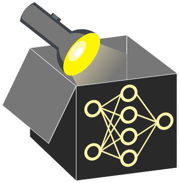
Interpretable Machine Learning



Feature Importances 1

Permutation Feature Importance (PFI)

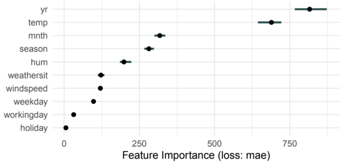
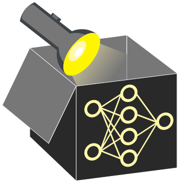


Figure: Bike Sharing Dataset

Learning goals

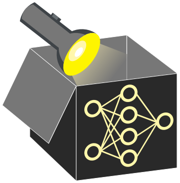
- Understand how PFI is computed
- Understanding strengths and weaknesses

MOTIVATION FOR PFI



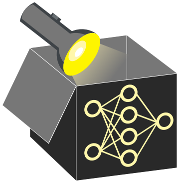
- **Goal:** Assess how important feature(s) X_S are for predictive performance of a **fixed trained model** $\hat{f}$ on a given dataset $\mathcal{D}$
- **Idea:** Estimate performance change when X_S is "made uninformative"

MOTIVATION FOR PFI



- **Goal:** Assess how important feature(s) X_S are for predictive performance of a **fixed trained model** $\hat{f}$ on a given dataset $\mathcal{D}$
- **Idea:** Estimate performance change when X_S is "made uninformative"
- **Question:** Can we make X_S uninformative by removing it from model?
  $\rightsquigarrow$ No, $\hat{f}$ was trained with X_S; retraining without X_S gives a different model

MOTIVATION FOR PFI



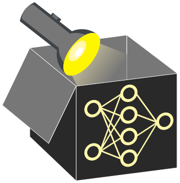
- **Goal:** Assess how important feature(s) X_S are for predictive performance of a **fixed trained model** $\hat{f}$ on a given dataset $\mathcal{D}$
- **Idea:** Estimate performance change when X_S is "made uninformative"
- **Question:** Can we make X_S uninformative by removing it from model?
  $\rightsquigarrow$ No, $\hat{f}$ was trained with X_S; retraining without X_S gives a different model
- **Solution:** Simulate feature removal by replacing X_S with a perturbed version $\tilde{X}_S$ that is independent of (X_{-S}, Y) but preserves distrib. $\mathbb{P}(X_S)$
  $\rightsquigarrow$ Compare baseline predictions $\hat{f}(X)$ with perturbed predictions $\hat{f}(\tilde{X}_S, X_{-S})$

$$\text{PFI}_S := \underbrace{\mathbb{E}\left[L\left(\hat{f}(\tilde{X}_S, X_{-S}), Y\right)\right]}_{\text{risk after "destroying" } X_S} - \underbrace{\mathbb{E}\left[L\left(\hat{f}(X), Y\right)\right]}_{\text{baseline risk}},$$

- **How to perturb X_S?**
  - Add random noise: distorts $\mathbb{P}(X_S)$ (not used)
  - Permutation: preserves marginal $\mathbb{P}(X_S)$, breaks dependence with Y (used)

PERMUTATION FEATURE IMPORTANCE (PFI)

▸ BREIMAN_2001



Sample estimator (using independent test set $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{n}$)

- Measure error **with feat. values** x_S and **with permuted feat. values** x_S
- Repeat permutation (e.g., m times) and average difference of both errors:

$$\widehat{PFI}_S = \frac{1}{m} \sum_{k=1}^{m} \left[\mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}S_{(k)}) - \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D})\right]$$

- $\mathcal{D}_S^{(k)}$: dataset with column(s) x_S are **permuted** once (in repetition k)
- $\mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{(x,y)\in\mathcal{D}} L(\hat{f}(x), y)$: Measures performance of $\hat{f}$ using $\mathcal{D}$
- Average over m permutations to reduce Monte-Carlo variance

