

Interpretable Machine Learning

Permutation Feature Importance (PFI)

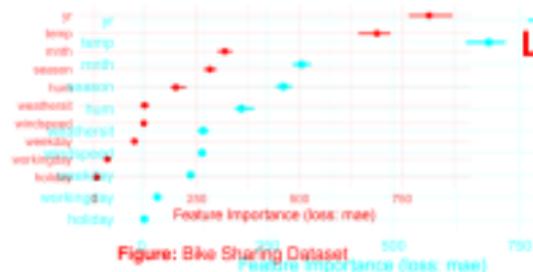
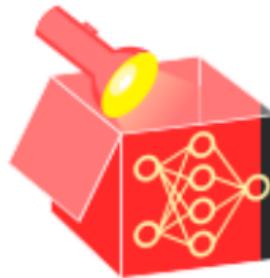


Figure: Bike Sharing Dataset
Feature Importance (loss: mae)

Figure: Bike Sharing Dataset

Learning goals

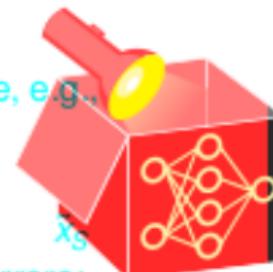
Learning goals

- Understand how PFI is computed
 - Understanding strengths and weaknesses
 - Testing Importance
- Understand how PFI is computed
 - Understanding strengths and weaknesses
 - Testing Importance

PERMUTATION FEATURE IMPORTANCE (PFI)

Breiman (2001) 1

Idea: "Destroy" feature of interest by permuting it so it becomes uninformative using informative, e.g., randomly permute obs b/w marginal distribution $P(x)$ (then $P(y)$ stays the same).
PFI for features s using tested data \mathcal{D} :



- Measure the error **without permuting feat.** and **with permuted feat. values \tilde{x}_s**
- Repeat permuting the feature (e.g., m times) and avg. the difference of both errors:

$$\widehat{PFI}_s = \frac{1}{m} \sum_{k=1}^m \widehat{PFI}_s = \frac{1}{m} \left(\sum_{k=1}^m \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}_{\tilde{x}(k)}) - \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) \right) \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) \text{ where } \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} L(\hat{f}(x), y)$$

where $\mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} L(\hat{f}(x), y)$

PERMUTATION FEATURE IMPORTANCE (PFI)

Breiman (2001) 1

Idea: "Destroy" feature of interest by permuting it so it becomes uninformative using informative, e.g., randomly permute obs b/w marginal distribution $P(x)$ (then $P(\tilde{x})$ stays the same).

PFI for feature x_S using test data \mathcal{D} :

- Measure the error **without permuting feat.** with \hat{f}

- Repeat permuting the feature (e.g., m times) and avg. the difference of both errors:

$$\widehat{PFI}_S = \frac{1}{m} \sum_{i=1}^m \widehat{PFI}_S = \frac{1}{m} \left(\sum_{k=1}^m \mathcal{R}_{\text{emp}}(\hat{f}, \tilde{\mathcal{D}}_S^{(k)}) - \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) \right) \text{ where } \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} L(\hat{f}(x), y)$$

The data \mathcal{D} where x_S is replaced with \tilde{x}_S is denoted as $\tilde{\mathcal{D}}_S$
where $\mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}) = \frac{1}{n} \sum_{(x,y) \in \mathcal{D}} L(\hat{f}(x), y)$

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

The data \mathcal{D} where x_S is replaced with \tilde{x}_S is denoted as $\tilde{\mathcal{D}}_S$.

