local Effects (ALE) Plot

- Examples: number of rented bikes problem
  - A regression tree
  - ALE plots



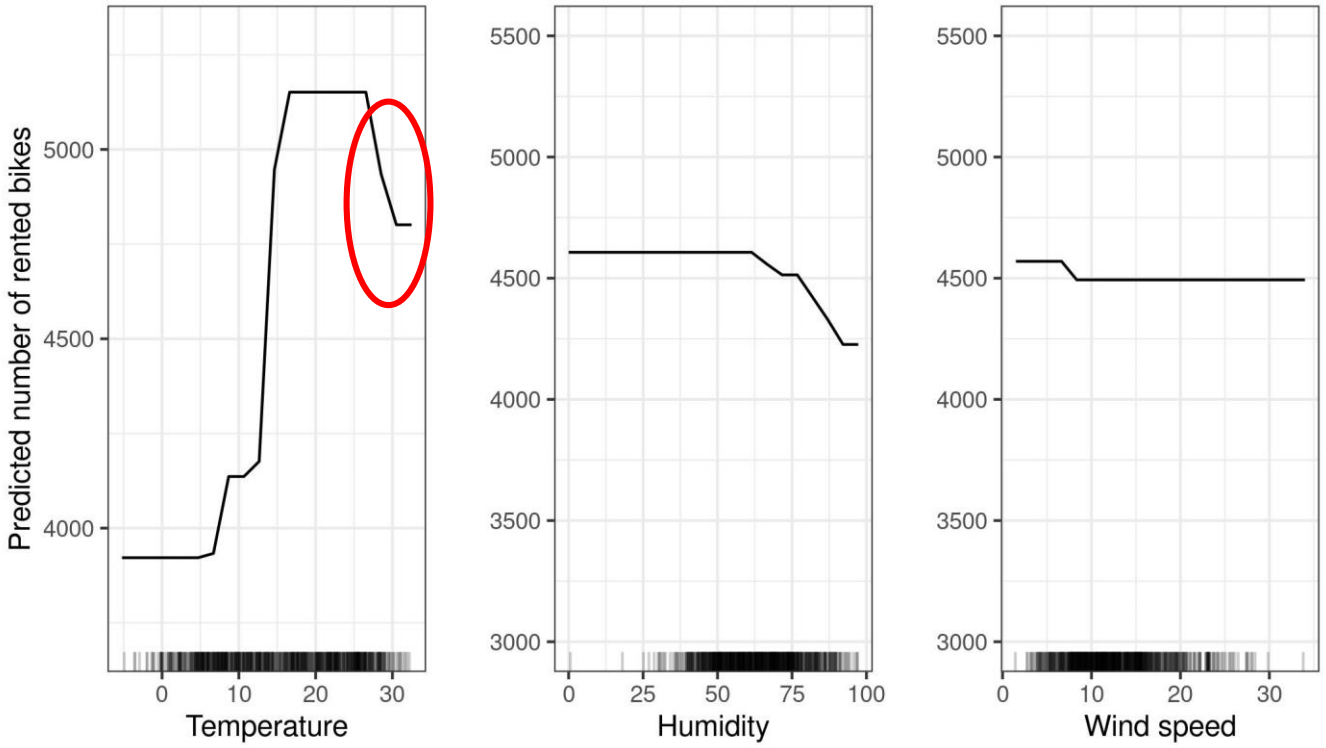
The temperature has a strong effect on the prediction. The average prediction rises with increasing temperature, but falls again above 25 degrees Celsius.

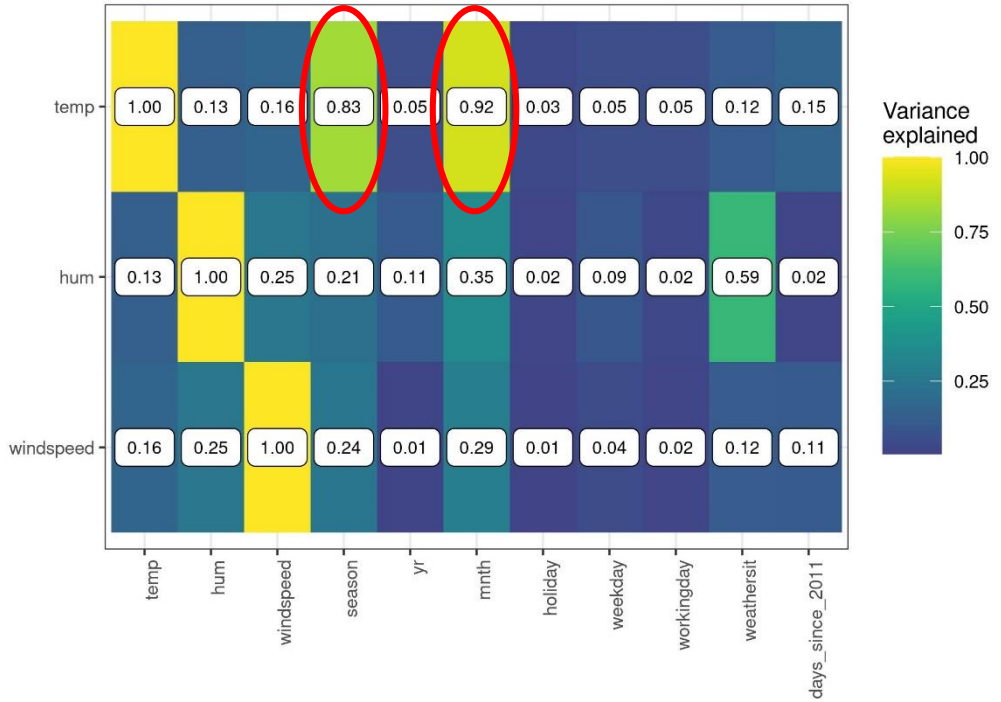
Humidity has a negative effect: When above 60%, the higher the relative humidity, the lower the prediction.

The wind speed does not affect the predictions much.

Accumulated Local Effects (ALE) Plot

- Examples: number of rented bikes problem
  - PDPs





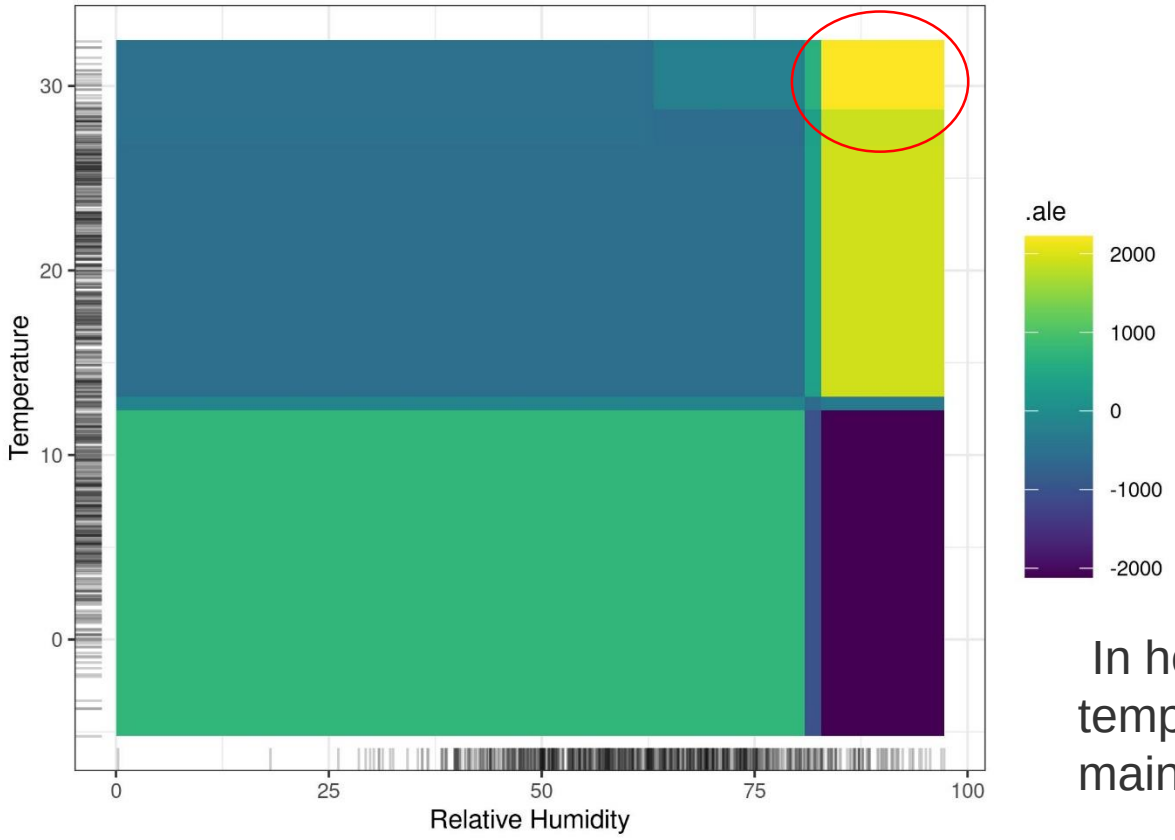
<Correlation between variables exist in this dataset>

a **smaller decrease** in predicted number of bikes for high temperature or high humidity.

The PDP uses all data instances to calculate the effect of high temperatures, even if they are, for example, instances with the season "winter".

