

HUMAN CENTERED MACHINE LEARNING

Lecture 5: Model Agnostic Interpretability Methods

Lecturer: Heysem Kaya



Outline

- Motivation
- Partial Dependence Plot
- Individual Conditional Expectation
- Permutation Feature Importance
- Global Surrogates
- Local Surrogates and LIME
- Supplement:
 - Feature Interaction
 - SHAP



Why do we need post-hoc methods?

- Consider the all resources / constraints in the form of
 - data and relevant pre-trained models,
 - time for the project and budget,
 - expertise of the ML researcher,
 - access to domain expertise,
 - expected product lifespan,
 - need for securing the intellectual property rights etc.

then decide whether an interpretable or black-box model is better

In search of causal effects

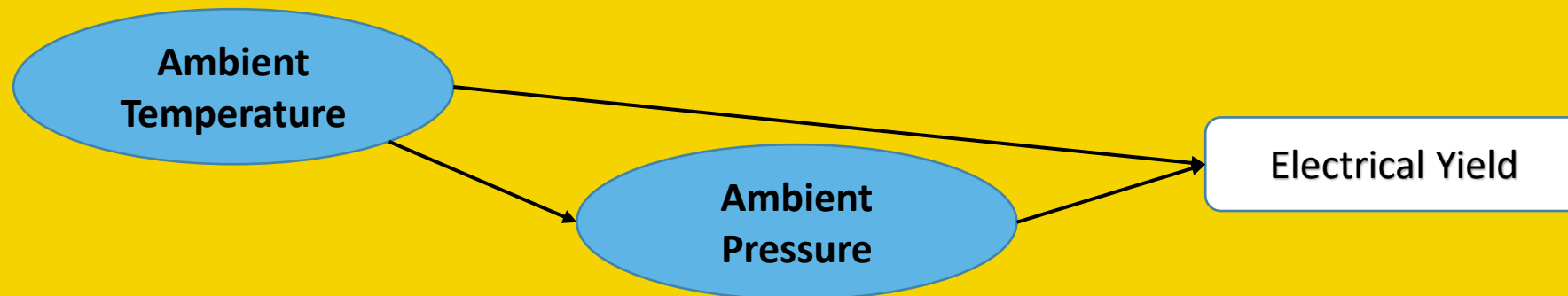
- Feature importance may be analyzed from different perspectives [Z]
 1. Feature weights in Linear Models: which features have the highest impact?
 2. Predictive effects: decrease in node impurity in DTs, permutation method.
 3. Causal effects: how much a change in one variable effects the outcome given all other feature values are fixed.
- Causality: *the predicted change in output due to a perturbation in the input will occur in the real system* [D].
- Causal features are not easy to find, are usually not available.
- Domain expertise may be needed to construct the causal graph.
- Explainability methods such as PDP and ICE may help discover causal effects and probable feature interactions [Z].

[Z] Zhao and Hastie, Causal interpretations of black-box models,, Journal of Business & Economic Statistics, 2019

[D] Doshi-Velez and Kim, Towards A Rigorous Science of Interpretable Machine Learning, ArXiv 2017

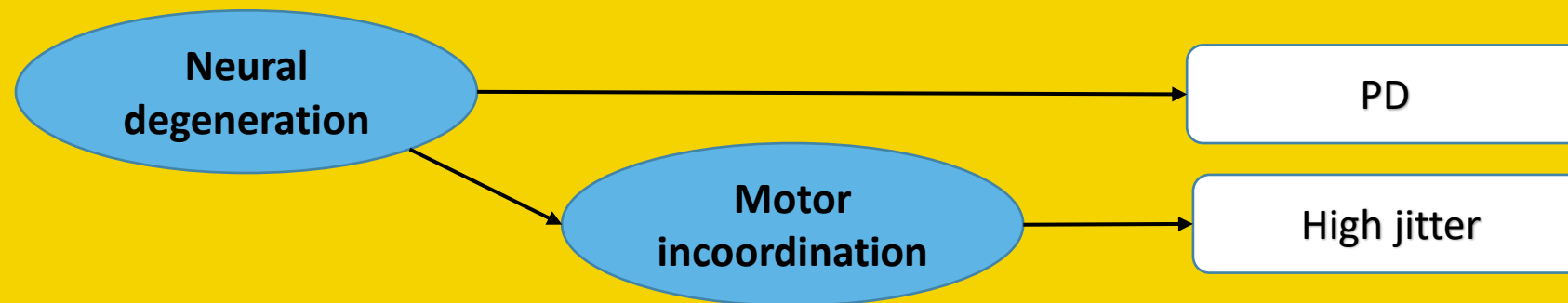
Class discussion: causal or predictive?

- Consider the following prediction problem
 - Predicting hourly power of a gas turbine from ambient variables (AT, AP, RH)
(AT / AP: Ambient Temperature / Pressure, RH: Relative Humidity)
- Which features are causal? Why?
- Remember the Ideal Gas Law: $PV = nRT$
(P: Pressure, V: Volume, n: Number of moles, R: Gas constant, T: Temperature)
 - Lower temperature, fixed PV -> more oxygen and hence higher energy yield



Class discussion: causal or predictive?

- Consider the following prediction problem
 - Predicting Parkinson's Disease using acoustic features (e.g. pitch and jitter)
- Are the acoustic features causal? Why?
- A common cause for PD and acoustic features?



Partial Dependence Plot

- Partial dependence plot aids us to analyze causal effects
- Idea: to analyze a feature / subset, marginalize out the others [F]
- For regression, the PDP for a feature subset x_S is defined as:

$$\hat{f}_{x_S}(x_S) = E_{x_C} [\hat{f}(x_S, x_C)] = \int \hat{f}(x_S, x_C) d\mathbb{P}(x_C)$$

where x_C is the compliment set (gen. all the remaining).

- From training data, we estimate $\hat{f}_{x_S}(x_S)$ by calculating the averages:
 - NB: For each value of x_S we need a run over the training data using the corresponding values for x_C .

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

PDP Toy Example

LivingArea	HasGarden	Bedrooms	Ask Price (K €)
70	0	1	275
80	1	2	300
90	1	2	325
85	0	2	290
120	1	3	400

LivingArea	HasGarden	Bedrooms	\hat{f}
70	0	1	275
80	0	2	285
90	0	2	305
85	0	2	290
120	0	3	375

Avg: 306

- $x_S : \{\text{HasGarden}\}$ with range: $\{0, 1\}$
- For each value v in $\text{range}(x_S)$
 - $\text{output}(v) \leftarrow 0$
 - For i from 1 to N
 - Generate a synthetic sample $s \leftarrow \{x_S = v, x_C^i\}$
 - $\text{output}(v) \leftarrow \text{output}(v) + f(s)$
 - $\text{output}(v) \leftarrow \text{output}(v) / N$
- return output

Output(0)= 306

PDP Toy Example

LivingArea	HasGarden	Bedrooms	Ask Price (K €)
70	0	1	275
80	1	2	300
90	1	2	325
85	0	2	290
120	1	3	400

LivingArea	HasGarden	Bedrooms	\hat{f}
70	1	1	290
80	1	2	300
90	1	2	325
85	1	2	310
120	1	3	400