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

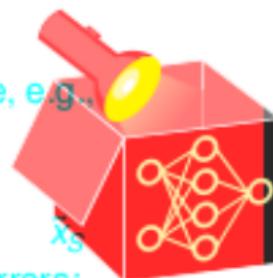
\mathcal{D}	$\tilde{\mathcal{D}}_S^{(1)}$	$\tilde{\mathcal{D}}_S^{(2)}$	$\tilde{\mathcal{D}}_S^{(3)}$	$\tilde{\mathcal{D}}_S^{(4)}$	$\tilde{\mathcal{D}}_S^{(5)}$	$\tilde{\mathcal{D}}_S^{(6)}$
x_1	1	4	7	2	1	3
x_2	4	5	8	5	2	5
x_3	7	8	1	8	3	7
x_4	1	4	7	2	4	7
x_5	2	5	8	5	3	6
x_6	5	8	1	5	2	6
x_7	8	1	4	8	5	9
x_8	1	4	7	3	4	7
x_9	4	7	2	6	3	4
x_{10}	7	8	5	7	6	9
x_{11}	2	5	8	3	5	8
x_{12}	5	8	1	6	9	1
x_{13}	8	1	4	9	2	6
x_{14}	1	4	7	5	3	7
x_{15}	4	7	2	6	8	9
x_{16}	7	8	5	7	6	9
x_{17}	2	5	8	3	5	8
x_{18}	5	8	1	6	9	1
x_{19}	8	1	4	9	2	6
x_{20}	1	4	7	5	3	7

Note: The S in x_S refers to a Subset of features for which we are interested in their effect on the prediction.

Here: We calculate the feature importance for one feature at a time $|S| = 1$.

Note: The S in x_S refers to a Subset of features for which we are interested in their effect on the prediction.

Here: We calculate the feature importance for one feature at a time $|S| = 1$.



PERMUTATION FEATURE IMPORTANCE

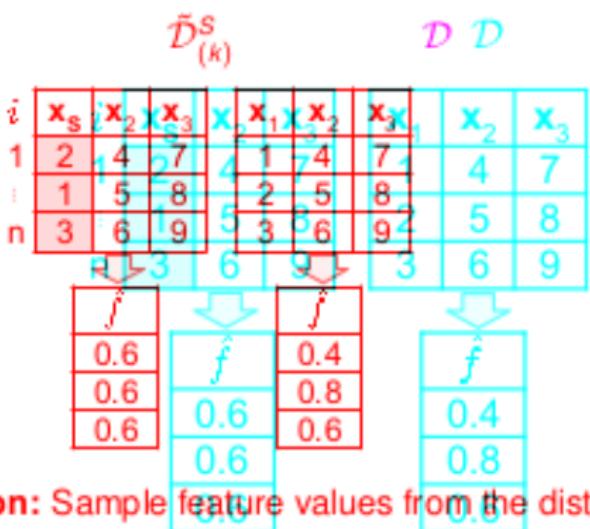
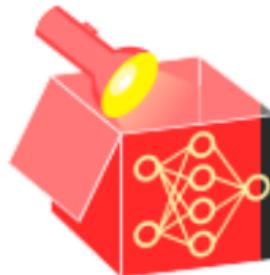
$\tilde{D}_{(k)}^S$ $D \ D$

i	x_s	\tilde{x}_s	x_2	x_3	x_1	\tilde{x}_1	x_3	\tilde{x}_3	x_1	x_2	x_3
1	2	14	27	4	1	74	71		4	7	
..	1	5	8	5	2	85	82		5	8	
n	3	6	9	3	6	9	9		6	9	
h	3	6	9	3	6	9	9		6	9	



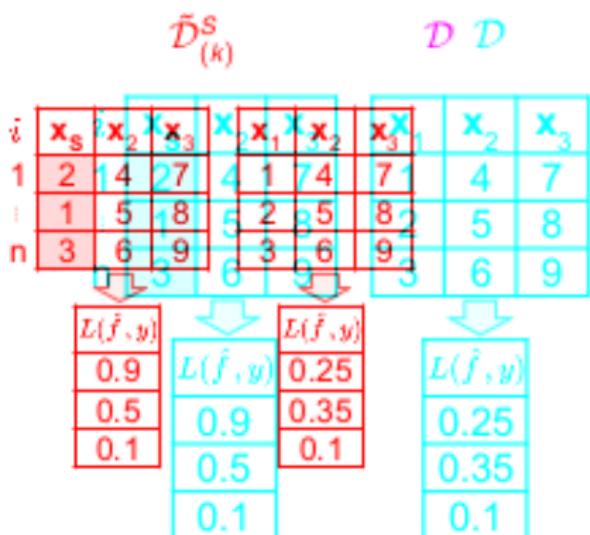
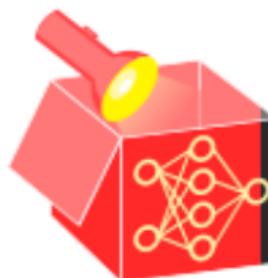
1. **Perturbation:** Sample feature values from the distribution of x_s ($P(X_s)$).
→ Randomly permute feature x_s .
1. **Perturbation:** Sample feature values from the distribution of x_s ($P(X_s)$).
→ Replace original feature with permuted feature \tilde{x}_s and create data \tilde{D}^S
containing \tilde{x}_s
→ Replace original feature with permuted feature \tilde{x}_s and create data \tilde{D}^S containing \tilde{x}_s