Accumulated Local Effects (ALE) Plot

- Examples: number of rented bikes problem
  - Second-order ALE plot
  - you will not see the main effect here
  - **Only Additional interaction effect!**



interaction between temperature and humidity:
- Hot and humid weather increases the prediction.
- In cold and humid weather an additional negative effect on the number of predicted bikes is shown.

In hot and humid weather, the combined effect of temperature and humidity is therefore not the sum of the main effects, **but larger than the sum**

Accumulated Local Effects (ALE) Plot

- Advantages
  - Still work when features are correlated
  - **faster to compute** than PDPs
  - The **interpretation of ALE plots is clear**

*in most situations I would **prefer ALE plots over PDPs**, because features are usually correlated to some extent.*

- Disadvantages
  - **ALE plots can become a bit shaky**
    - the interval number is too small, the ALE plots might not be very accurate. If the number is too high, the curve can become shaky.
  - **Second-order effect plots can be a bit annoying to interpret**
    - it is only the additional effect of the interaction
  - The **implementation of ALE plots is much more complex** and less intuitive compared to partial dependence plots.

Permutation Feature Importance

Permutation Feature Importance

- A measure for the feature importance
  - the increase in the prediction error of the model after we permuted the feature's values

  - A feature is "**important**"
    - if **shuffling its values increases the model error**, because in this case the model relied on the feature for the prediction.
  - A feature is **"unimportant"**
    - if **shuffling its values leaves the model error unchanged**, because in this case the model ignored the feature for the prediction.

Permutation Feature Importance

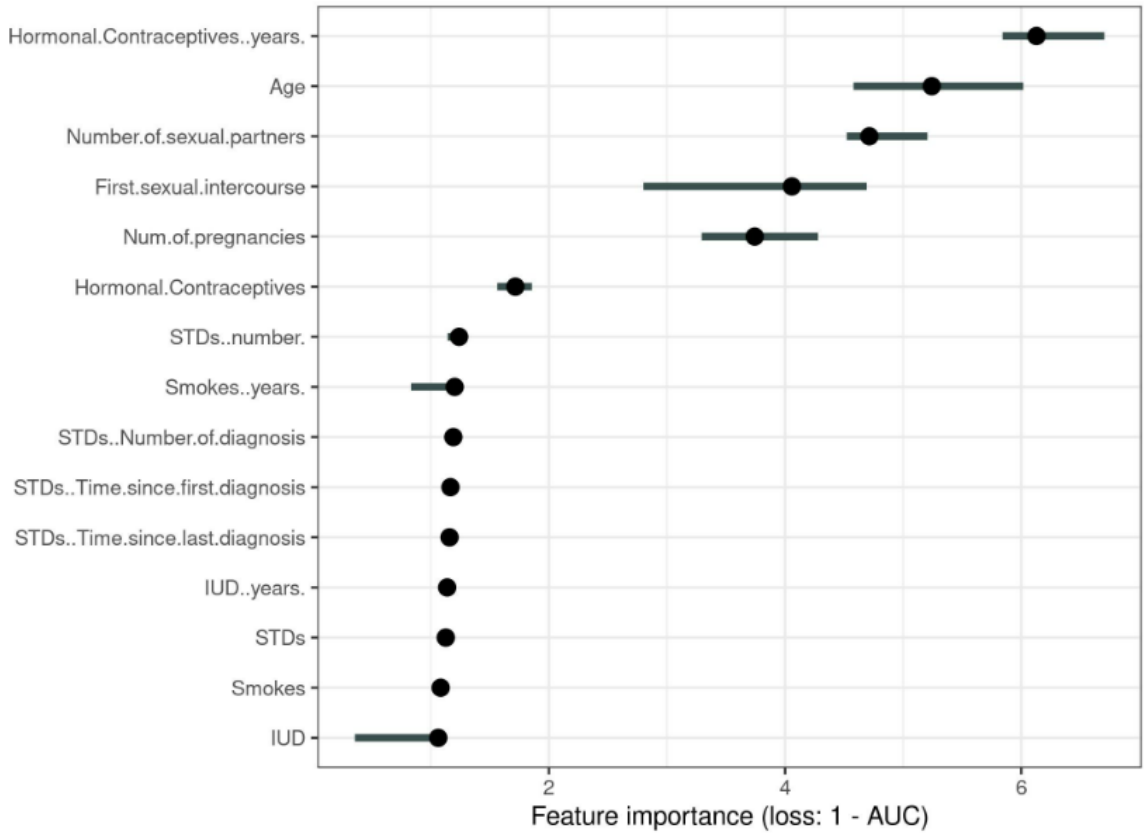
The permutation feature importance algorithm based on Fisher, Rudin, and Dominici (2018):

Input: Trained model $\hat{f}$, feature matrix X, target vector y, error measure $L(y, \hat{f})$.

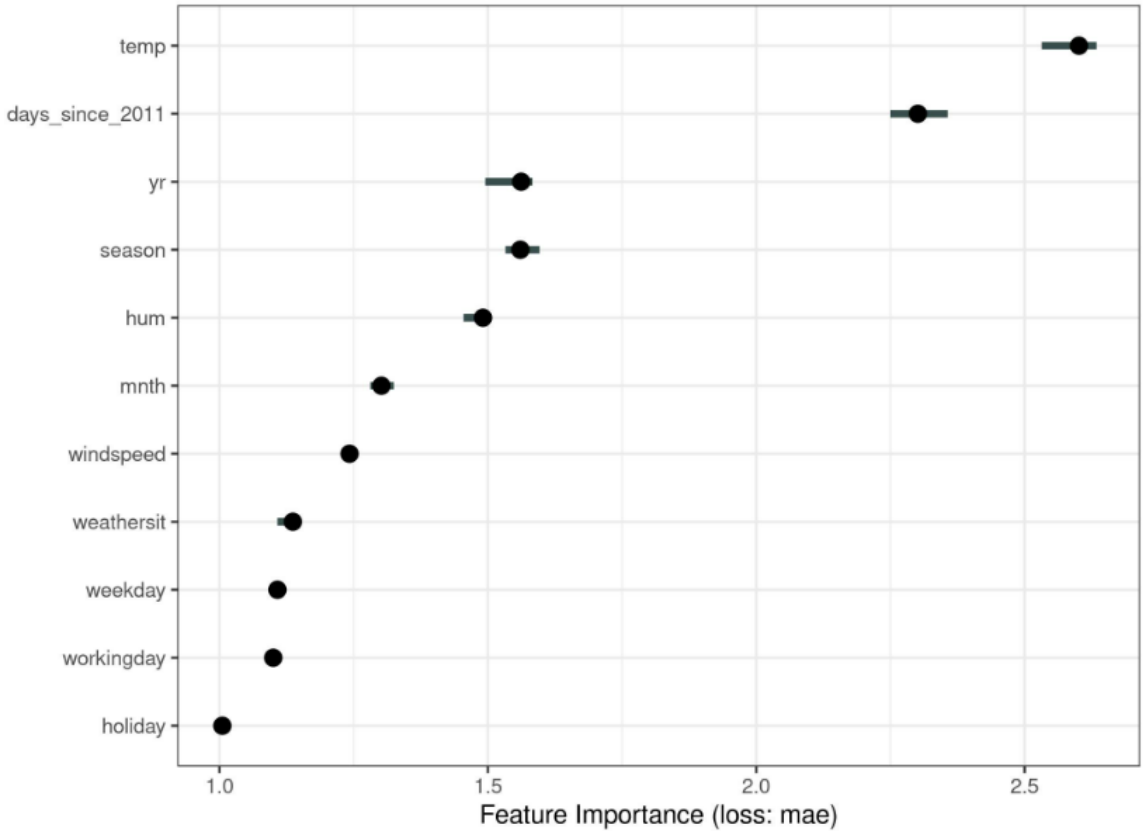
1. Estimate the original model error $e_{orig} = L(y, \hat{f}(X))$ (e.g. mean squared error)
2. For each feature $j \in \{1, \ldots, p\}$ do:
   - Generate feature matrix X_{perm} by permuting feature j in the data X. This breaks the association between feature j and true outcome y.
   - Estimate error $e_{perm} = L(Y, \hat{f}(X_{perm}))$ based on the predictions of the permuted data.
   - Calculate permutation feature importance as quotient $FI_j = e_{perm}/e_{orig}$ or difference $FI_j = e_{perm} - e_{orig}$
3. Sort features by descending FI.