Example of permuting feature x_S with $S = \{1\}$ and $m = 6$ permutations:

$\mathcal{D}$

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	7
2	5	8
3	6	9

$\Rightarrow$

$\tilde{\mathcal{D}}^S_{(1)}$

$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	7
2	5	8
3	6	9

$\tilde{\mathcal{D}}^S_{(2)}$

$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
2	4	7
1	5	8
3	6	9

$\tilde{\mathcal{D}}^S_{(3)}$

$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
2	4	7
3	5	8
1	6	9

$\tilde{\mathcal{D}}^S_{(4)}$

$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	7
3	5	8
2	6	9

$\tilde{\mathcal{D}}^S_{(5)}$

$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
3	4	7
1	5	8
2	6	9

$\tilde{\mathcal{D}}^S_{(6)}$

$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
3	4	7
2	5	8
1	6	9

Note: S refers to a subset of features, here $|S| = 1$ to measure impact of permuting x_1 on performance

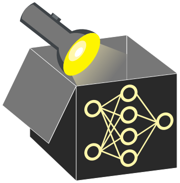
PERMUTATION FEATURE IMPORTANCE

$\color{red}{\mathcal{D}S_{(k)}}$

i	$\mathbf{x_S}$	$\mathbf{x_2}$	$\mathbf{x_3}$
1	2	4	7
⋮	1	5	8
n	3	6	9

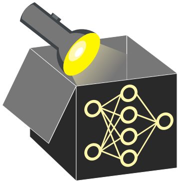
$\color{blue}{\mathcal{D}}$

$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$
1	4	7
2	5	8
3	6	9



1. **Perturbation:** Sample feature values from the distribution of x_S ($P(X_S)$).
   $\Rightarrow$ Randomly permute feature x_S
   $\Rightarrow$ Replace x_S with permuted feat. x_S and create data $\mathcal{D}S$ containing x_S

PERMUTATION FEATURE IMPORTANCE



$\mathcal{D}S_{(k)}$

i	$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	2	4	7
⋮	1	5	8
n	3	6	9

$\mathcal{D}$

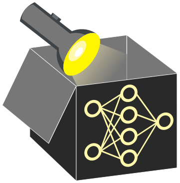
$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	7
2	5	8
3	6	9

$\hat{f}$
0.6
0.6
0.6

$\hat{f}$
0.4
0.8
0.6

1. **Perturbation:** Sample feature values from the distribution of x_S ($P(X_S)$).
   $\Rightarrow$ Randomly permute feature x_S
   $\Rightarrow$ Replace x_S with permuted feat. x_S and create data $\mathcal{D}S$ containing x_S
2. **Prediction:** Make predictions for both data, i.e., $\mathcal{D}$ and $\mathcal{D}S$

PERMUTATION FEATURE IMPORTANCE



$\mathcal{D}S_{(k)}$

i	x_S	x_2	x_3
1	2	4	7
⋮	1	5	8
n	3	6	9

$L(\hat{f}, y)$
0.9
0.5
0.1

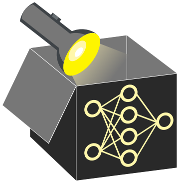
$\mathcal{D}$

x_1	x_2	x_3
1	4	7
2	5	8
3	6	9

$L(\hat{f}, y)$
0.25
0.35
0.1

3. **Aggregation:**
   - Compute the loss for each observation in both data sets

PERMUTATION FEATURE IMPORTANCE



$\mathcal{D}S_{(k)}$

i	$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	2	4	7
⋮	1	5	8
n	3	6	9

$\mathcal{D}$

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	4	7
2	5	8
3	6	9

ΔL
0.65
0.15
0

$L(\hat{f}, y)$		$L(\hat{f}, y)$
0.9		0.25
0.5	-	0.35
0.1		0.1

3. Aggregation:

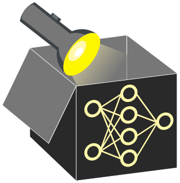
- Compute the loss for each observation in both data sets
- Take the difference of both losses ΔL for each observation