PERMUTATION FEATURE IMPORTANCE



1. **Perturbation:** Sample feature values from the distribution of x_s ($P(X_s)$).
→ Randomly permute feature x_s
1. **Perturbation:** Sample feature values from the distribution of x_s ($P(X_s)$).
→ Replace original feature with permuted feature \tilde{x}_s and create data $\tilde{\mathcal{D}}^s$ containing \tilde{x}_s
→ Randomly permute feature x_s
2. **Prediction:** Make predictions for both data, i.e., \mathcal{D} and $\tilde{\mathcal{D}}^s$ and create data $\tilde{\mathcal{D}}^s$ containing \tilde{x}_s
2. **Prediction:** Make predictions for both data, i.e., \mathcal{D} and $\tilde{\mathcal{D}}^s$

PERMUTATION FEATURE IMPORTANCE



3. Aggregation:

- Compute the loss for each observation in both data sets
 - Compute the loss for each observation in both data sets

PERMUTATION FEATURE IMPORTANCE

Diagram illustrating the computation of permutation feature importance:

The input data is shown in two tables: $\tilde{\mathcal{D}}_{(k)}^s$ and \mathcal{D} .

$\tilde{\mathcal{D}}_{(k)}^s$ (left) contains observations i (rows) and features $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$ (columns). The last column is labeled ΔL . The values are as follows:

i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	ΔL
1	2	14	27	4	1	74	71	0.45	7
2	1	5	18	5	2	85	82	0.55	8
n	3	6	9	3	6	9	3	0	9
h	3	6	9	6	9	3	3	0	9

\mathcal{D} (right) contains observations i (rows) and features $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$ (columns). The last column is labeled ΔL . The values are as follows:

i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	ΔL
1	2	14	27	4	1	74	71	0.45	7
2	1	5	18	5	2	85	82	0.55	8
n	3	6	9	3	6	9	3	0	9
h	3	6	9	6	9	3	3	0	9

Below the tables, three loss matrices are shown:

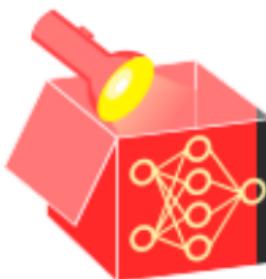
- $L(\hat{f}, y)$ (left):

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

0.9	-	0.25	-
-	-	-	-
-	-	-	-
- $L(\hat{f}, y)$ (right):

0.9	-	0.25	-
-	-	-	-
-	-	-	-

A red arrow points from the middle matrix to the right matrix, indicating the aggregation step where losses are combined.



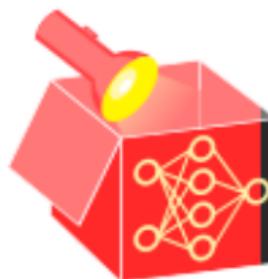
3. Aggregation:

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

PERMUTATION FEATURE IMPORTANCE

$$RR_{\text{emp}}(\hat{f}(\tilde{D}_{(k)}^S)) \rightarrow RR_{\text{emp}}(\hat{f}(D)D)$$

i	x_1	x_2	x_3	ΔL	x_1	x_2	x_3	ΔL
1	2	14	27	0.65	7	1	4	0.65
2	1	5	18	0.55	8	2	5	0.55
n	3	6	9	0	9	3	6	0