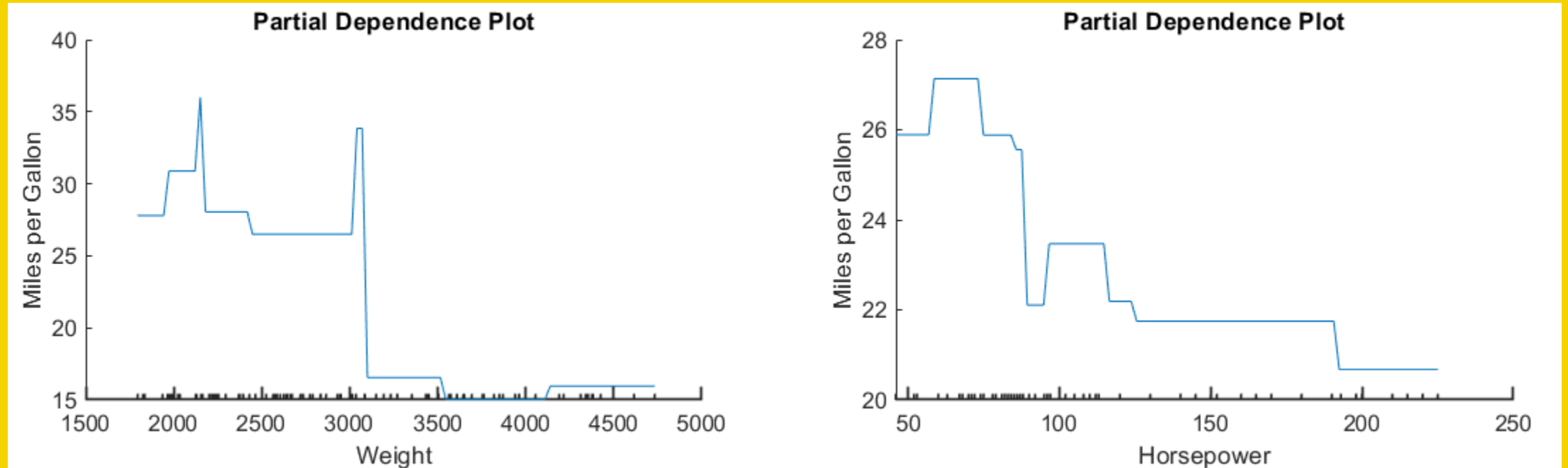
Avg: 325

- $x_S : \{\text{HasGarden}\}$ with range: $\{0, 1\}$
- For each value v in $\text{range}(x_S)$
 - $\text{output}(v) \leftarrow 0$
 - For i from 1 to N
 - Generate a synthetic sample $s \leftarrow \{x_S = v, x_C^i\}$
 - $\text{output}(v) \leftarrow \text{output}(v) + f(s)$
 - $\text{output}(v) \leftarrow \text{output}(v) / N$
- return output

Output(0)= 306

Output(1)= 325

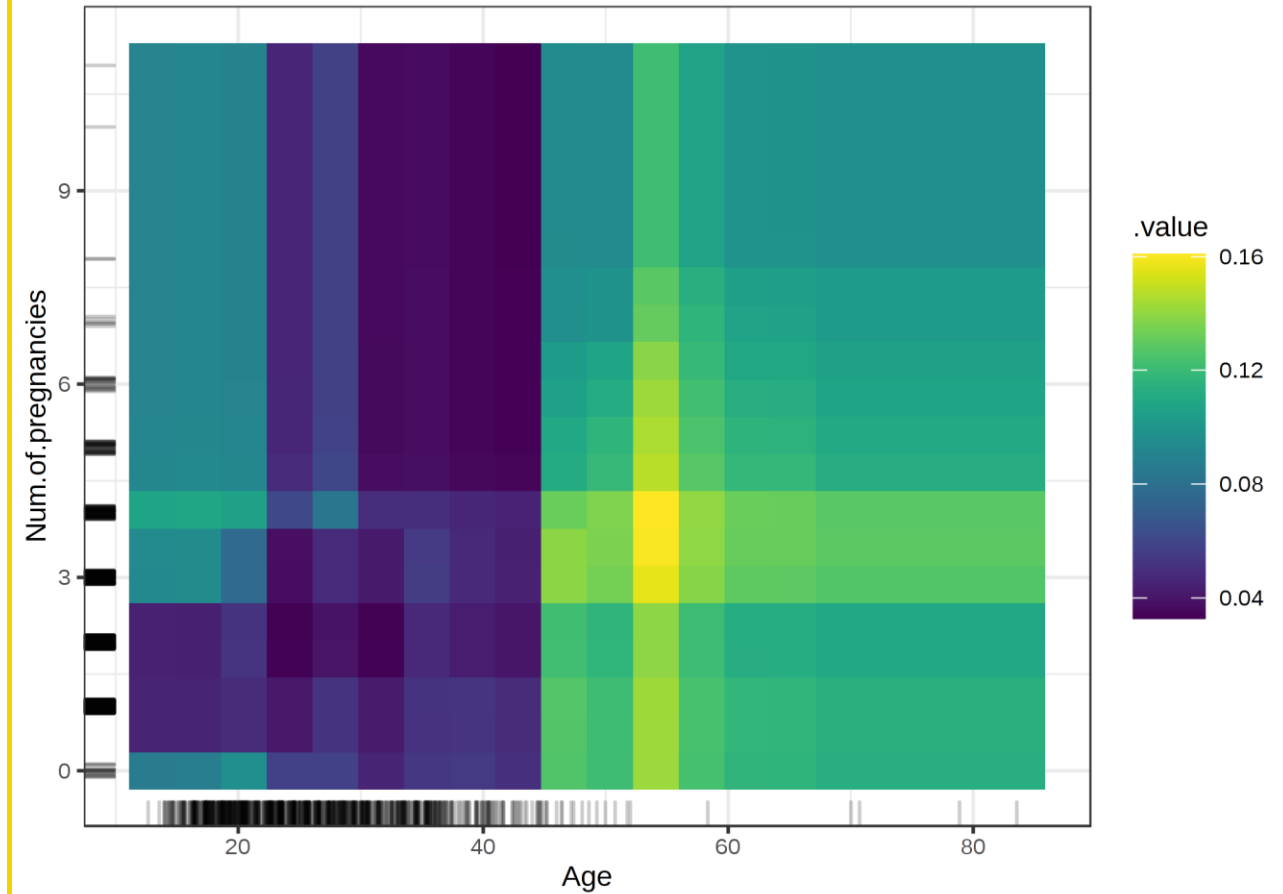
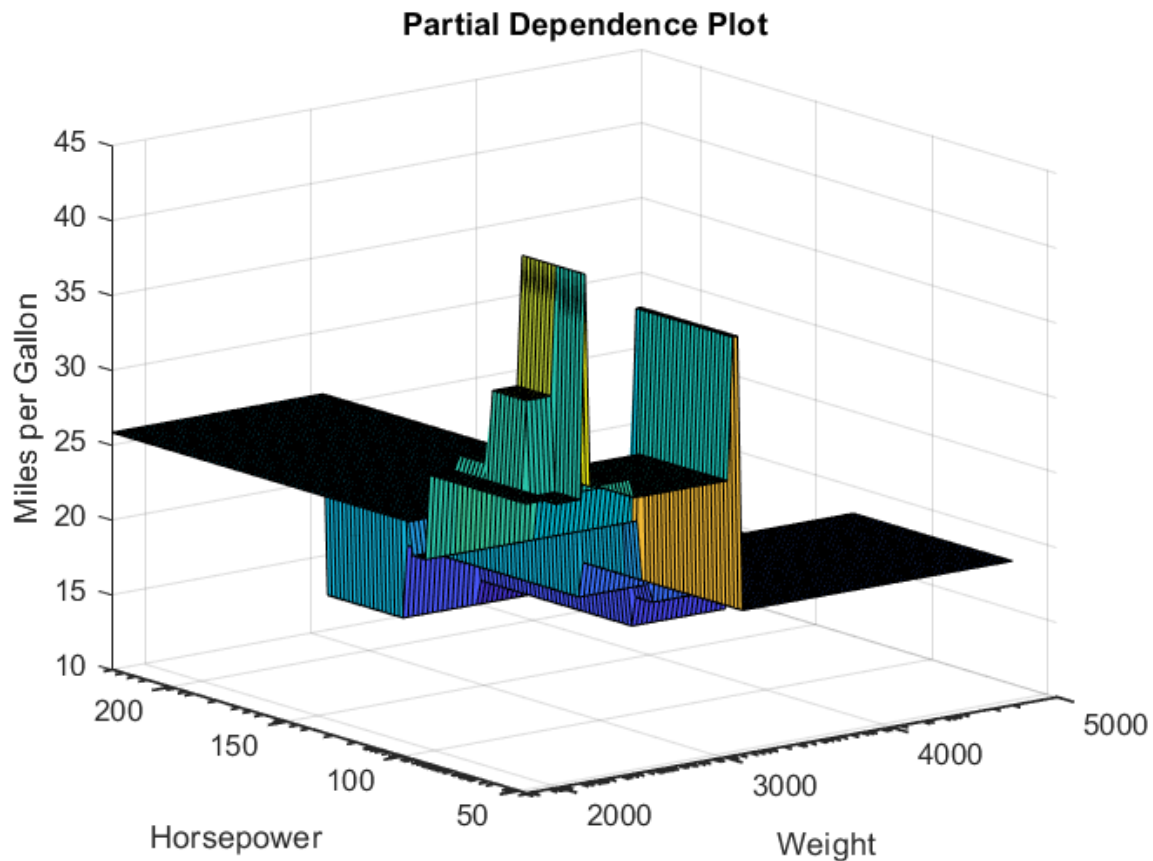
PDP Example on Auto MPG Data*



- Note that there is a very high linear correlation (~ 0.87) between Weight and Horsepower!
- We can hypothesize that at design stage, car weight is a causal factor for engine horsepower.
- This inhibits analysis of individual causal effect of the feature.