PERMUTATION FEATURE IMPORTANCE

$$\mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}S_{(k)}) - \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D})$$

i	$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	ΔL
1	2	4	7	1	4	7	0.65
⋮	1	5	8	2	5	8	0.15
n	3	6	9	3	6	9	0

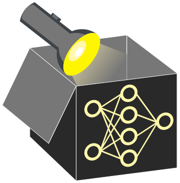
= 0.267



3. Aggregation:

- Compute the loss for each observation in both data sets
- Take the difference of both losses ΔL for each observation
- Average this change in loss across all observations
  Note: Same as computing $\mathcal{R}_{\text{emp}}$ on both data sets and taking difference

PERMUTATION FEATURE IMPORTANCE



$$\mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}S_{(k)}) - \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D})$$

	i	x_S	x_2	x_3	x_1	x_2	x_3	ΔL
1	1	2	4	7	1	4	7	0.65
	⋮	1	5	8	2	5	8	0.15
	n	3	6	9	3	6	9	0

= 0.267

⋮

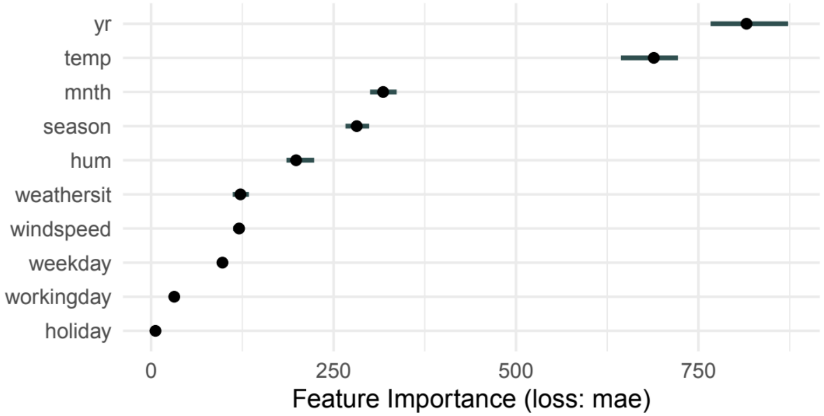
	i	x_S	x_2	x_3	x_1	x_2	x_3	ΔL
m	1	3	4	7	1	4	7	0.85
	⋮	2	5	8	2	5	8	0
	n	1	6	9	3	6	9	0.35

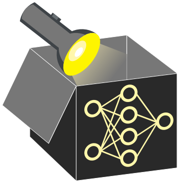
= 0.4

$\widehat{PFI}_S$ = ½ (0.267 + 0.4)

3. **Aggregation:**
   - Compute the loss for each observation in both data sets
   - Take the difference of both losses ΔL for each observation
   - Average this change in loss across all observations
   - Repeat perturbation and average over multiple repetitions

EXAMPLE: BIKE SHARING DATASET

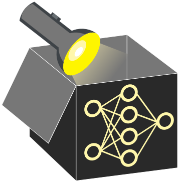




Interpretation:

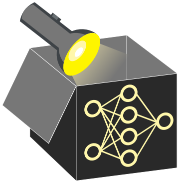
- `yr` and `temp` are most important feats using mean absolute error (MAE)
- Destroying info. about `yr` by permuting it increases MAE of model by 816
- Error bars show 5% and 95% quantiles over multiple permutations

COMMENTS ON PFI



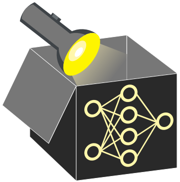
- Interpretation: Increase in error when feature's information is destroyed

COMMENTS ON PFI



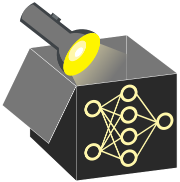
- Interpretation: Increase in error when feature's information is destroyed
- Results can be unreliable due to random permutations
  $\Rightarrow$ Solution: Average results over multiple repetitions