Permutation Feature Importance

- Examples
  - Classification (verbical cancer)
  - Model : randomforest
  - Measure : 1-AUC



  - Regression (rented bikes)
  - Model : SVM
  - Measure : MAE



Permutation Feature Importance

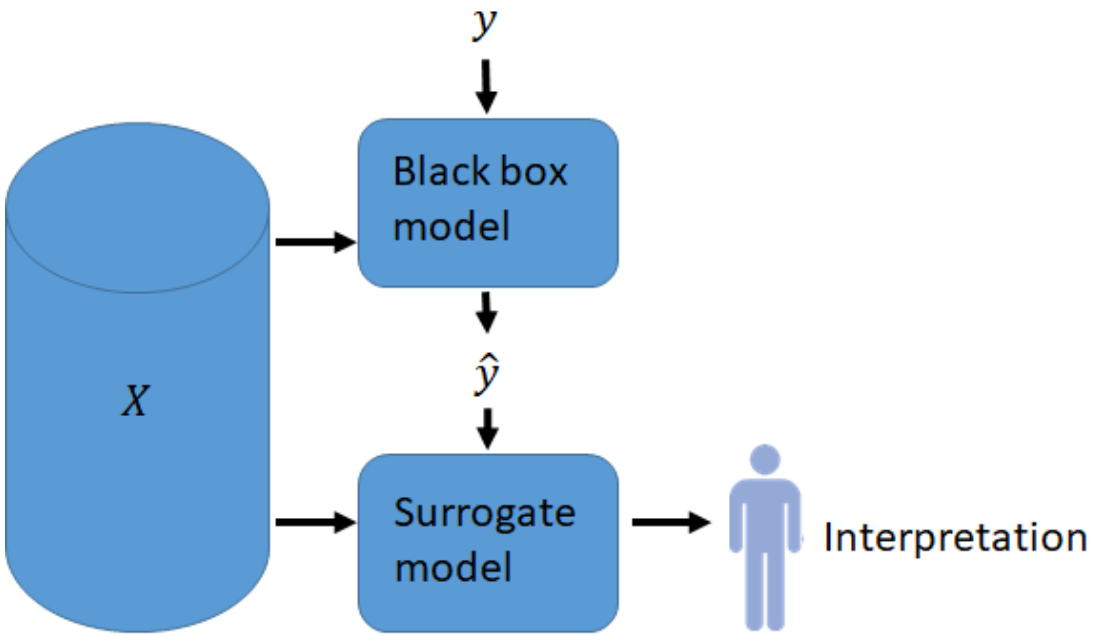
- Advantages
  - **Nice interpretation**
  - Permutation feature importance **does not require retraining the model**

- **Disadvantages**
  - You **need access to the true outcome**. If someone only provides you with the model and unlabeled data, you cannot compute the permutation feature importance.
  - When the permutation is repeated, the **results might vary greatly**
  - If features are correlated, the permutation feature importance **can be biased by unrealistic data instances**. (same as with PDPs)

Global Surrogate

Global Surrogate

- Global surrogate model
  - an interpretable model that is trained **<u>to approximate the predictions of a black box model</u>**
  - approximate our black box prediction function *f* as closely as possible with the surrogate model prediction function *g*, under the constraint that *g* is interpretable. For the function *g* any **<u>interpretable model</u>** can be used.
    - Linear model, decision tree, …



Global Surrogate

- Procedure

1. Select a dataset X. This can be the same dataset that was used for training the black box model or a new dataset from the same distribution. You could even select a subset of the data or a grid of points, depending on your application.
2. For the selected dataset X, get the predictions of the black box model.
3. Select an interpretable model type (linear model, decision tree, …).
4. Train the interpretable model on the dataset X and its predictions.
5. Congratulations! You now have a surrogate model.
6. Measure how well the surrogate model replicates the predictions of the black box model.
7. Interpret the surrogate model.

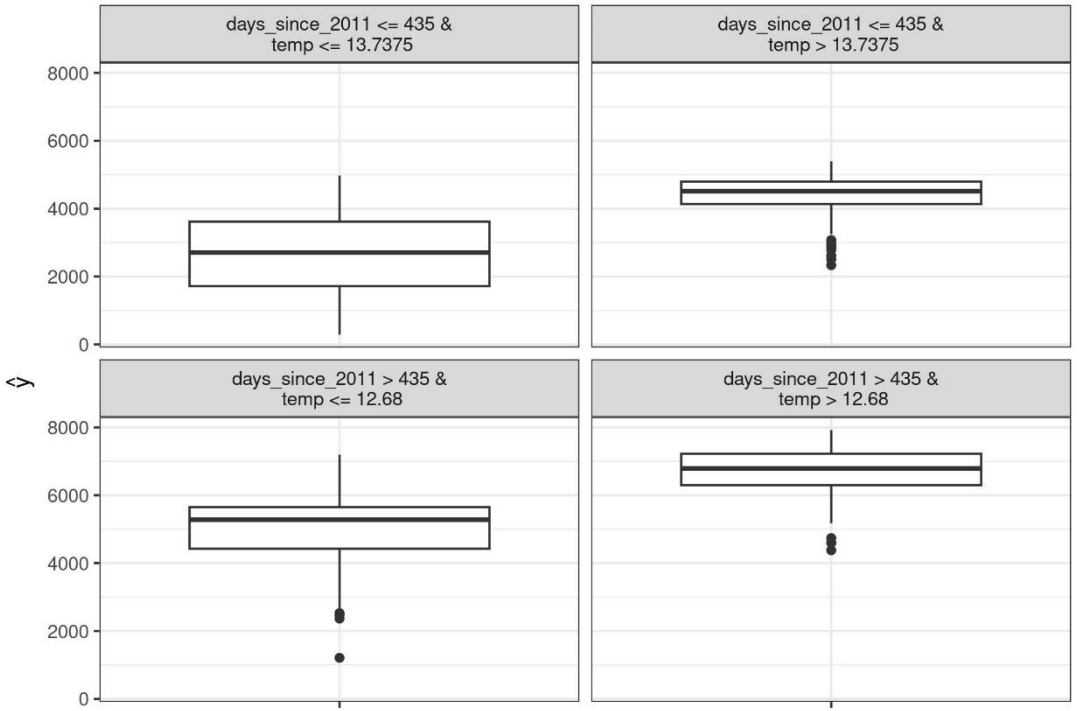
$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_*^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=1}^{n}(\hat{y}^{(i)} - \bar{\hat{y}})^2}$$

the percentage of variance that is captured by the surrogate model.
1: approximate very well
0: fails to explain the black box model

Global Surrogate

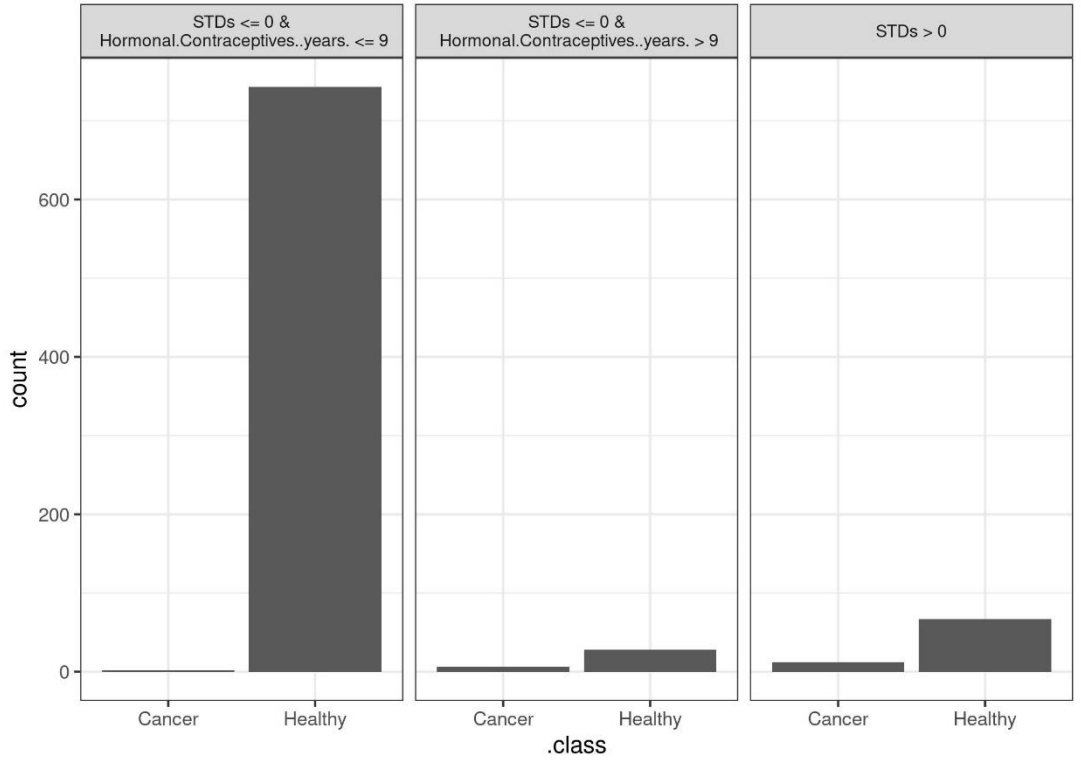
- Example
  - Rented bike problem
  - Original model: SVM
  - Surrogate model: Decision tree
  - R squared: 0.77

If the fit were perfect, we could throw away the support vector machine and use the tree instead.



  - Cervical cancer problem
  - Original model: random forest
  - Surrogate model: Decision tree
  - R squared: 0.19

we should not overinterpret the tree when drawing conclusions about the complex model.