How to handle interaction in PDP?

Analyze the interacting features together in PDP. Left: Auto MPG data. Right: Cervical cancer risk plot from [M]





Remarks on Partial Dependence Plot

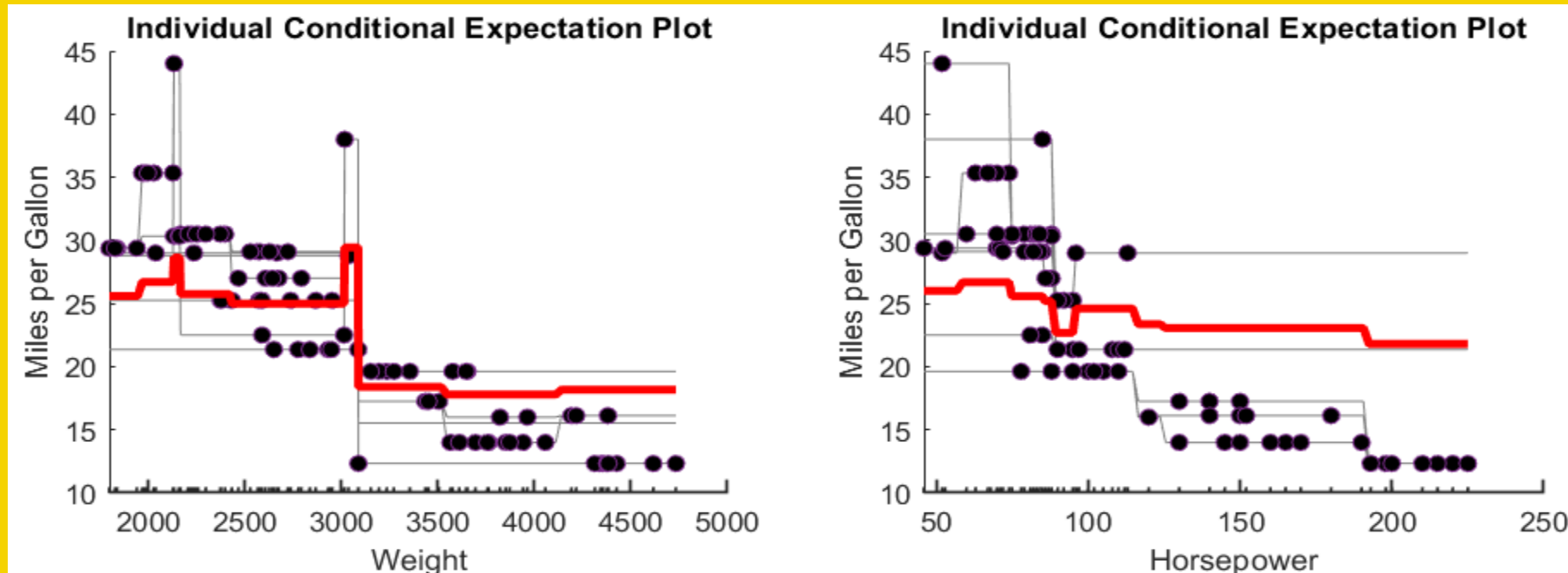
- PDP of x_S is expectation of f over marginal distribution of x_C
- It is not the conditional expectation $E[f(X_S, X_C) | X_S = x_S]$
- For categorical features, the output is one value per category
- For multi-class classification, we can have one plot for each class
- Drawback: having dependence between x_S and x_C may result in anomalies. Why?
- The feature combination given to $f()$ may not be realistic.
- Pearl's 'Back-door criterion' [P], also in [Z]

[P] Pearl, 'Comment: Graphical Models, Causality and Intervention', *Statistical Science*, 1993

[Z] Zhao and Hastie, Causal interpretations of black-box models, *Journal of Business & Economic Statistics*, 2019

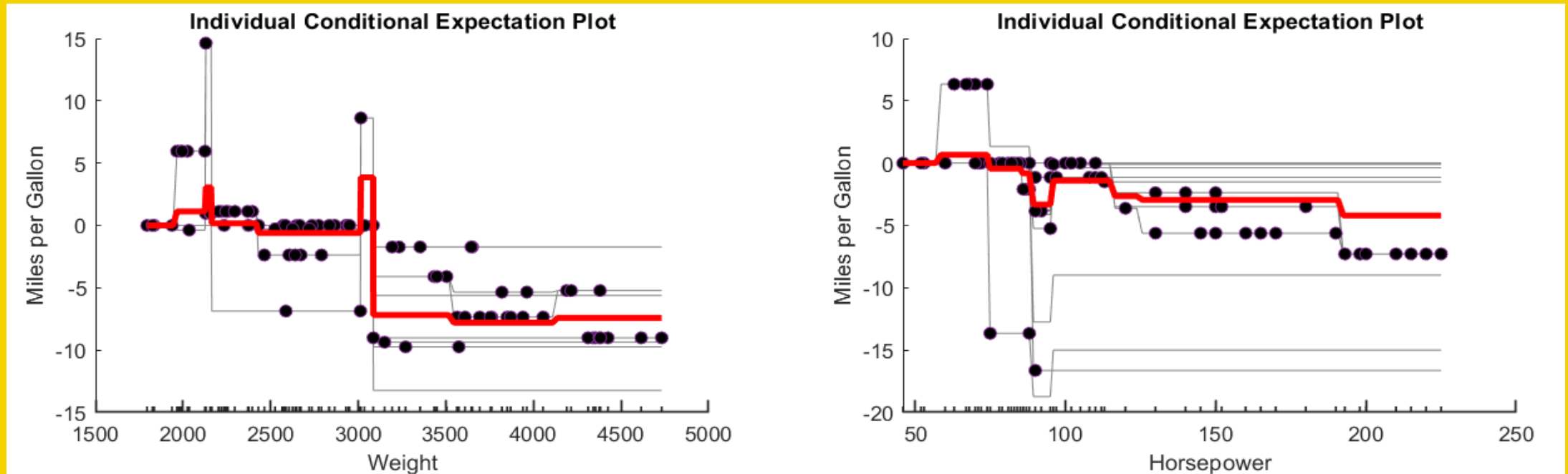
Individual Conditional Expectation

- Extends the PDP to show effect over individual instances [G]
- Helps discover the probable feature interactions
 - If plot pattern changes over instances -> evidence for feature interaction



ICE Centering Trick

- ICE plots start at different values -> hard to tell whether the ICE curves differ between individuals
- A simple solution is centering: i.e. subtracting a reference prediction





Remarks on Individual Conditional Expectation

- The term *conditional expectation* may be misleading. Anomalies in case a feature of x_C is causal descendant of a feature in x_S
- PDP and ICE are commonly used on GAMs for causal effect analysis
- If $\hat{f}(x) = \hat{f}(x_S, x_C) = g(x_S) + h(x_C)$ (i.e. no cross-set interaction)
 - PDP and ICE plots for $f_{x_S}(x_S)$ depend only on $g(x_S)$ as $\sum_i h(x_C^{(i)})$ is constant
 - a derivative ICE plot, $\frac{\delta \hat{f}(x)}{\delta x_S} = g'(x_S)$ can be easily used for causal effects