3. Aggregation:

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: This is equivalent to computing RR_{emp} on both data sets and taking the difference

PERMUTATION FEATURE IMPORTANCE

$$\mathcal{RP}_{\text{emp}}(\hat{f}(\tilde{D}_{(k)}^S)) \rightarrow \mathcal{RP}_{\text{emp}}(\hat{f}(D)) D$$

i	x_s	x_2	x_3	x_1	x_2	x_3	ΔL	x_3	ΔL	
1	2	14	27	4	1	74	71	0.65	7	0.65
2	1	5	18	5	2	35	82	0.45	8	= 0.267
n	3	6	9	3	6	93	0	9	0.15	
	1	3	7	1	4	7	0.65	7		
m	2	15	8	4	2	75	81	0	7	= 0.4
n	1	6	9	3	6	92	0.35	8	0.85	
	2	5	8	2	5	8	0	8		
n	1	6	9	3	6	9	0.35	9	0	

$$PFI_S = \frac{1}{2} (0.267 + 0.4)$$

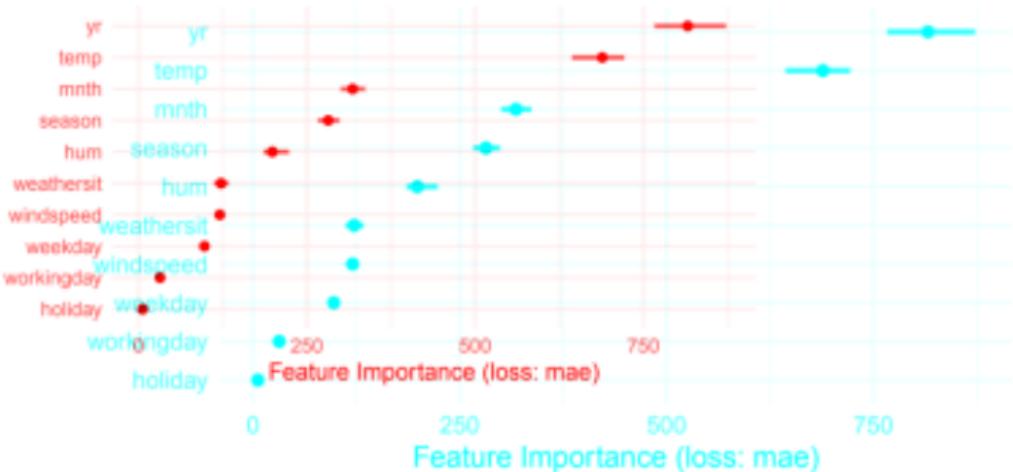


3. Aggregation:

3. Aggregate

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

EXAMPLE: BIKE SHARING DATASET



Interpretation:

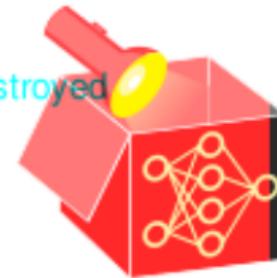
- Year (yr) and Temperature (temp) are most important features

Interpretation:

- Destroying information about yr by permuting it increases mean absolute error of model by 816
- Year (yr) and Temperature (temp) are most important features
- Destroying information about yr by permuting it increases mean absolute error of model by 816
- 5% and 95% quantile of repetitions due to multiple permutations are shown as error bars
- 5% and 95% quantile of repetitions due to multiple permutations are shown as error bars

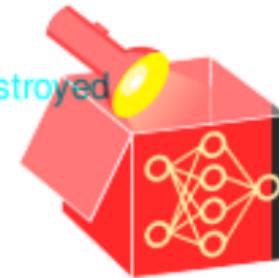
COMMENTS ON PFFI

- Interpretation PFFI is the increase of model error when feature's information is destroyed



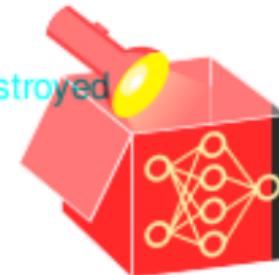
COMMENTS ON PFFI

- Interpretation: PFFI is the increase of model error when feature's information is destroyed
- Results can be unreliable due to random permutations
- Results can be unreliable due to random permutations
 - ⇒ Solution: Average results over multiple repetitions