Global Surrogate

- Advantages
  - The surrogate model method is **flexible**: Any interpretable models from can be used.
  - the approach is very **intuitive** and straightforward

- Disadvantages
  - Conclusions are about the model, not about the data (or real world)
  - It is not clear what the best **cut-off for R-squared** is in order to be confident

Prototypes and Criticisms

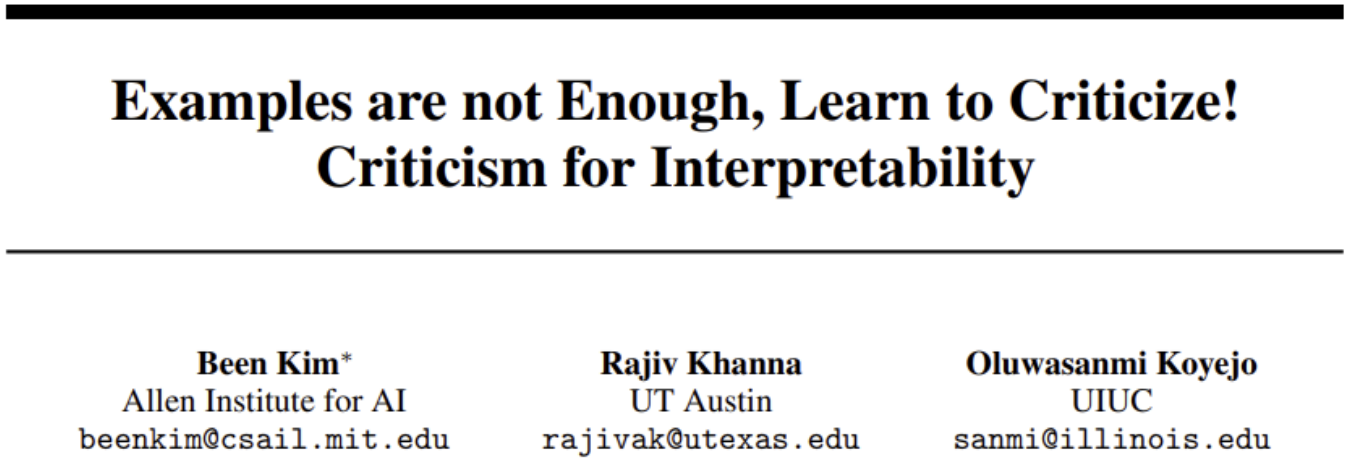
Example-based Explanations

- Example-based explanations only make sense if we can represent an instance of the data in a humanly understandable way.
  - This **works well for images**, because we can view them directly.
  - It is more challenging to represent tabular data in a meaningful way, because an instance can consist of hundreds or thousands of features.

- Thing B is similar to thing A and A caused Y, so I predict that B will cause Y as well.

- The **k-nearest neighbors (kNN)** method works explicitly with example-based predictions
  - The prediction of a knn can be explained by returning the k neighbors
    - only meaningful if we have a good way to represent a single instance.

Prototypes and Criticisms



Examples are not Enough, Learn to Criticize!
Criticism for Interpretability

Been Kim*
Allen Institute for AI
beenkim@csail.mit.edu

Rajiv Khanna
UT Austin
rajivak@utexas.edu

Oluwasanmi Koyejo
UIUC
sanmi@illinois.edu

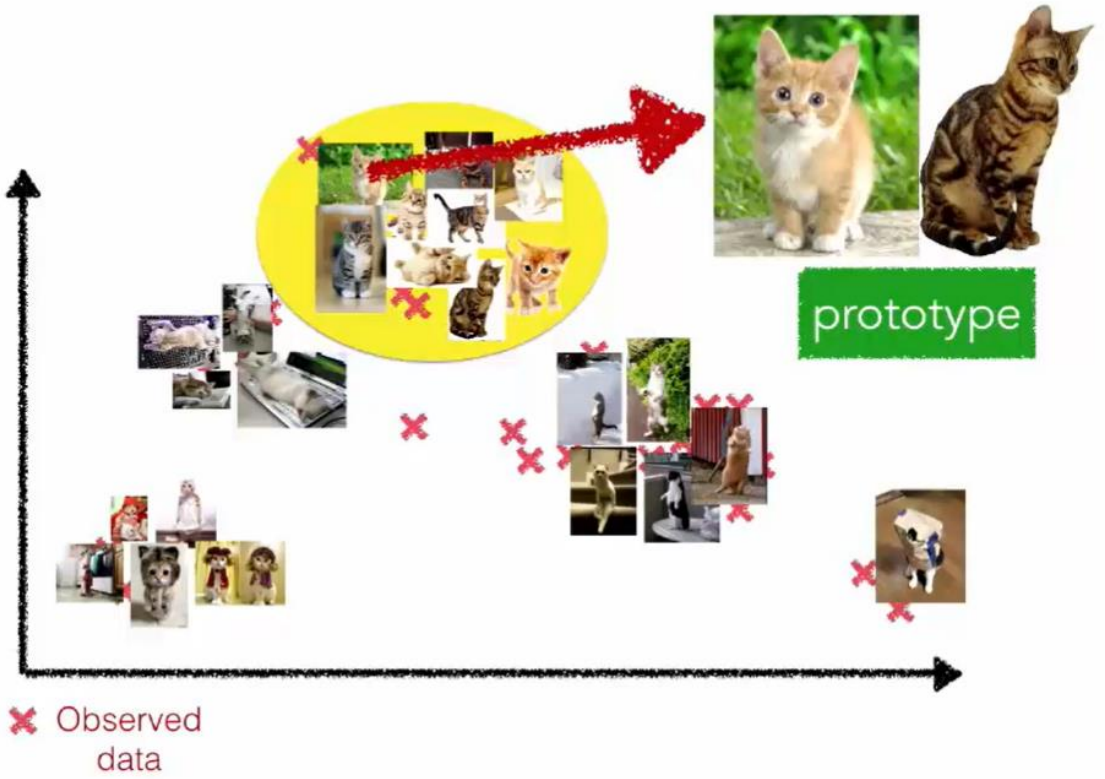
Kim, Been, Rajiv Khanna, and Oluwasanmi O. Koyejo. "Examples are not enough, learn to criticize! criticism for interpretability." *Advances in neural information processing systems* 29 (2016).

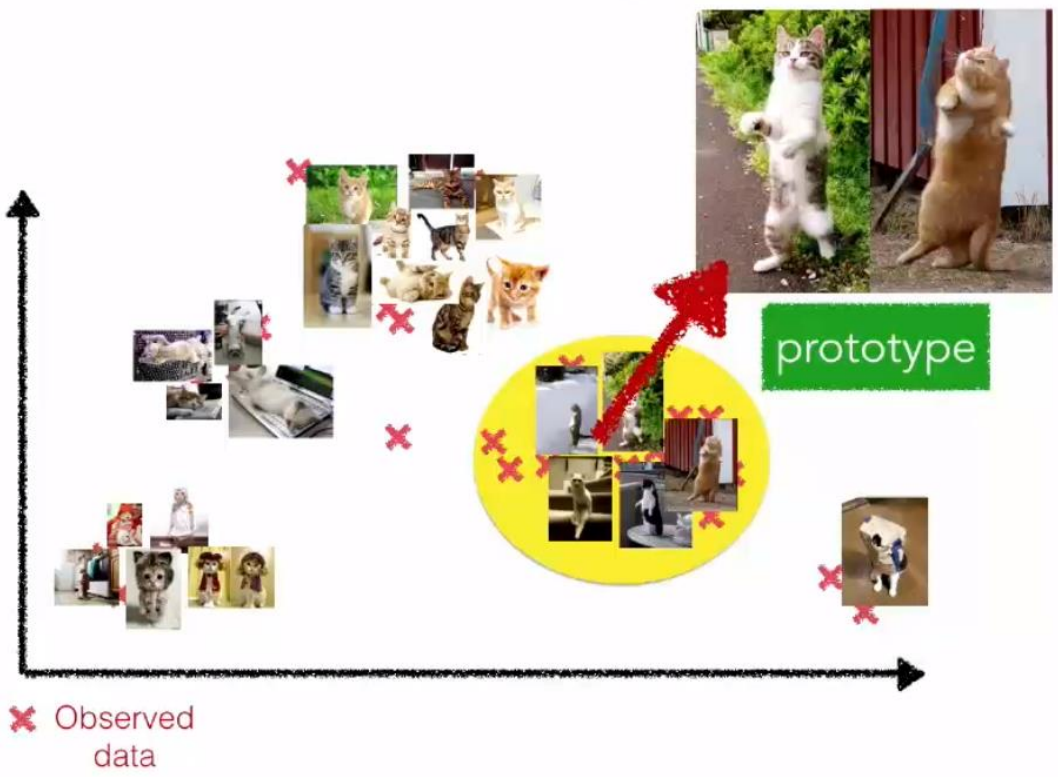
Author's presentation:
https://learn.microsoft.com/en-us/events/neural-information-processing-systems-conference-nips-2016/examples-are-not-enough-learn-to-criticize-criticism-interpretability