COMMENTS ON PFI



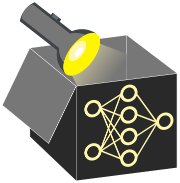
- Interpretation: Increase in error when feature's information is destroyed
- Results can be unreliable due to random permutations
  $\Rightarrow$ Solution: Average results over multiple repetitions
- Permuting features despite correlation/dependence with other features can lead to unrealistic combinations of feature values
  $\rightsquigarrow$ Extrapolation issue

COMMENTS ON PFI



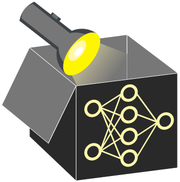
- Interpretation: Increase in error when feature's information is destroyed
- Results can be unreliable due to random permutations
  $\Rightarrow$ Solution: Average results over multiple repetitions
- Permuting features despite correlation/dependence with other features can lead to unrealistic combinations of feature values
  $\rightsquigarrow$ Extrapolation issue
- PFI automatically includes importance of interaction effects with other features
  $\Rightarrow$ Permuting x_j also destroys interactions with permuted feature
  $\Rightarrow$ PFI score contains importance of all interactions with permuted feature

COMMENTS ON PFI



- Interpretation: Increase in error when feature's information is destroyed
- Results can be unreliable due to random permutations
  $\Rightarrow$ Solution: Average results over multiple repetitions
- Permuting features despite correlation/dependence with other features can lead to unrealistic combinations of feature values
  $\rightsquigarrow$ Extrapolation issue
- PFI automatically includes importance of interaction effects with other features
  $\Rightarrow$ Permuting x_j also destroys interactions with permuted feature
  $\Rightarrow$ PFI score contains importance of all interactions with permuted feature
- Interpretation of PFI depends on whether training or test data is used

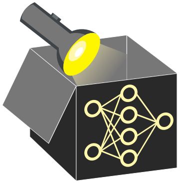
COMMENTS ON PFI - EXTRAPOLATION



Example: Let $y = x_3 + \epsilon_y$, with $\epsilon_y \sim \mathcal{N}(0, 0.1)$.

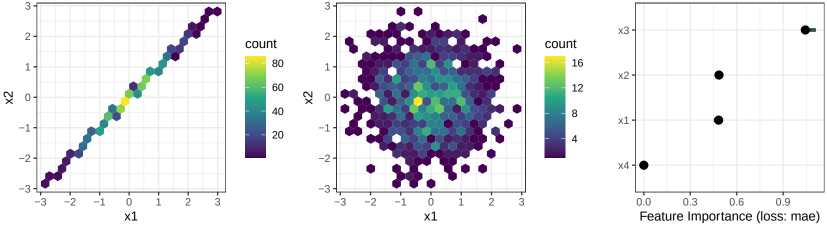
- $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$; highly correlated ($\epsilon_1 \sim \mathcal{N}(0, 1)$, $\epsilon_2 \sim \mathcal{N}(0, 0.01)$)
- $x_3 := \epsilon_3$, $x_4 := \epsilon_4$, with $\epsilon_3, \epsilon_4 \sim \mathcal{N}(0, 1)$; all noise terms ϵ_j are indep.
- Fitting a linear model yields $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$

COMMENTS ON PFI - EXTRAPOLATION



Example: Let $y = x_3 + \epsilon_y$, with $\epsilon_y \sim \mathcal{N}(0, 0.1)$.