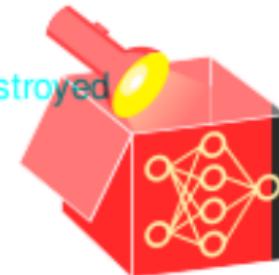


COMMENTS ON PFFI

- Interpretation: PFF is the increase of model error when feature's information is destroyed
- Results can be unreliable due to random permutations
 - Results can be unreliable due to random permutations
 - ⇒ Solution: Average results over multiple repetitions
 - Permuting features despite correlation with other features can lead to unrealistic combinations
 - Permuting features despite correlation with other features can lead to unrealistic combinations of feature values (since under dependence)
 $P(x_j, x_{-j}) \neq P(x_j)P(x_{-j}) \rightsquigarrow$ Extrapolation issue

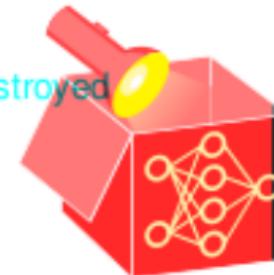


COMMENTS ON PFI



- Interpretation: PFI is the increase of model error when feature's information is destroyed
 - Results can be unreliable due to random permutations
- Results can be unreliable due to random permutations
 - ⇒ Solution: Average results over multiple repetitions
 - Permuting features despite correlation with other features can lead to unrealistic combinations
- Permuting features despite correlation with other features can lead to unrealistic combinations of feature values (since under dependence)
 - $P(x_i, x_{-i}) \neq P(x_i)P(x_{-i})$ ⇒ Extrapolation issue
 - ⇒ Permutation also destroys information of interactions where permuted feature is involved
- PFI automatically includes importance of interaction effects with other features
 - ⇒ Permutation also destroys information of interactions where permuted feature is involved
 - ⇒ Importance of all interactions with the permuted feature are contained in PFI score

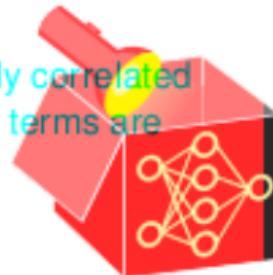
COMMENTS ON PFI



- Interpretation of PFI is the increase of model error when feature's information is destroyed
 - Results can be unreliable due to random permutations
- Results can be unreliable due to random permutations
 - ⇒ Solution: Average results over multiple repetitions
 - Permuting features despite correlation with other features can lead to unrealistic combinations
- Permuting features despite correlation with other features can lead to unrealistic combinations of feature values (since under dependence)
 - $P(x_i, x_{-i}) \neq P(x_i)P(x_{-i})$ ⇒ Extrapolation issue
 - ⇒ Permutation also destroys information of interaction effects with other features
- PFI automatically includes importance of interaction effects with other features
 - ⇒ Permutation also destroys information of interactions where permuted feature is involved
 - Interpretation of PFI depends on whether training or test data is used
 - ⇒ Importance of all interactions with the permuted feature are contained in PFI score
- Interpretation of PFI depends on whether training or test data is used