Prototypes and Criticisms

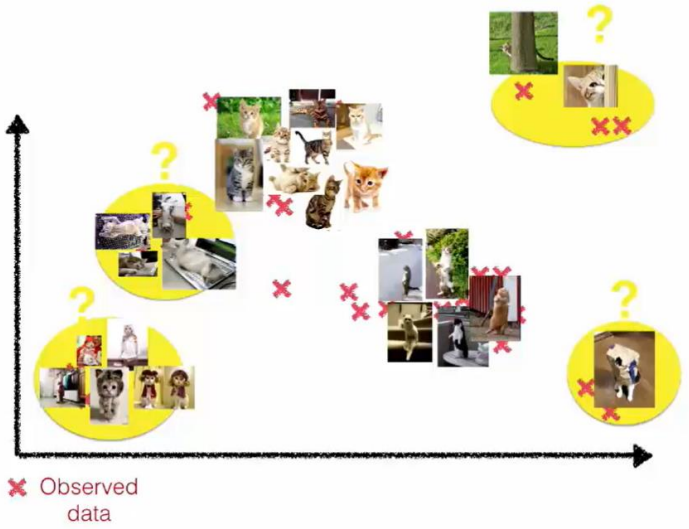
- Understanding data through examples

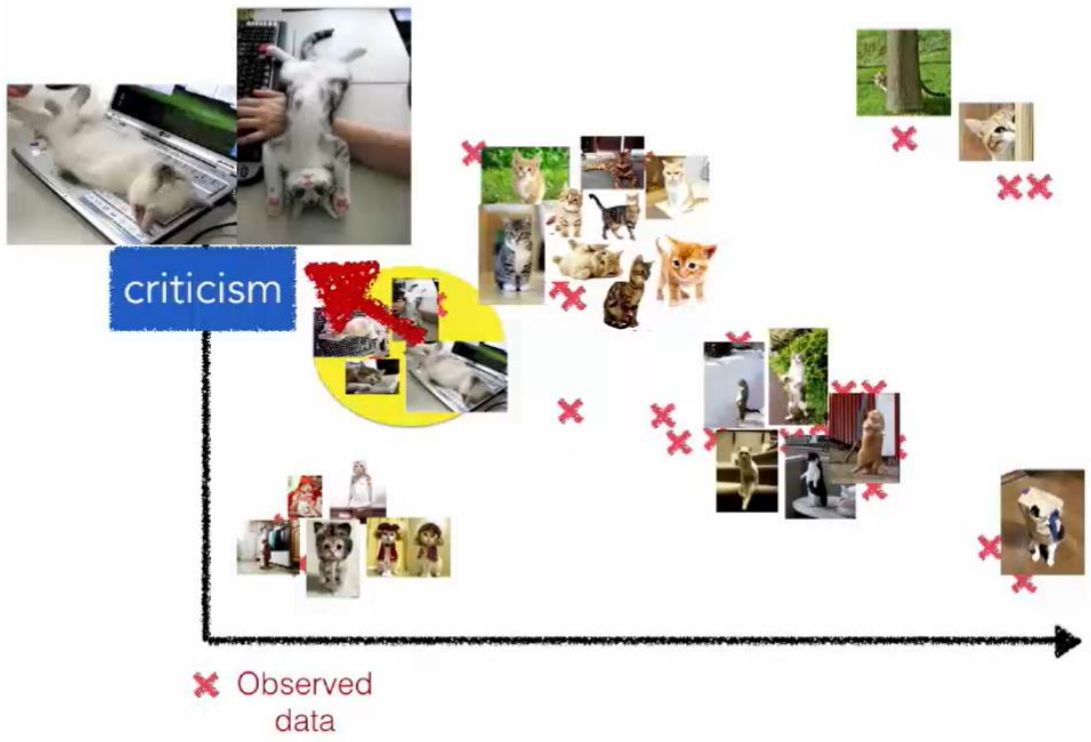
Prototypes and Criticisms

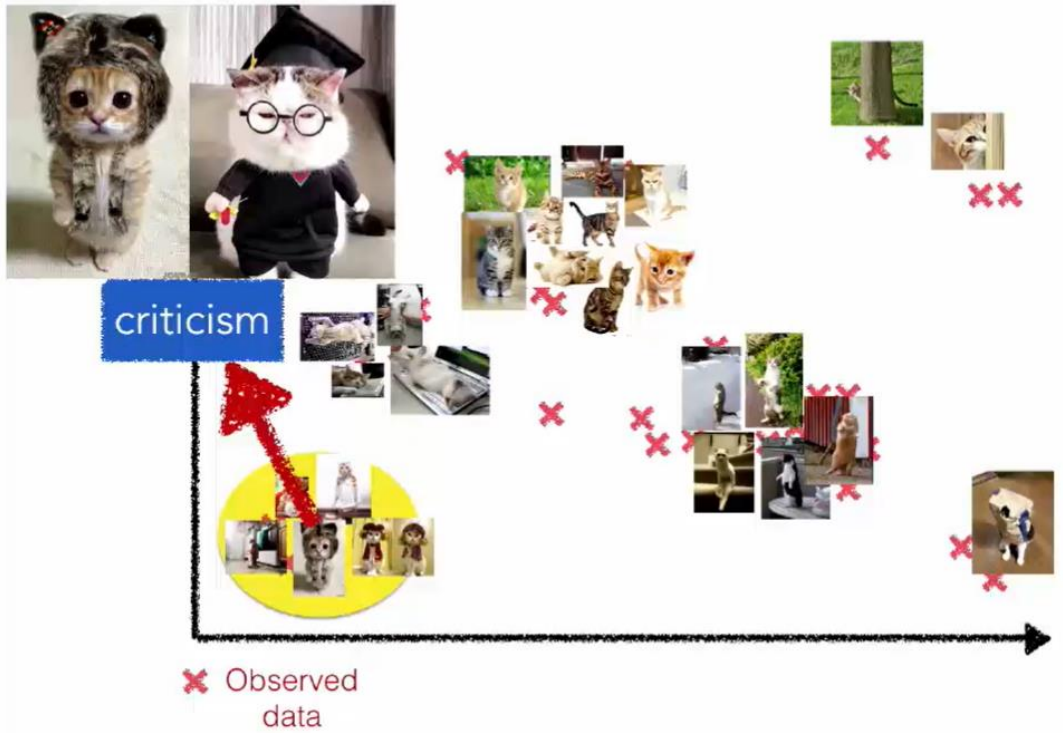




Prototypes and Criticisms

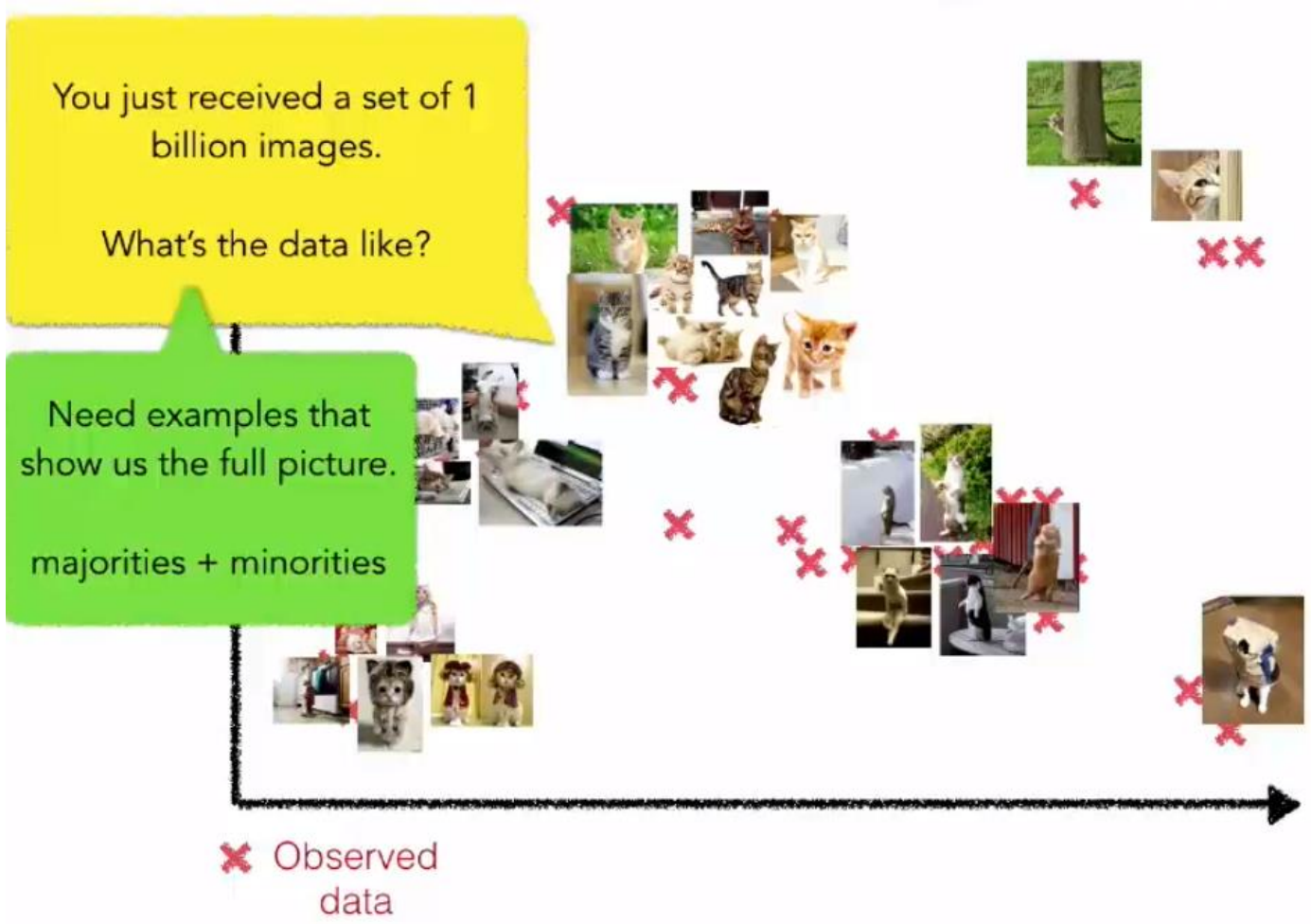






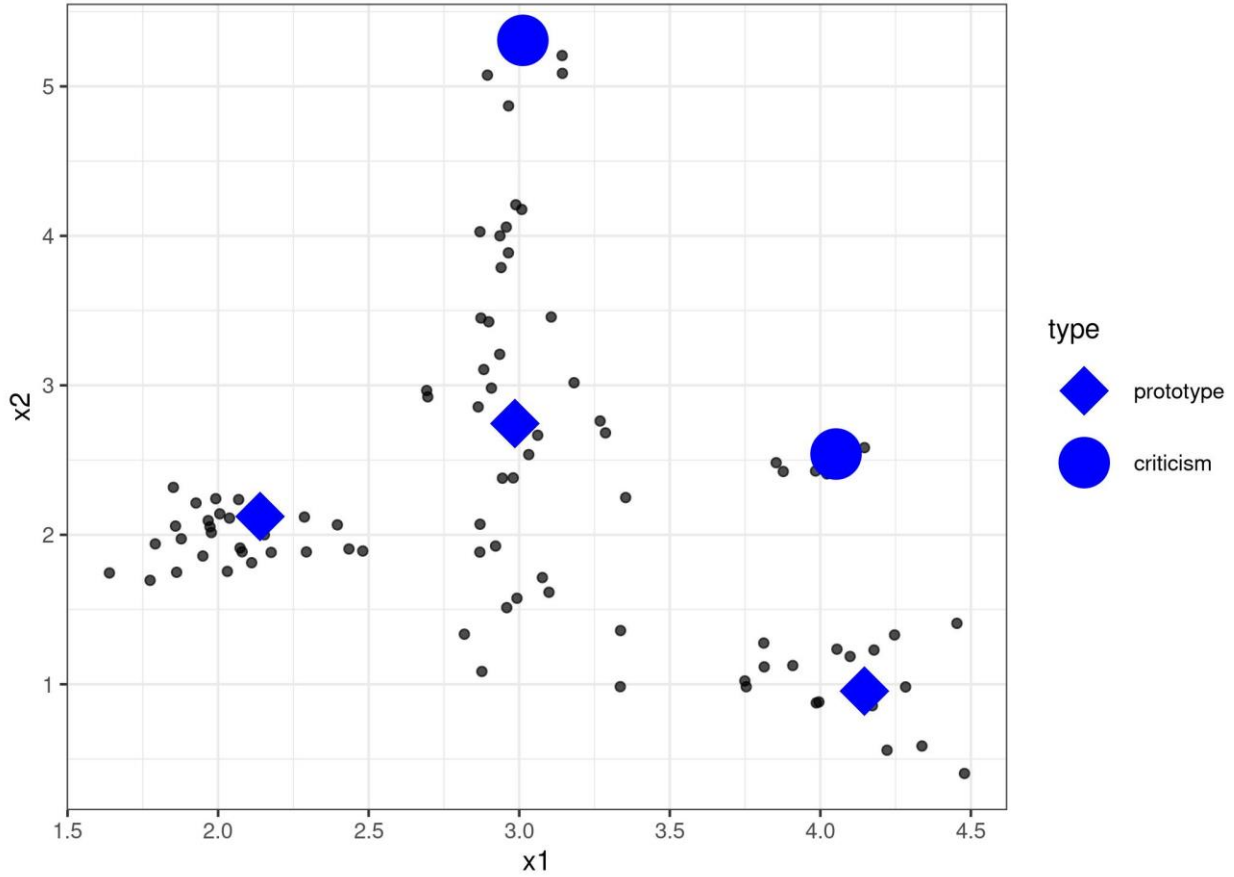
Prototypes and Criticisms

- **MMD-critic**



Prototypes and Criticisms

- 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.
  - The purpose of criticisms is to provide insights together with prototypes, especially for data points which the prototypes do not represent well.



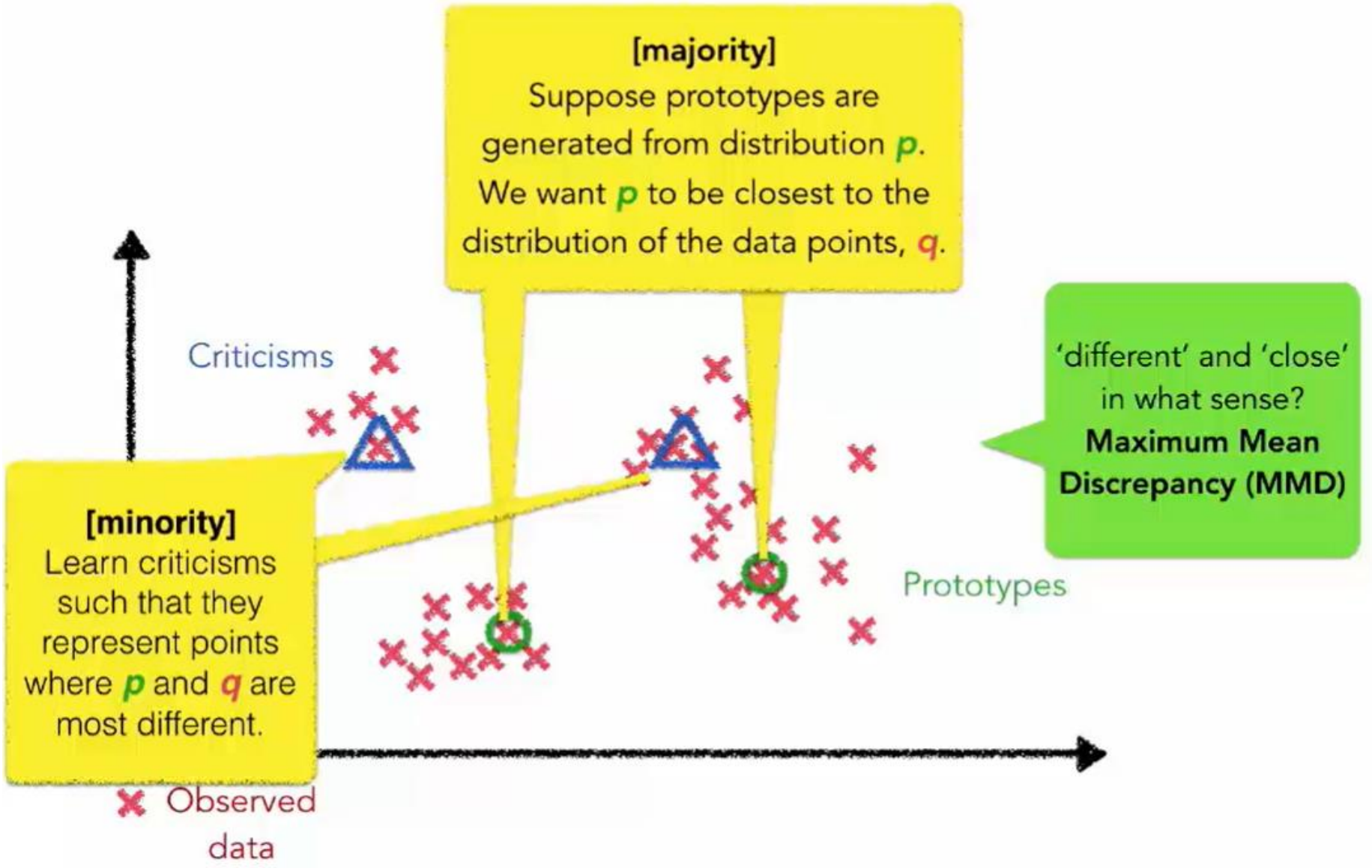
Prototypes and Criticisms

- MMD-critic
  - selects **prototypes** that minimize the discrepancy between the two distributions.
    - the distribution of the data vs the distribution of the selected prototype