Permutation Feature Importance Algorithm

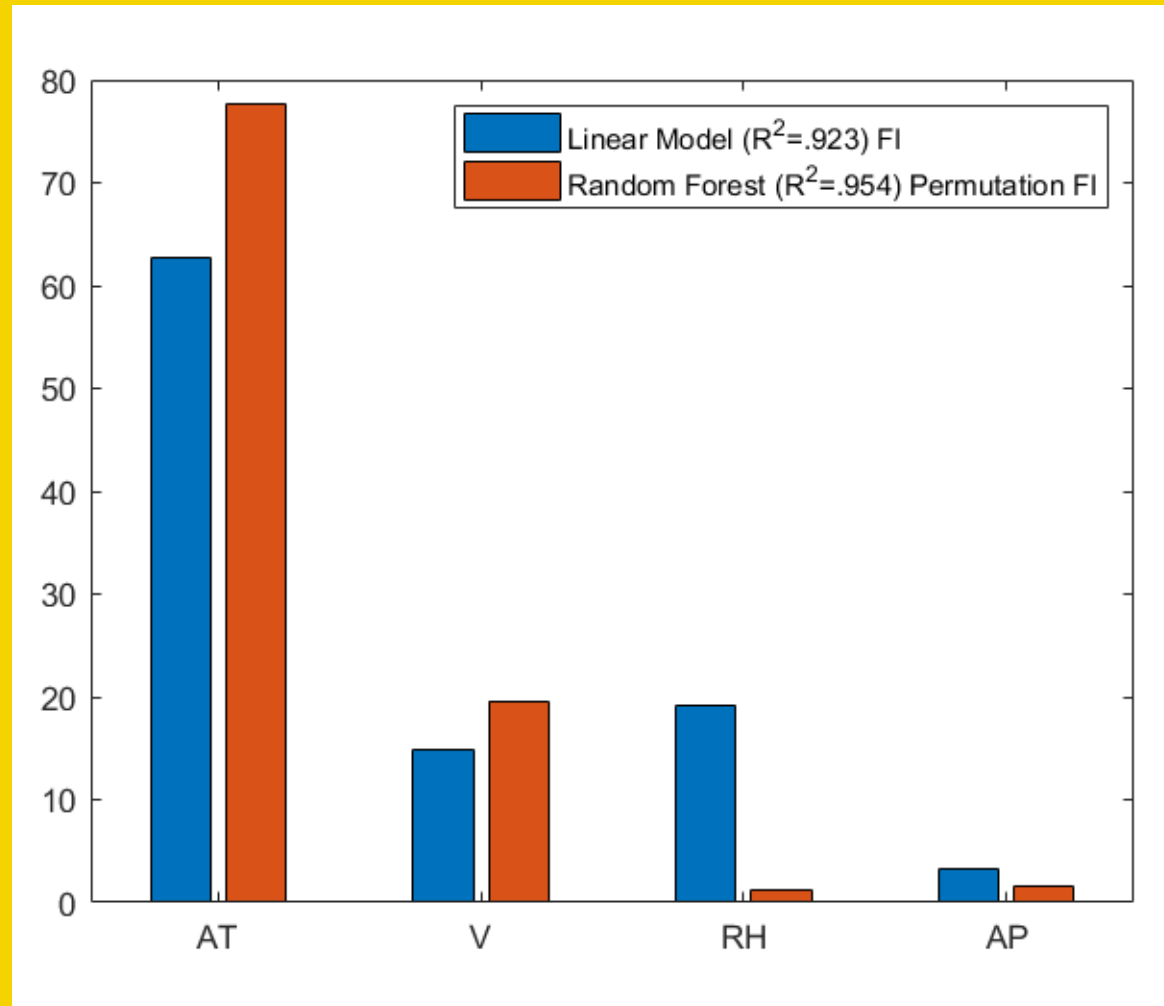
- Input: Trained model f , feature matrix X , target y , error measure $E(y, f)$.
- Estimate the original model error $e^{\text{orig}} = E(y, f(X))$
- For each feature $j = 1, \dots, p$ do:
 - Generate feature matrix X^{perm} by permuting feature j in the data X .
 - Estimate error $e^{\text{perm}} = E(Y, f(X^{\text{perm}}))$ based on the predictions of the permuted data.
 - Calculate permutation feature importance $FI^j = e^{\text{perm}} / e^{\text{orig}}$.
(Alternatively, the difference can be used: $FI^j = e^{\text{perm}} - e^{\text{orig}}$)
- [importance, ranking]=sort(FI, 'descend')

[B] Breiman, Random Forests, Machine Learning, 2001.

[F] Fisher et al., All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously, JMLR, 2019

Permutation Feature Importance Example

Comparison of linear model FI and Random Forest (K=10) based permutation FI on Gas Turbine Dataset*



- Importances are normalized to 100.
- The two methods do not agree in ranking
 - Only the top feature (AT) is common
 - RH importance has the largest gap



Remarks on Permutation Feature Importance

- A simple and effective feature importance attribution method.
- Can be (and is popularly) used for feature ranking / selection.
- Feature interactions and causal relations may still matter.
- Grouping variables for permutation
 - Allows easier interpretation of models using high-dimensional feature sets
 - Allows domain knowledge incorporation
 - Computationally more efficient
 - Handles feature interaction issue

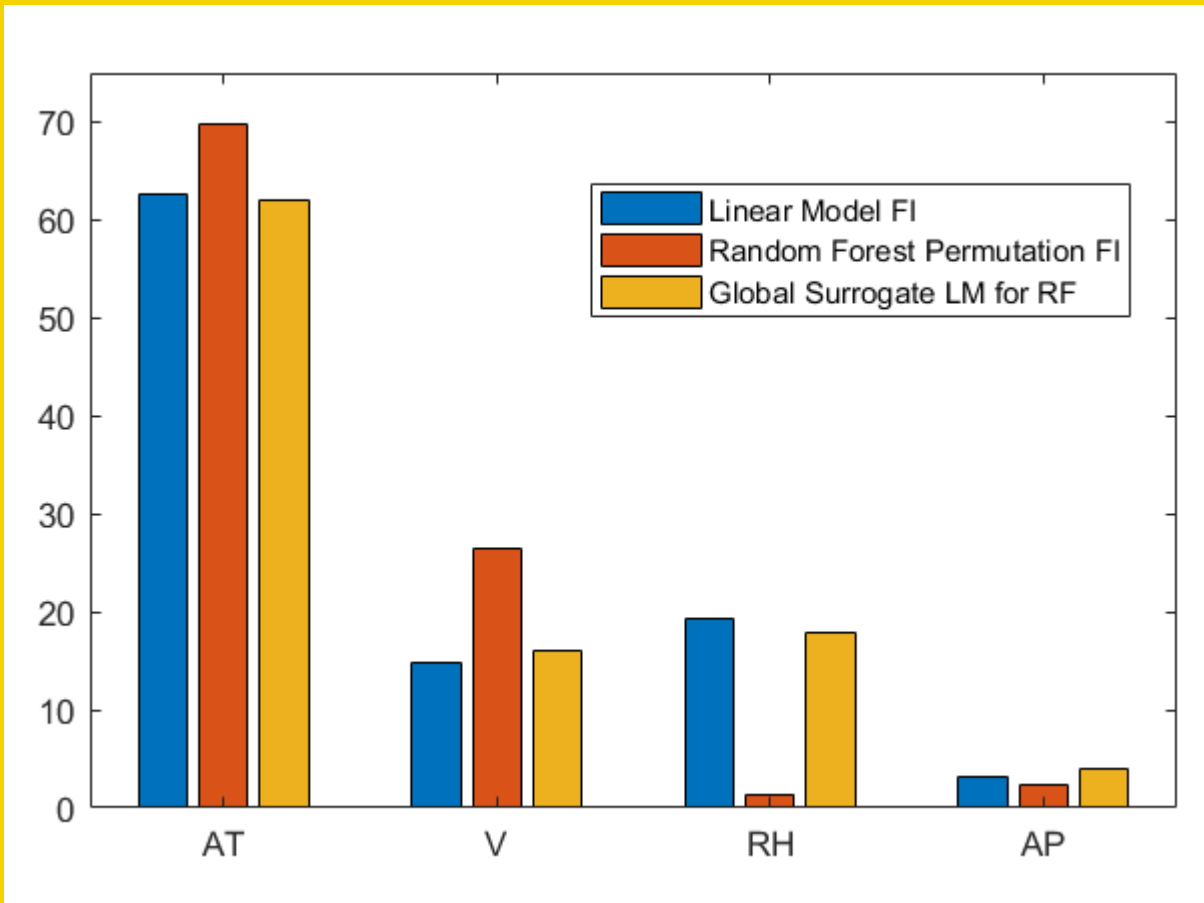
Global Surrogate

- Idea: Approximate the blackbox model with an interpretable model
- Let $\hat{y} = f(x)$ denote the blackbox function and its output
 - We use a training set $X = \{x\}$ and $\hat{Y} = \{f(x)\}$ to train an interpretable model
- A popular practice in the ML sector*