COMMENTS ON PFFI-EXTRAPOLATION

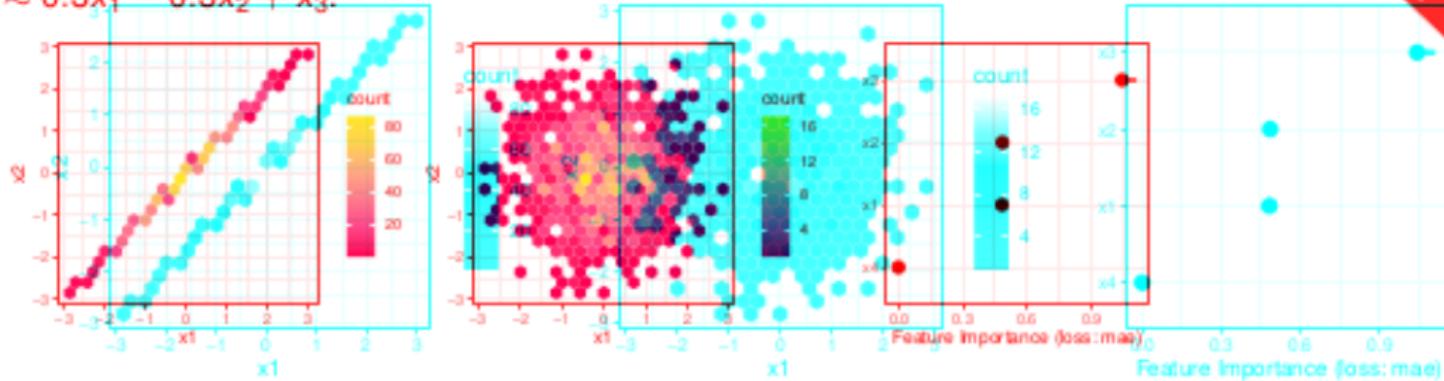
Example: Let $y = x_1 + x_2 + \epsilon_1$ with $\epsilon_1 \sim N(0, 1)$ where $x_1 = \epsilon_1$, $x_2 = \epsilon_2$ and ϵ_1, ϵ_2 are highly correlated ($\epsilon_1 \sim N(0, 1), \epsilon_2 \sim N(0, 0.01)$) and $x_3 = \epsilon_3, x_4 = \epsilon_4$ with $\epsilon_3, \epsilon_4 \sim N(0, 1)$. All noise terms are independent. Fitting a LM yields

$$\hat{y}(x) \approx 0.3x_1 - 0.3x_2 + x_3.$$


COMMENTS ON PFI-EXTRAPOLATION

Example: Let $y = x_3 + x_2 \epsilon_1 + x_1 \epsilon_2$ with $\epsilon_i \sim N(0, 1)$ where $x_1 = x_2 = x_3 = x_1 + x_2$ are highly correlated ($\epsilon_1 \sim N(0, 1)$, $\epsilon_2 \sim N(0, 0.01)$) and $x_3 = \epsilon_3$, with $\epsilon_3 \sim N(0, 1)$. All noise terms are independent. Fitting a LM yields

$$\hat{y}(x) \approx 0.3x_1 - 0.3x_2 + x_3.$$



Hexbin plot of x_1, x_2 before permuting x_1 (left), after permuting x_1 (center), and PFI

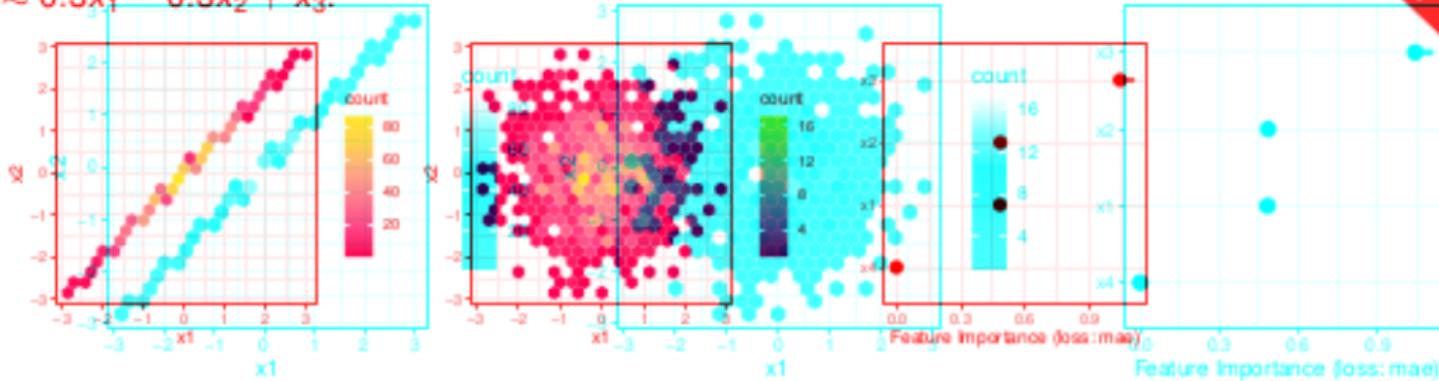
Hexbin plot of x_1, x_2 before permuting x_1 (left), after permuting x_1 (center), and PFI scores (right)