  - Data points from regions that are not well explained by the prototypes are selected as **criticisms.**

- The MMD-critic procedure
  1. Select the number of prototypes and criticisms.
  2. **Find prototypes** with greedy search.
     - Prototypes are selected so that the distribution of the prototypes is close to the data distribution.
  3. **Find criticisms** with greedy search.
     - Points are selected as criticisms where the distribution of prototypes differs from the distribution of the data.

Prototypes and Criticisms

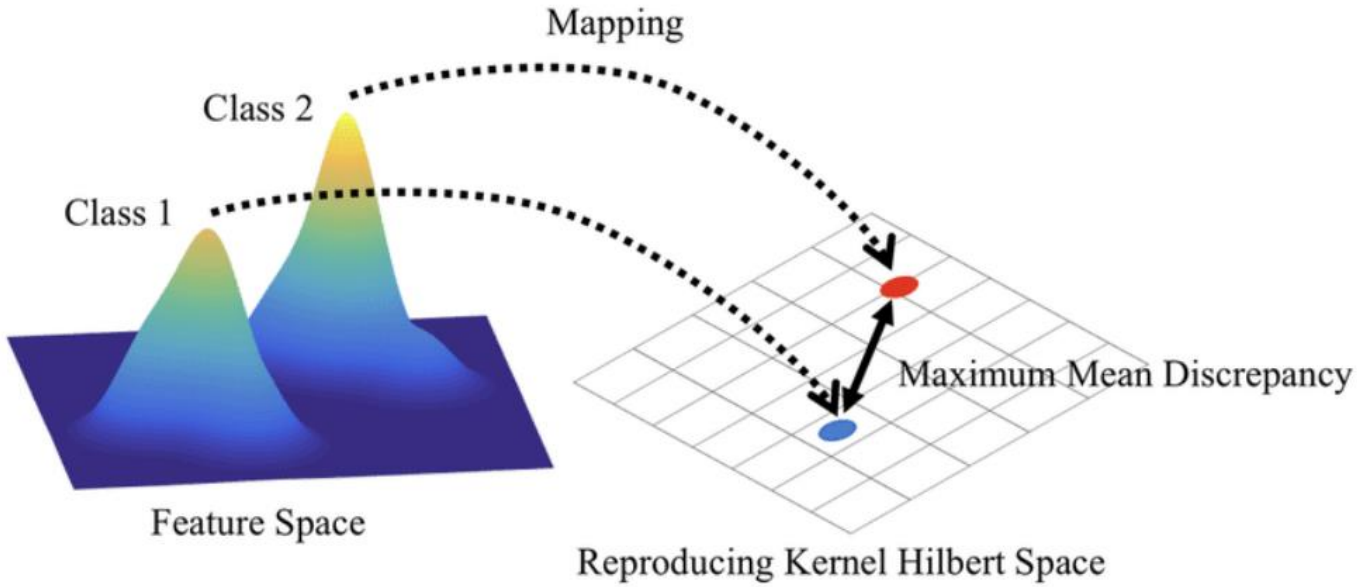
- MMD-critic



Prototypes and Criticisms

https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html

- Maximum Mean Discrepancy (MMD)
  - statistical measure used to determine **the difference between two probability distributions**