Global Surrogate Example

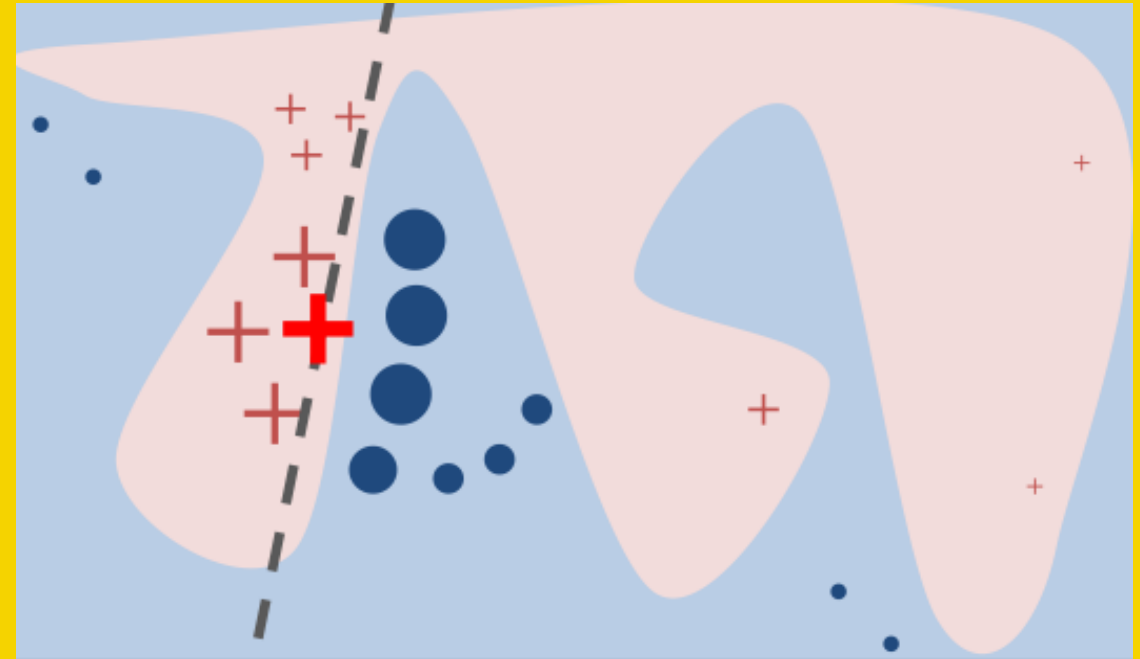
Comparison of a linear model FI and Random Forest (K=10) based permutation FI and FI of a global surrogate linear model for Random Forest on Gas Turbine Dataset*



- Surrogate LM $R^2 = 0.965$ to approximate RF
- What are the anomalies observed?
- Global surrogates may fall short:
 - They may not model global complexity
 - They may not model feature interactions
 - Results may reflect their own structural bias

Local Surrogate

- Data may lie on globally non-linear but locally linear manifolds!
- How to implement?
- A group of instances
 - Clustering: cluster data and fit interpretable local models
 - Sensitive sub-groups
- Explaining a single (test) instance
 - Generate data *similar* to a test instance and fit an interpretable model
 - Exemplar method: Local Interpretable Model-Agnostic Explanations (LIME)



LIME

- “A novel explanation technique that explains the predictions of any classifier”
 - in an interpretable and faithful manner,
 - by learning an interpretable model
 - locally around the prediction»
- Focus on instance-wise predictions to overcome
 - Data leakage: in some cases subject IDs have high correlations with target class
 - Dataset (covariance) shift: the distribution of inputs/outputs are different in training and test and/or in real life

LIME: Desired Characteristics for Explainers

1. **interpretability**: provide qualitative understanding between the input variables and the response
2. **local fidelity**: it must correspond to how the model behaves in the vicinity of the instance being predicted
3. **model-agnostic**: explainer should be able to explain any model
4. providing a **global perspective** to evaluate the model via explanations

LIME: Definitions

- Complexity of an explainable model $\Omega(g)$
 - the depth of a decision tree (DT)
 - the number of non-zero weight elements for a linear model
- let $\mathbf{f}: \mathbf{R}^d \rightarrow \mathbf{R}$ be the predictive model mapping the original features to the target
- and let $g: \{0, 1\}^p \rightarrow \mathbf{R} \in G$ be an interpretable model whose complexity is measured with $\Omega(g)$, and
- $\pi_x(z)$ denote the proximity measure between x and z (so as to define locality around x)
- model tries to minimize the total loss: $E(x) = \min_{g \in G} L(f, g, \pi_x) + \Omega(g)$

LIME with Linear Models

- Explanations are **feature weights** for linear models.
- They are tailored for each predicted class.
- For $K > 2$ class classification, explanation is given for the predicted class.

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f , Number of samples N

Require: Instance x , and its interpretable version x'

Require: Similarity kernel π_x , Length of explanation K

$\mathcal{Z} \leftarrow \{\}$

for $i \in \{1, 2, 3, \dots, N\}$ **do**

$z'_i \leftarrow \text{sample_around}(x')$

$\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$

end for

$w \leftarrow \text{K-Lasso}(\mathcal{Z}, K)$ \triangleright with z'_i as features, $f(z)$ as target

return w

LIME: Model Explanation by Submodular Pick

- For model explanation, a set of n predictions are casted first to obtain a $n \times d'$ dimensional weight matrix W .
- Here, the absolute value of the weights are used in W construction.
- The sum of weights for each feature j characterizes the feature importance I_j .
- The model explanation algorithm tries to
 - maximize coverage of features used in global explanation by eliminating redundancy in selected examples
 - while simultaneously picking instances using important features:

$$c(V, W, I) = \sum_{j=1}^{d'} \mathbf{1}[\exists i \in V: W_{ij} > 0] I_j$$

LIME: Model Explanation by Submodular Pick

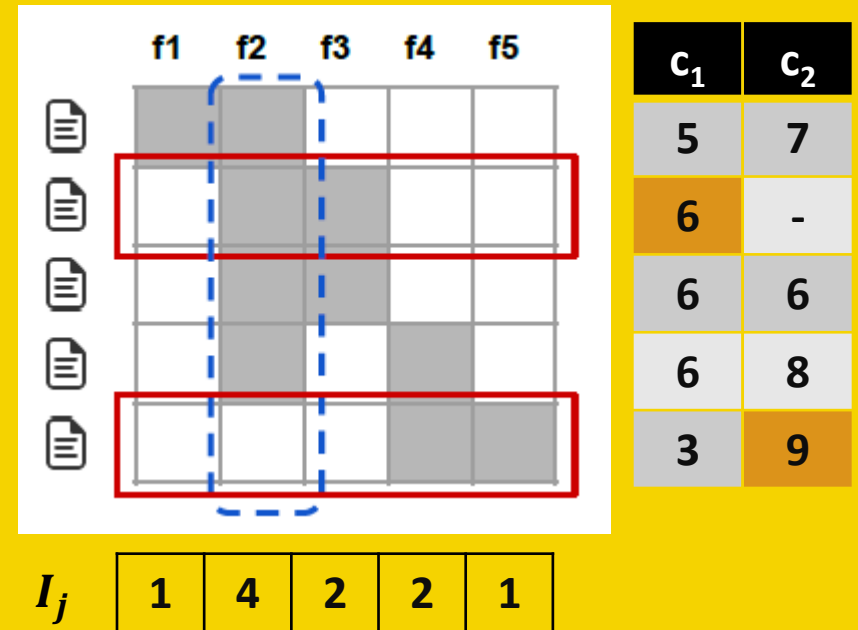
Algorithm 2 Submodular pick (SP) algorithm

Require: Instances X , Budget B

```

for all  $x_i \in X$  do
   $\mathcal{W}_i \leftarrow \text{explain}(x_i, x'_i)$  ▷ Using Algorithm 1
end for
for  $j \in \{1 \dots d'\}$  do
   $I_j \leftarrow \sqrt{\sum_{i=1}^n |\mathcal{W}_{ij}|}$  ▷ Compute feature importances
end for
 $V \leftarrow \{\}$ 
while  $|V| < B$  do ▷ Greedy optimization of Eq
   $V \leftarrow V \cup \operatorname{argmax}_i c(V \cup \{i\}, \mathcal{W}, I)$ 
end while
return  $V$ 
  