COMMENTS ON PFI-EXTRAPOLATION

Example: Let $y = x_3 + x_1 + \epsilon_1$ with $\epsilon_1 \sim N(0, 0.1)$ where $x_1 = x_1 + \epsilon_1$, $x_2 = x_2 + \epsilon_2$ are highly correlated ($\epsilon_1 \sim N(0, 1)$, $\epsilon_2 \sim N(0, 0.01)$) and $x_3 = \epsilon_3$, with $\epsilon_3 \sim N(0, 1)$. All noise terms are independent. Fitting a LM yields

$$\hat{y}(x) \approx 0.3x_1 - 0.3x_2 + x_3.$$



Hexbin plot of x_1, x_2 before permuting x_1 (left), after permuting x_1 (center), and PFI scores (right) $\Rightarrow x_1$ and x_2 should be irrelevant for the prediction $\hat{y}(x)$ for $\{x : P(x) > 0\}$

Hexbin plot of x_1, x_2 before permuting x_1 (left), after permuting x_1 (center), and PFI scores (right) $\Rightarrow x_1$ and x_2 should be irrelevant for the prediction $\hat{y}(x)$ for $\{x : P(x) > 0\}$

$\{x : P(x) > 0\}$ as $0.3x_1 - 0.3x_2 \approx 0$

\Rightarrow PFI evaluates model on unrealistic obs. outside $P(x) \approx 0$: x_1, x_2 are considered relevant (PFI > 0)

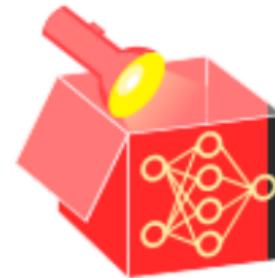
relevant (PFI > 0)

COMMENTS ON PFFI-INTERACTIONS

Example: Let x_1, x_2, \dots, x_4 be independently and uniformly sampled from $\{0, 1\}$ and

$$y := x_1 x_2 + x_3 + x_4 \text{ with } \epsilon_y \sim N(0, 1), \epsilon_y \sim N(0, 1)$$

Fitting a LM yields $\hat{f}(x) \approx x_1 x_2 + x_3 + x_4$.



COMMENTS ON PFF-INTERACTIONS

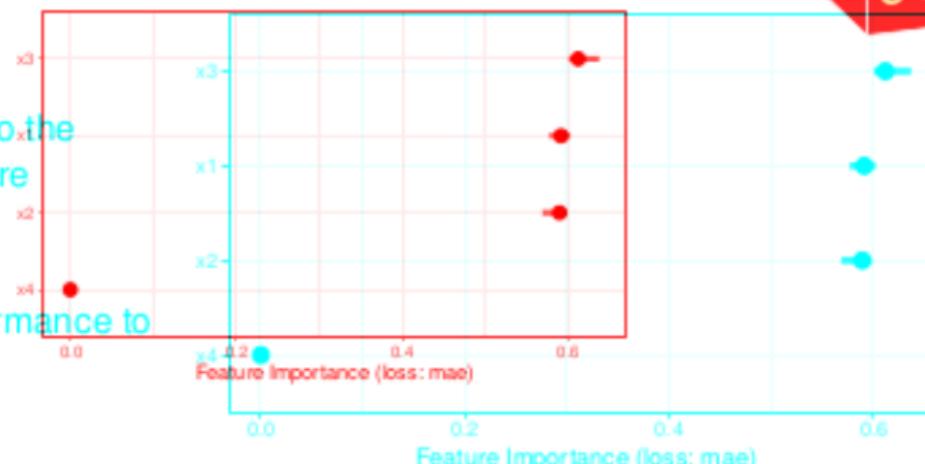
Example: Let x_1, x_2, \dots, x_4 be independently and uniformly sampled from $\{-1, 1\}$ and

$$y := x_1 x_2 + x_3 + x_4 \text{ with } \epsilon_y \sim N(0, 1), \epsilon_y \sim N(0, 1)$$

Fitting a LM yields $\hat{y}(x) \approx x_1 x_2 + x_3 + x_4$.