Radial basis function (RBF) kernel : (Gaussian kernel)

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

$$\varphi(x) = \begin{bmatrix} \varphi_1(x) \\ \varphi_2(x) \\ \varphi_3(x) \\ \vdots \end{bmatrix}$$

Kernel Trick

Firstly given an X, a feature map ϕ maps X to an another space $\mathcal{F}$ such that $\phi(X) \in \mathcal{F}$. Assuming $\mathcal{F}$ satisfies the necessary conditions, we can benefit from the kernel trick to compute the inner product in $\mathcal{F}$:

$$X, Y \text{ such that } k(X, Y) = \langle \phi(X), \phi(Y) \rangle_{\mathcal{F}}$$

Prototypes and Criticisms

https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html

- Maximum Mean Discrepancy (MMD)

1. Feature means: Given a probability measure P on $\mathcal{X}$, feature means (or mean embedding as sometimes called in the literature) is an another feature map that takes $\phi(X)$ and maps it to the means of every coordinate of $\phi(X)$:

$$\mu_p\left(\phi(X)\right) = [E[\phi(X_1], \cdots, E[\phi(X_m]]^T \tag{1}$$

MMD is a distance (difference) between feature means.

Inner product of feature means of $X \sim P$ and $Y \sim Q$ can be written in terms of kernel function such that:

$$\langle \mu_P\left(\phi(X), \mu_Q\left(\phi(Y)\right\rangle_{\mathcal{F}} = E_{P,Q}\left[\langle \phi(X), \phi(Y)\rangle_{\mathcal{F}}\right] = E_{P,Q}\left[k(X,Y)\right] \tag{2}$$

Prototypes and Criticisms

https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html

- Maximum Mean Discrepancy (MMD)

MMD is a distance (difference) between feature means.

2. Maximum mean discrepancy: Given X, Y maximum mean discrepancy is the distance between feature means of X, Y:

$$MMD^2(P,Q) = \|\mu_P - \mu_Q\|^2_{\mathcal{F}} \tag{3}$$

For convenience we have left out the $\phi(\cdot)$ parts. If we use the norm induced by the inner product such that $\|x\| = \sqrt{\langle x, x \rangle}$, the equation (3) becomes

$$MMD^2(P,Q) = \langle \mu_P - \mu_Q,\ \mu_P - \mu_Q \rangle = \langle \mu_P, \mu_P \rangle - 2\langle \mu_P, \mu_Q \rangle + \langle \mu_Q, \mu_Q \rangle$$

Using the equation (2), finally above expression becomes

$$MMD^2(P,Q) = E_P\left[k(X,X)\right] - 2E_{P,Q}\left[k(X,Y)\right] + E_Q\left[k(Y,Y)\right] \tag{4}$$

Prototypes and Criticisms

https://www.onurtunali.com/ml/2019/03/08/maximum-mean-discrepancy-in-machine-learning.html

- Maximum Mean Discrepancy (MMD)

3. Empirical estimation of MMD: Even though we are working with distributions so far, in real life settings we don't have access to the underlying distribution of our data. For this reason, it is possible to use an estimate for the equation (4) with following formula:

$$MMD^2(X,Y) = \underbrace{\frac{1}{m(m-1)}\sum_i\sum_{j\neq i} k(\mathbf{x_i},\mathbf{x_j})}_{\text{A}} - \underbrace{2\frac{1}{m.m}\sum_i\sum_j k(\mathbf{x_i},\mathbf{y_j})}_{\text{B}} + \underbrace{\frac{1}{m(m-1)}\sum_i\sum_{j\neq i} k(\mathbf{y_i},\mathbf{y_j})}_{\text{C}} \quad (5)$$

Prototypes and Criticisms

- Maximum Mean Discrepancy (MMD)

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

m: the number of prototypes z
n: is the number of data points x
The prototypes z are a selection of data points x

First term: the average proximity of the prototypes to each other

Second term: average proximity between the prototypes and all other data points

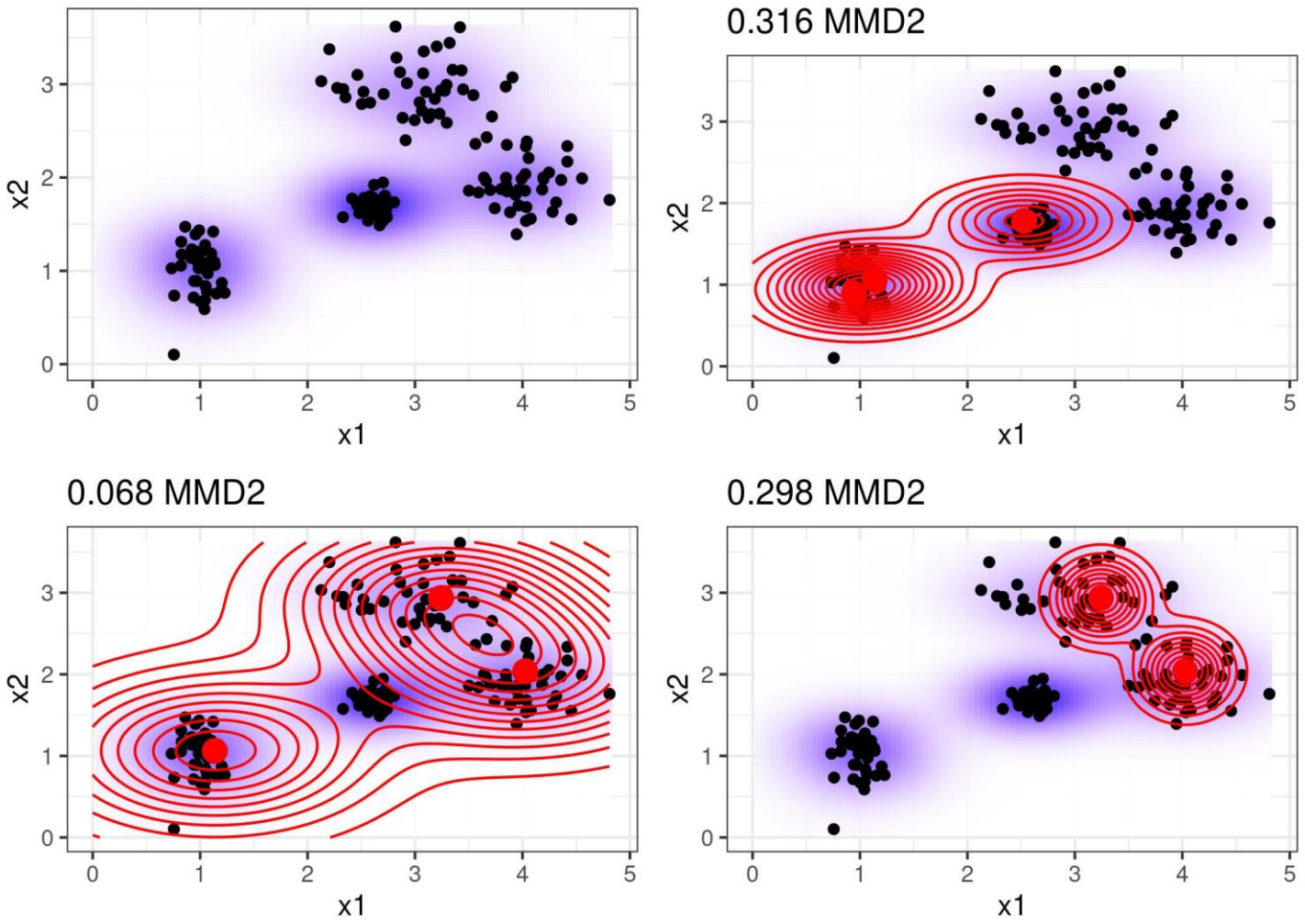
Last term: the average proximity of the data points to each other

what happen if you used all n data points as prototypes ?

The closer MMD2 is to zero, the better the distribution of the prototypes fits the data.

Prototypes and Criticisms

- Maximum Mean Discrepancy (MMD)
  - example



Prototypes and Criticisms

- Maximum Mean Discrepancy (MMD)
  - Kernel

Radial basis function (RBF) kernel :
(Gaussian kernel)

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

γ : scaling parameter

  - decreases with the distance between the two points
  - ranges between zero and one
    - Zero when the two points are infinitely far apart;
    - one when the two points are equal.

Prototypes and Criticisms

- **Finding Prototypes**
  - that can minimize overall MMD
  - Using Greedy search

- Start with an empty list of prototypes.
- While the number of prototypes is below the chosen number m:
  - For each point in the dataset, check how much MMD2 is reduced when the point is added to the list of prototypes. Add the data point that minimizes the MMD2 to the list.
- Return the list of prototypes.