```

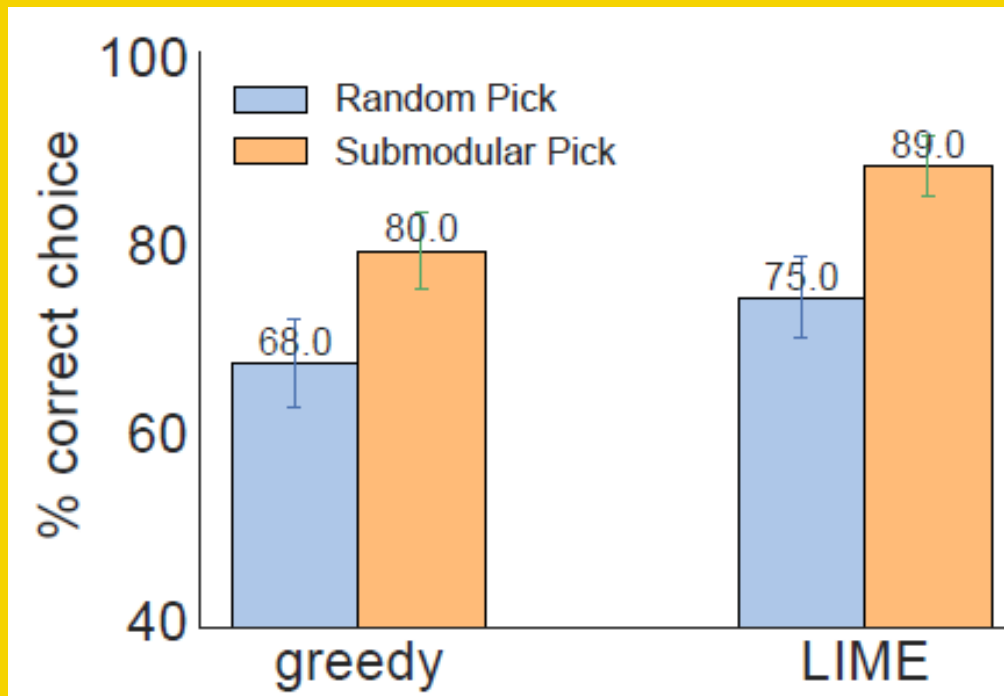
Toy Example



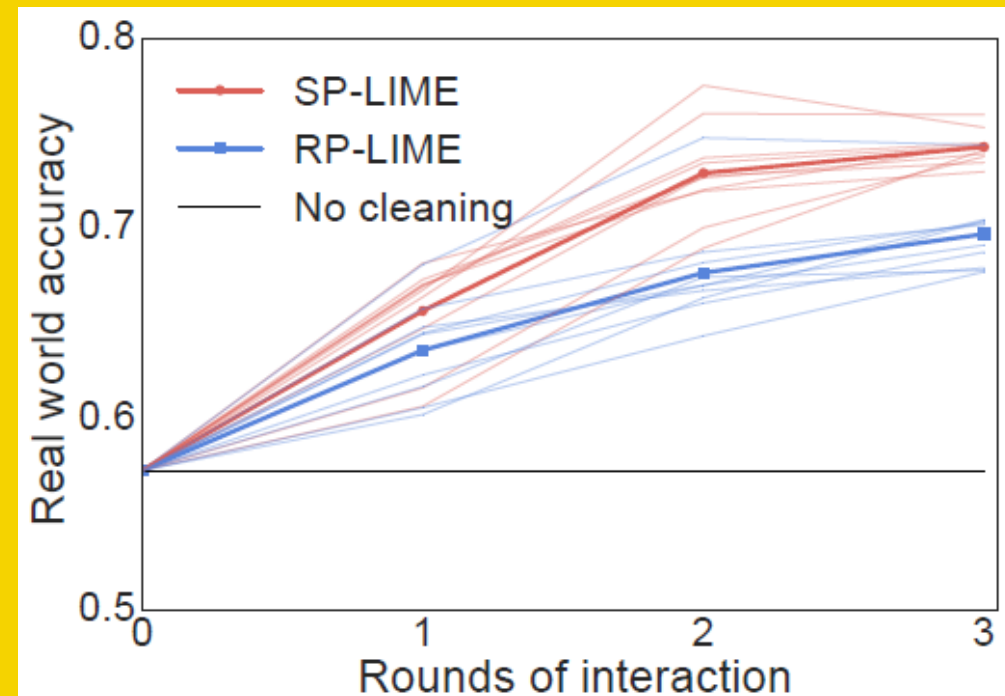
$$c(V, W, I) = \sum_{j=1}^{d'} \mathbf{1}[\exists i \in V: W_{ij} > 0] I_j$$

Effect of Submodular Pick

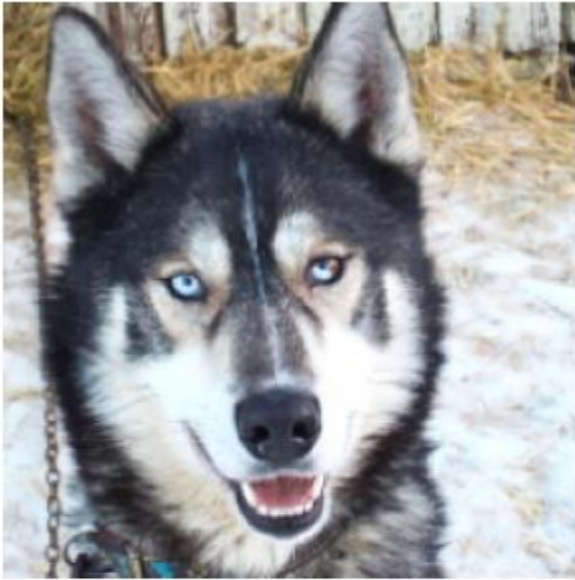
Average accuracy of human subject in choosing between two classifiers.



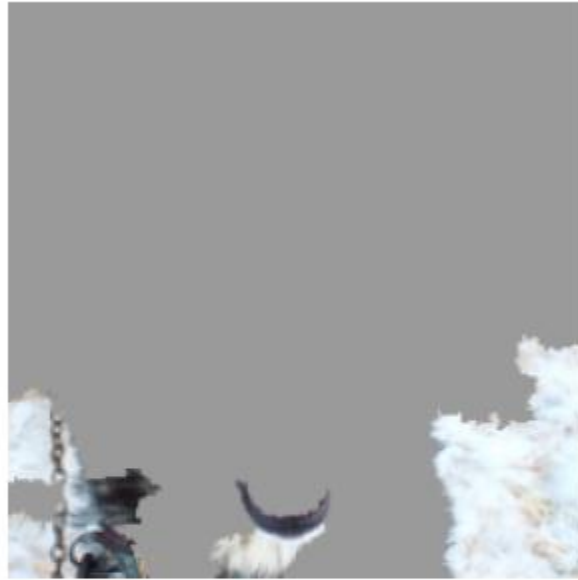
Ordinary people (AMT) in the loop for feature selection based on LIME.



Husky vs Wolf Experiment



(a) Husky classified as wolf



(b) Explanation

- Image based applications use superpixels (Superpixel: a cluster of connected pixels sharing a similar color)
- Text based applications: use Bag of Words and remove words for sampling

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27



LIME

Summary & Remarks

- Instance-wise explanations are shown to support both experts and non-expert humans on
 - model selection,
 - assessing trust,
 - improving untrustworthy models,
 - and getting insights into predictions.
- It inspired other works with a stronger theoretical background, e.g. SHAP.
- Flexibility or vulnerability*: How to
 - define proximity measure $\pi_x(z)$?
 - choose the interpretable version x' ?
 - sample around x' ?

Literature for today

- Lecture 5: Model Agnostic Interpretability Methods
- Required Reading
 - C. Molnar's book, [Chapter 5](#)
 - [LIME] "Why Should I Trust You?" Explaining the Predictions of Any Classifier, Ribeiro et al. KDD 2016 (<https://www.kdd.org/kdd2016/papers/files/rfp0573-ribeiroA.pdf>)
- Suggested Reading
 - [SHAP] A Unified Approach to Interpreting Model Predictions, Lundberg and Lee, NeurIPS 2017 (<http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf>)

Literature next lecture

- Lecture 6: Neural Network Interpretability
- Required Reading
 - C. Molnar's book, [Chapter 7](#)
 - Grad-Cam: Visual explanations from deep networks via gradient-based localization, Selvaraju et al. ICCV 2017.
https://openaccess.thecvf.com/content_ICCV_2017/papers/Selvaraju_Grad-CAM_Visual_Explanations_ICCV_2017_paper.pdf
- Suggested Reading
 - Layer-Wise Relevance Propagation: An Overview, Montavon et al. Explainable AI: Interpreting, Explaining and Visualizing Deep Learning 2019
https://link.springer.com/chapter/10.1007/978-3-030-28954-6_10

Quiz for today



A quiz for this lecture is available on course Blackboard page.



This is not graded but meant to provide formative evaluation.



The deadline for submitting the quiz is Wednesday May 26, 17:00!



We will be discussing the answers during the next lecture.



Supplement

- The rest of the slides may be covered if we have time remaining.
- They will not be included in the exam.