- $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$; highly correlated ($\epsilon_1 \sim \mathcal{N}(0, 1)$, $\epsilon_2 \sim \mathcal{N}(0, 0.01)$)
- $x_3 := \epsilon_3$, $x_4 := \epsilon_4$, with $\epsilon_3, \epsilon_4 \sim \mathcal{N}(0, 1)$; all noise terms ϵ_j are indep.
- Fitting a linear model yields $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$

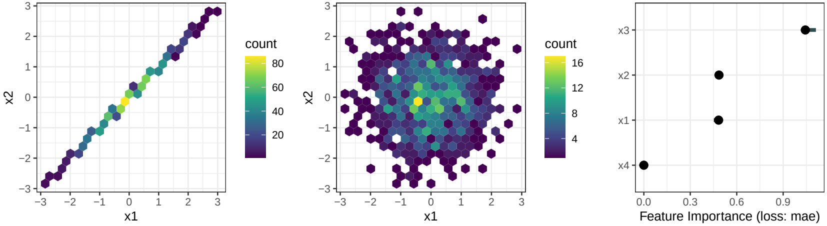


Hexbin plot of (x_1, x_2) before (left) and after (center) permuting x_1;
PFI scores (right).

COMMENTS ON PFI - EXTRAPOLATION

Example: Let $y = x_3 + \epsilon_y$, with $\epsilon_y \sim \mathcal{N}(0, 0.1)$.

- $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$; highly correlated ($\epsilon_1 \sim \mathcal{N}(0,1)$, $\epsilon_2 \sim \mathcal{N}(0, 0.01)$)
- $x_3 := \epsilon_3$, $x_4 := \epsilon_4$, with $\epsilon_3, \epsilon_4 \sim \mathcal{N}(0,1)$; all noise terms ϵ_j are indep.
- Fitting a linear model yields $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$



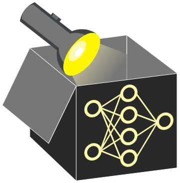
Hexbin plot of (x_1, x_2) before (left) and after (center) permuting x_1;
PFI scores (right).

$\Rightarrow$ x_1, x_2 cancel in $\hat{f}$ since $x_1 \approx x_2$, hence $0.3x_1 - 0.3x_2 \approx 0$
$\rightsquigarrow$ should be irrelevant

$\Rightarrow$ Permuting x_1 breaks joint structure $\rightsquigarrow$ unrealistic inputs

$\Rightarrow$ $PFI > 0$ due to extrapolation (PFI evaluates model on unrealistic inputs)
$\rightsquigarrow$ x_1, x_2 are misleadingly considered relevant

COMMENTS ON PFI - INTERACTIONS

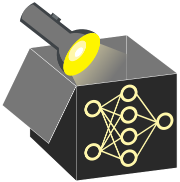


Example: Let $x_1, \ldots, x_4$ be independently and uniformly sampled from $\{-1, 1\}$ and

$$y := x_1 x_2 + x_3 + \epsilon_Y \text{ with } \epsilon_Y \sim N(0, 1)$$

Fitting a LM yields $\hat{f}(x) \approx x_1 x_2 + x_3$.

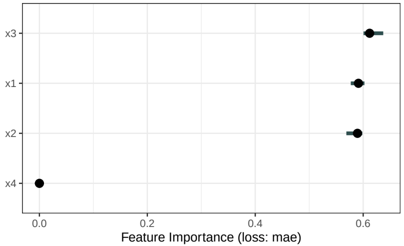
COMMENTS ON PFI - INTERACTIONS



Example: Let $x_1, \ldots, x_4$ be independently and uniformly sampled from $\{-1, 1\}$ and

$$y := x_1 x_2 + x_3 + \epsilon_Y \text{ with } \epsilon_Y \sim N(0, 1)$$

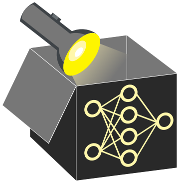
Fitting a LM yields $\hat{f}(x) \approx x_1 x_2 + x_3$.



Although x_3 alone contributes as much to the prediction as x_1 and x_2 jointly, all three are considered equally relevant.