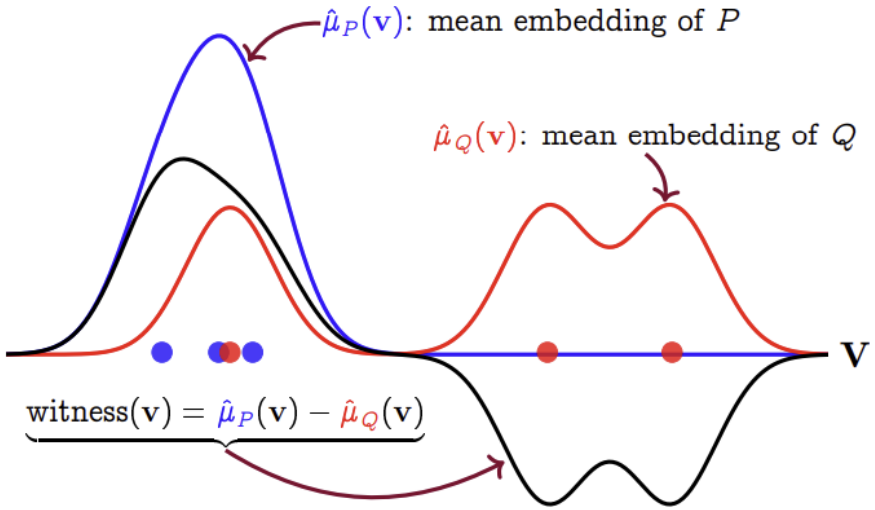
Prototypes and Criticisms

- Witness function
  - 두 분포의 (Prototype 분포와 data 분포) density 차이에 대한 함수

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

data point의 distribution에 fit하는 정도 prototype의 distribution에 fit하는 정도

Conceptually...



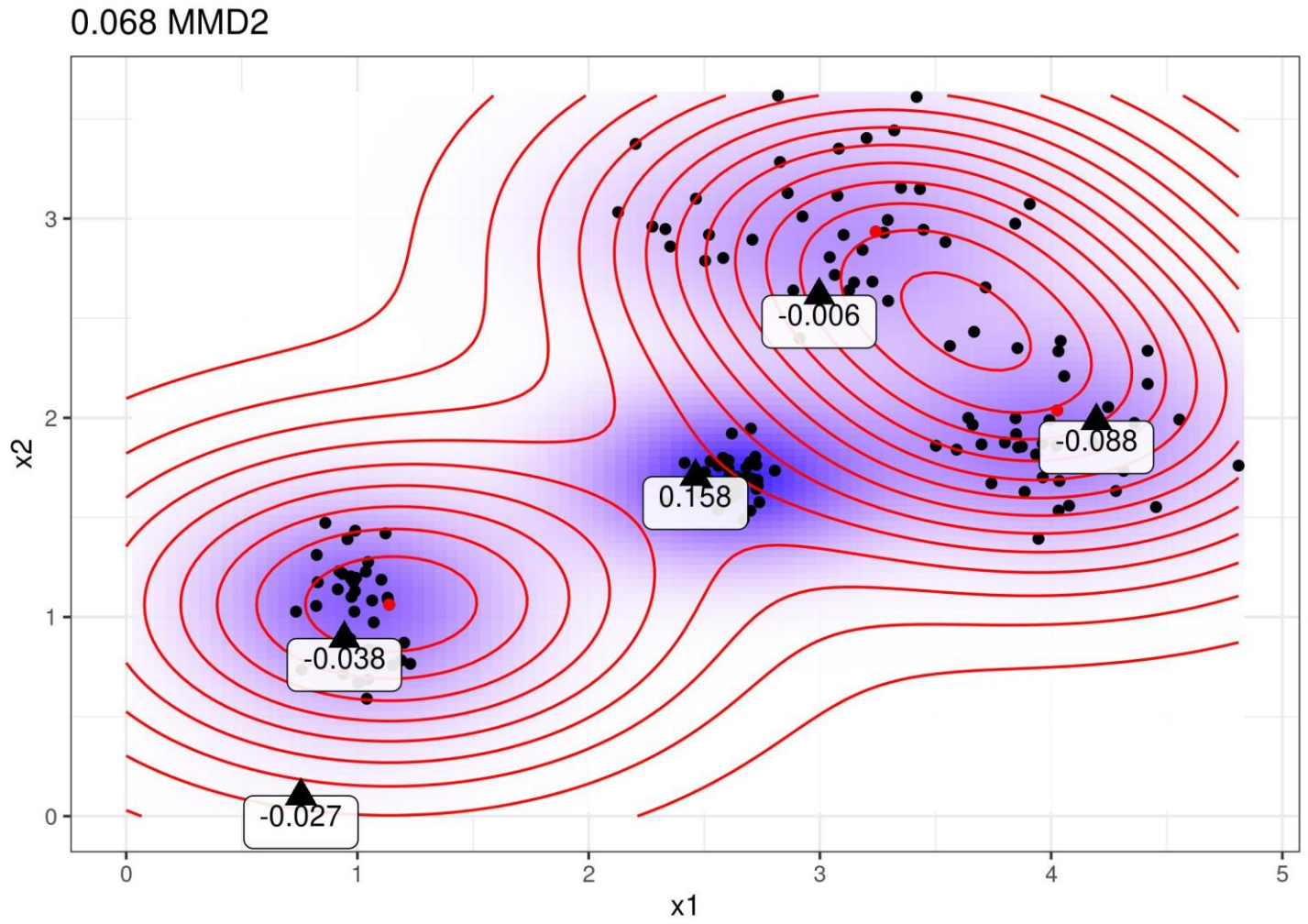
Zero: x에서 두 distribution이 비슷하다.

Negative: prototype에는 fit한데, data에는 fit하지 않다.
- the prototype distribution overestimates the data distribution
- we select a prototype but there are only few data points nearby

Positive: data에 fit한데, prototype에는 fit하지 않다.
- the prototype distribution underestimates the data distribution
- there are many data points around x but we have not selected any prototypes nearby

Prototypes and Criticisms

- Witness function
  - 두 분포의 (Prototype 분포와 data 분포) density 차이에 대한 함수



Prototypes and Criticisms

- **Fining Criticisms**
  - that can maximize the absolute value of witness function (+ regularizer term)
    - Regularizer term : to diversify the selected criticism
  - Using Greedy search

Prototypes and Criticisms

- **How can MMD-critic be used for interpretable machine learning? (3 ways)**

1. helping to better understand the data distribution
   - especially if you have a complex data distribution

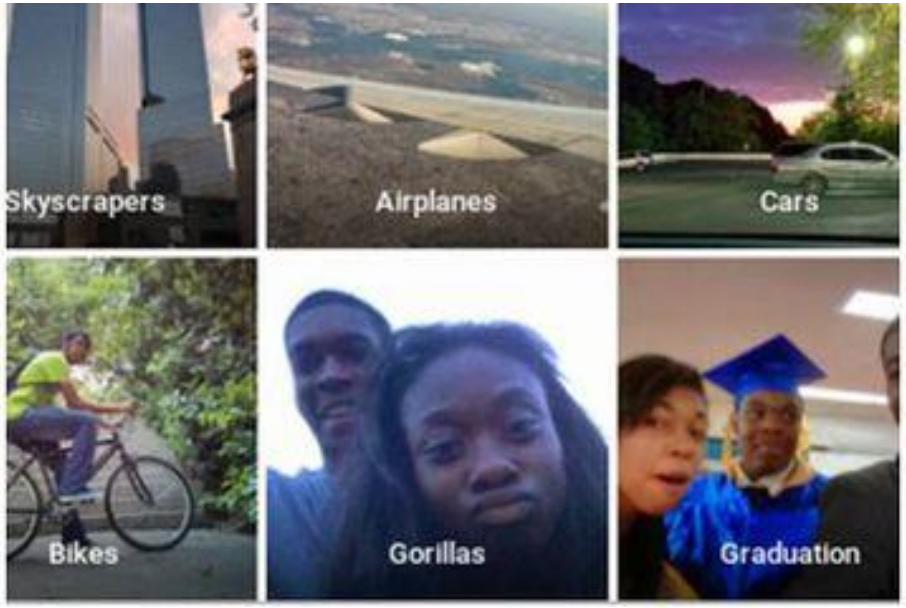
2. create an interpretable prediction model: "nearest prototype model"
   - Find the nearest prototype, which yields the highest value of the kernel function.

$$\hat{f}(x) = argmax_{i \in S} k(x, x_i)$$

Prototypes and Criticisms

- **How can MMD-critic be used for interpretable machine learning? (3 ways)**

3. make any black box model globally explainable
   - Find prototypes and criticisms with MMD-critic.
   - Train a machine learning model as usual.
   - Predict outcomes for the prototypes and criticisms with the machine learning model.
   - Analyse the predictions: In which cases was the algorithm wrong? Now you have a number of examples that represent the data well and help you to find the weaknesses of the machine learning model.



images of a person with dark skin (probably) as a criticism with the notorious "gorilla" classification

Prototypes and Criticisms

- Examples

Examples of prototypes
(handwritten digits dataset)

different coloring, movement of dogs, dogs in costumes, …

Figure 2: Learned prototypes and criticisms from Imagenet dataset (two types of dog breeds)

Prototypes and Criticisms

- Advantages
  - **works with any type of data and any type of machine learning model**.
  - free to **choose the number of prototypes and criticisms**.
  - The algorithm is **easy to implement**.
  - very flexible in the way it is used
    - Understand data / used as an interpretable model / help to understand the predictions of black-box models

- Disadvantages
  - **Distinction between prototype and criticism is based on** the number of prototypes.
    - The two are essentially similar in that they represent something.
  - You have to **choose the number of prototypes and criticisms**