Feature Interaction

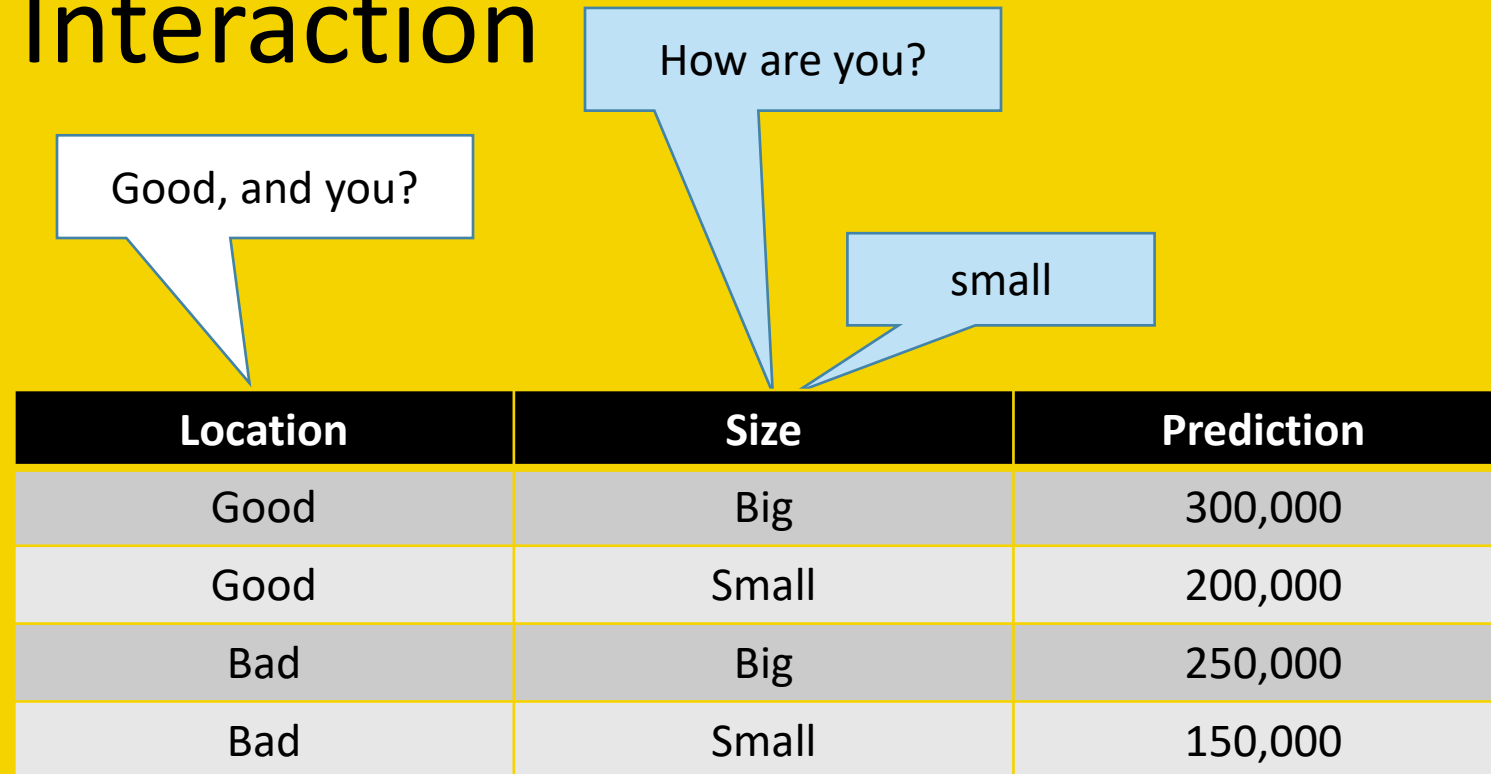


Diagram illustrating feature interaction with callouts:

- Callout: "Good, and you?" points to the first row (Good location, Big size).
- Callout: "How are you?" points to the second row (Good location, Small size).
- Callout: "small" points to the second row (Good location, Small size).

Location	Size	Prediction
Good	Big	300,000
Good	Small	200,000
Bad	Big	250,000
Bad	Small	150,000

Feature interaction: effect of one feature depends on the value of another

Feature Interaction

A prediction table for house price using two features. Example from [M]

Location	Size	Prediction
Good	Big	300,000
Good	Small	200,000
Bad	Big	250,000
Bad	Small	150,000

- Consider mapping binary attributes to $\{0, 1\}$ such that ‘good’ location and ‘big’ size evaluate to 1.
- Then a perfect linear model decomposition can be obtained: $f(x) = w_0 + w_1x_1 + w_2x_2$,
where $w_0 = 150.000$, $w_{Location} = 50.000$, and $w_{Size} = 100.000$

Feature Interaction

Now the effect of size depends on location, i.e. features *interact*!

Location	Size	Prediction
Good	Big	400,000
Good	Small	200,000
Bad	Big	250,000
Bad	Small	150,000

- Thus, we should include an interaction term $f(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2$ where $w_0 = 150.000$, $w_{Location} = 50.000$, and $w_{Size} = 100.000$, $w_{Int} = 100.000$

Feature Interaction – How to quantify?

- Several methods are proposed to quantify feature interaction [H, F, G]
- We will focus on Friedman and Popescu's approach [H]
- Recall additive $\hat{f}(x) = \hat{f}(x_S, x_C) = g(x_S) + h(x_C)$
- Lets denote partial dependence function of x_j as $PD_j(x_j)$
- If x_i and x_j do not interact, then $PD_{ij}(x_i, x_j) = PD_i(x_i) + PD_j(x_j)$
- If x_i does not interact with any other feature, in complement set C:

$$\hat{f}(x) = PD_i(x_i) + PD_C(x_C)$$

[H] Hooker, "Discovering additive structure in black box functions", Proc KDDM, 2004.

[F] Friedman and Popescu, "Predictive learning via rule ensembles." The Annals of Applied Statistics, 2008.

[G] Greenwell et al., "A simple and effective model-based variable importance measure." arXiv 2018.

Friedman's H-statistic

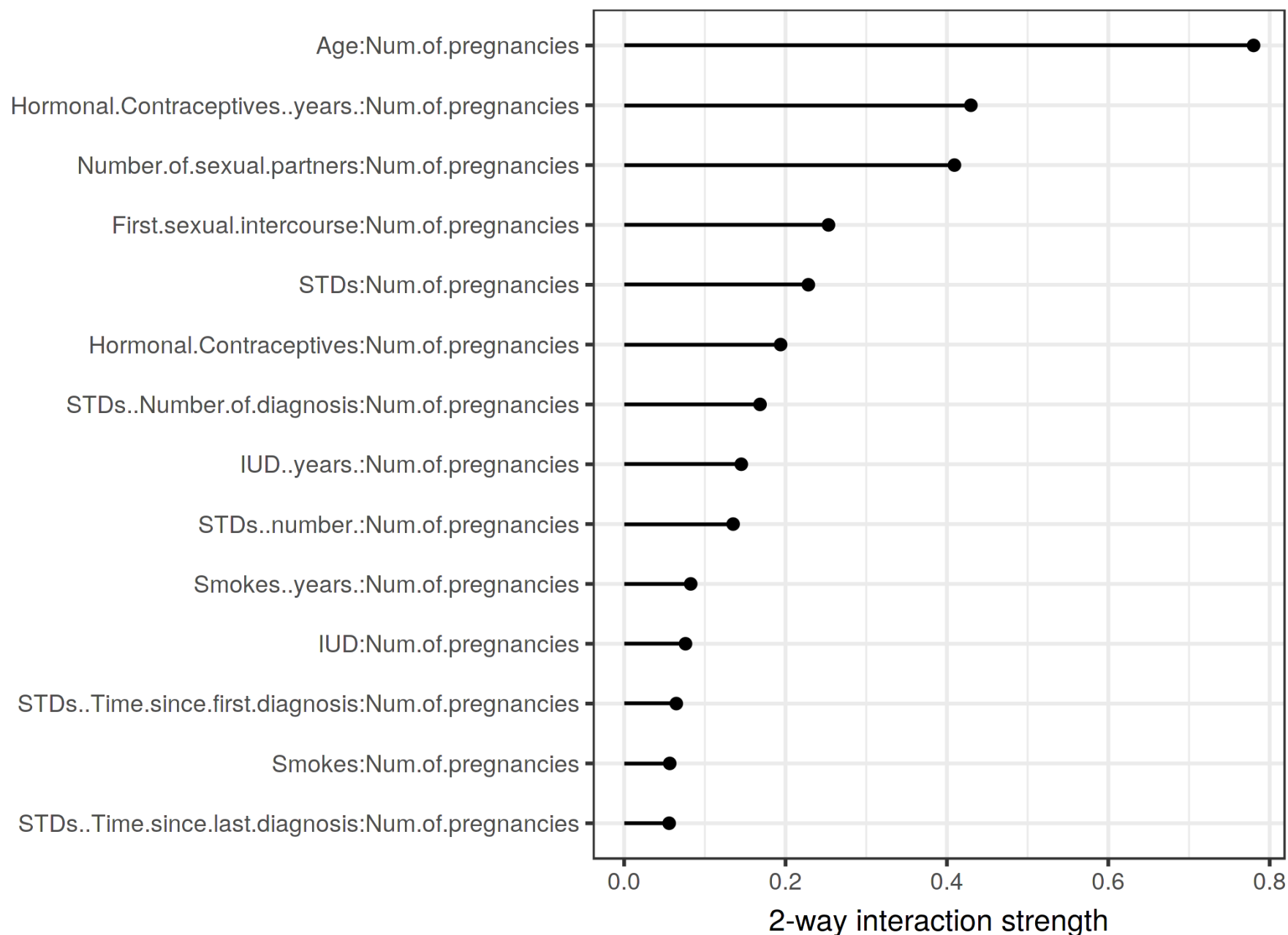
- Idea: Calculate the proportion of variance explained by the interaction effect
- Pairwise interaction H-statistic:

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

- One feature vs. rest H-statistic:

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

- Clearly, higher the H-statistic, higher the interaction. Significance?