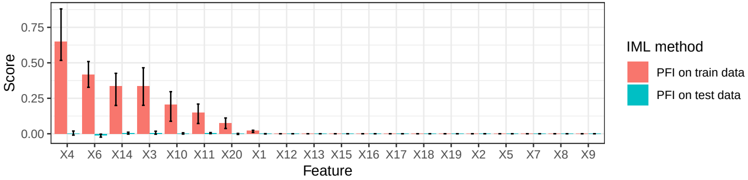
$\Rightarrow$ PFI does not fairly attribute the performance to the individual features.

COMMENTS ON PFI - TRAIN VS. TEST DATA



Example:

- $x_1, \ldots, x_{20}, y$ are independently sampled from $\mathcal{U}(-10, 10)$
- Train set: $n = 50$ (intentionally small) and large test set
- Model: `xgboost` with default settings (overfits strongly)



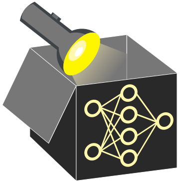
Observation:

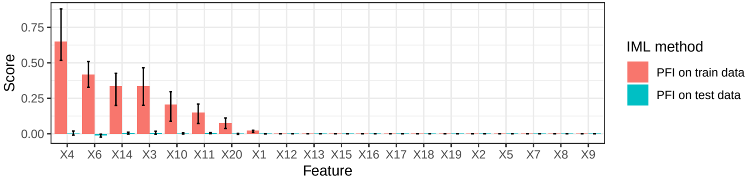
- PFI on train data highlights features that the model overfitted to.
- PFI on test data detects no relevant features.

COMMENTS ON PFI - TRAIN VS. TEST DATA

Example:

- $x_1, \ldots, x_{20}, y$ are independently sampled from $\mathcal{U}(-10, 10)$
- Train set: $n = 50$ (intentionally small) and large test set
- Model: `xgboost` with default settings (overfits strongly)



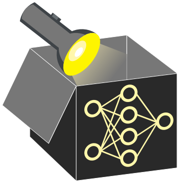


Observation:

- PFI on train data highlights features that the model overfitted to.
- PFI on test data detects no relevant features.

Why? *PFI* $\neq 0$ if permuting a feature breaks a dependency the model relies on. Model overfits due to spurious feature-target dependencies in train that vanish on test.
$\Rightarrow$ To find features that help the model to generalize, compute PFI on test data.

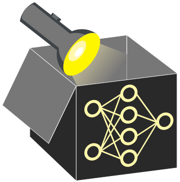
IMPLICATIONS OF PFI



Can we get insight into whether the ...

1. feature x_j is causal for the prediction?
   - $PFI_j \neq 0 \Rightarrow$ model relies on x_j
   - As the train vs. test data example shows, the converse does not hold

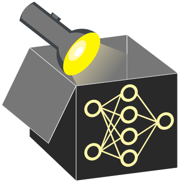
IMPLICATIONS OF PFI



Can we get insight into whether the ...

1. feature x_j is causal for the prediction?
   - $PFI_j \neq 0 \Rightarrow$ model relies on x_j
   - As the train vs. test data example shows, the converse does not hold
2. feature x_j contains prediction-relevant information?
   - $PFI_j \neq 0 \Rightarrow x_j$ is dependent on y, x_{-j}, or both (due to extrapolation)
   - x_j is not exploited by model (regardless of its usefulness for y) $\Rightarrow PFI_j = 0$

IMPLICATIONS OF PFI



Can we get insight into whether the ...

1. feature x_j is causal for the prediction?
   - $PFI_j \neq 0 \Rightarrow$ model relies on x_j
   - As the train vs. test data example shows, the converse does not hold
2. feature x_j contains prediction-relevant information?
   - $PFI_j \neq 0 \Rightarrow x_j$ is dependent on y, x_{-j}, or both (due to extrapolation)
   - x_j is not exploited by model (regardless of its usefulness for y) $\Rightarrow PFI_j = 0$
3. model requires access to x_j to achieve it's prediction performance?
   - As shown by the extrapolation example, such insight is not possible