Cervical cancer task
feature interactions.
Figure from [M].



Remarks on Friedman's H-Statistic

- Pros
 - Theory is based on partial dependence functions
 - Can be applied to any model, comparable across features
 - Can be used on interactions of any type and dimension
- Cons
 - Multiway interactions are computationally complex
 - Test statistic requires model adjustment -> model specific
 - Does not tell the shape of an interaction
- Mediation
 - Bivariate PDP analysis for pairs of strongly interacting features

SHapley Additive exPlanations

- SHAP argues to unify seven methods including LIME, and in some ways extends the basic ideas from it.
 - Particularly, locally faithful and simplified (linear) binary features based explanation models.
- Let f be the original prediction model to be explained and g the explanation model.
 - Focus is on local methods designed to explain a prediction $f(x)$ based on a single input x , as proposed in LIME.
- SHAP is inspired from LIME, but it combines all relevant methods in a game-theoretic framework.

SHAP: Definitions

- **Def 1** Additive feature attribution methods have an explanation model that is a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i,$$

where $z' \in \{0, 1\}^M$, M is the number of simplified input features, and $\phi_i \in \mathbb{R}$.

- The feature attributions (importance scores) can be correctly estimated from game-theory*

* S Shapley. "A value for n-person games". In: Contributions to the Theory of Games, 1953

Lloyd Stan Lipovetsky and Michael Conklin. "Analysis of regression in game theory approach". In: Applied Stochastic Models in Business and Industry, 2001

Shapley Regression Values

- SRV are feature attribution scores that are weighted average of differences of model predictions trained on possible subsets of $\mathbf{F} \setminus \{i\}$:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

- Problem: 2^D possible subsets: infeasible
- Solution: We can use approximations (sampling)
- This combinatorial value is known and used in game theory but was not used in explanation models.

Desired Properties for Additive Feature Attributions

1. **Local Accuracy:** The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$, where $\phi(0) = f(h_x(0))$ represents the model output with all simplified inputs are missing.
 2. **Missingness** constrains features where $x'_i = 0$ to have no attributed impact.
 3. **Consistency** states that if a model changes so that some simplified input's contribution increases or stays the same regardless of the other inputs, that input's attribution should not decrease.
- Unique solution: Shapley Values

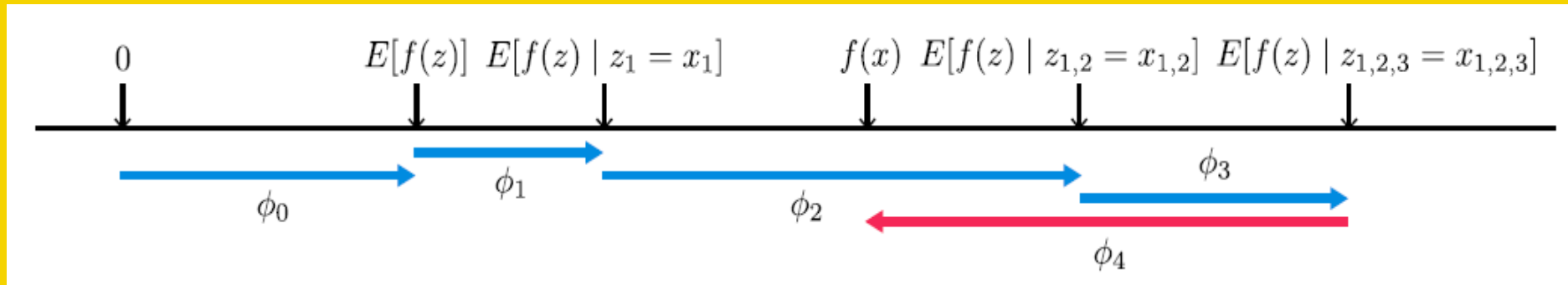
SHAP: Theorem 1

- Only one possible explanation model g follows Definition 1 and satisfies Properties 1, 2, and 3:

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

- where $|z'|$ is the number of non-zero entries in z' , and $z' \subseteq x'$ represents all z' vectors where the non-zero entries are a subset of the non-zero entries in x' .

SHAP Values



- SHAP values attribute to each feature the change in the expected model prediction when conditioning on that feature.
- They explain how to get from the base value $E[f(z)]$ that would be predicted if we did not know any features to the current output $f(x)$.

SHAP: Approximation

- SHAP value computation is hard and even infeasible when $|S|$ is high.
- Approximations assuming feat. independence and (local) model linearity to compute $f(h_x(z'))$:

$f(h_x(z')) = E[f(z) \mid z_S]$	SHAP explanation model simplified input mapping
$= E_{z_{\bar{S}} \mid z_S}[f(z)]$	expectation over $z_{\bar{S}} \mid z_S$
$\approx E_{z_{\bar{S}}}[f(z)]$	assume feature independence
$\approx f([z_S, E[z_{\bar{S}}]]).$	assume model linearity

- If we assume feature independence when approximating conditional expectations, then SHAP values can be estimated directly using the Shapley sampling values method.

Kernel SHAP

Remember LIME:

$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Theorem 2 (Shapley kernel) *Under Definition 1, the specific forms of $\pi_{x'}$, L , and Ω that make solutions of Equation 2 consistent with Properties 1 through 3 are:*

$$\Omega(g) = 0,$$

$$\pi_{x'}(z') = \frac{(M-1)}{(M \text{ choose } |z'|)|z'|(M-|z'|)},$$

$$L(f, g, \pi_{x'}) = \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_{x'}(z'),$$

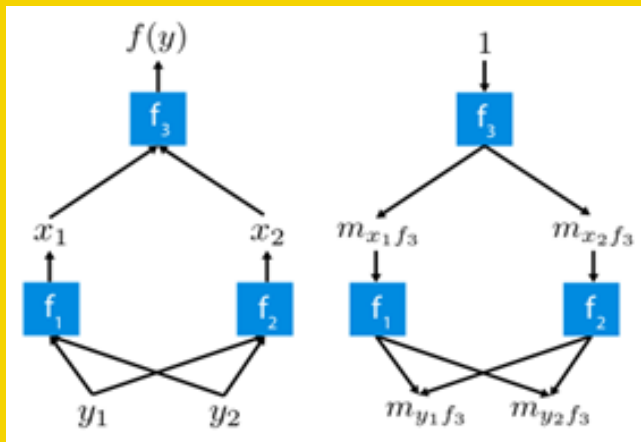
where $|z'|$ is the number of non-zero elements in z' .

Model Specific SHAP Approximations

- **Linear SHAP** (proportional to the linear reg. scores):

$$f(x) = \sum_{j=1}^M w_j x_j + b: \phi_0(f, x) = b \text{ and } \phi_i(f, x) = w_j(x_j - E[x_j])$$

- **Deep SHAP (DeepLIFT + Shapley values)**: Forward computation and backward importance weight estimation.



$$\begin{aligned} m_{x_j f_3} &= \frac{\phi_i(f_3, x)}{x_j - E[x_j]} \\ \forall_{j \in \{1, 2\}} \quad m_{y_i f_j} &= \frac{\phi_i(f_j, y)}{y_i - E[y_i]} \\ m_{y_i f_3} &= \sum_{j=1}^2 m_{y_i f_j} m_{x_j f_3} && \text{chain rule} \\ \phi_i(f_3, y) &\approx m_{y_i f_3} (y_i - E[y_i]) && \text{linear approximation} \end{aligned}$$