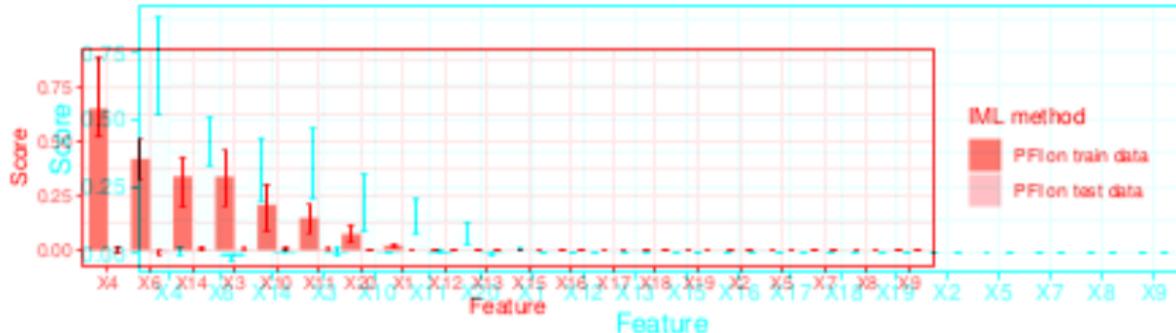
Although x_3 alone contributes as much to the prediction as x_1 and x_2 jointly, all three are considered equally relevant.

⇒ PFF does not fairly attribute the performance to performance to the individual features.



COMMENTS ON PFI - TEST VS TRAINING DATA

Example: x_1, \dots, x_{20}, y are independently sampled from $\mathcal{U}(-10, -10)$. An xgboost model with default hyperparameters is fit on a small training set of 50 observations. The model overfits heavily.



IML method
PFI on train data
PFI on test data

Figure: While PFI on test data considers all features to be irrelevant, PFI on train

Figure: While PFI on test data considers all features to be irrelevant, PFI on train data exposes the features on which the model overfitted.
features on which the model overfitted.

COMMENTS ON PFI - TEST VS TRAINING DATA

Example: x_1, \dots, x_{20}, y are independently sampled from $\mathcal{U}(-10, 10)$. An xgboost model with default hyperparameters is fit on a small training set of 50 observations. The model overfits heavily.

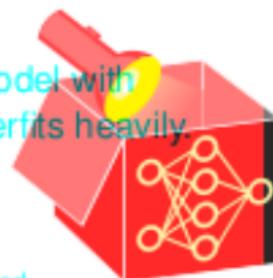
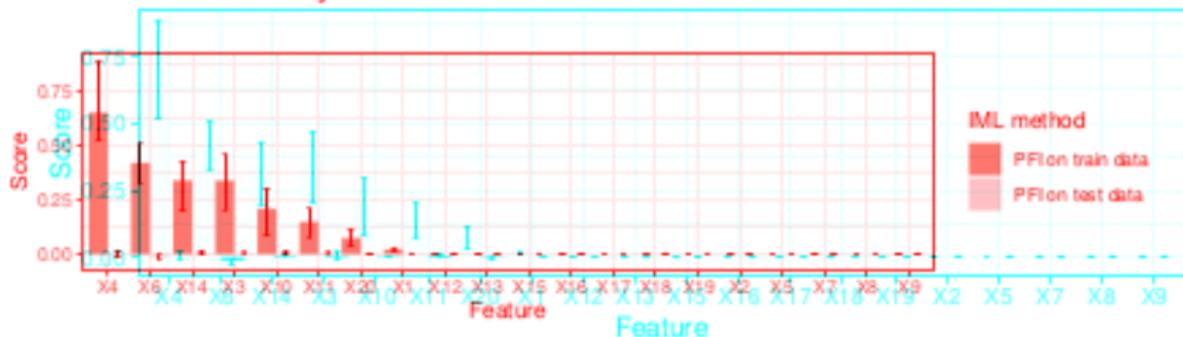


Figure: While PFI on test data considers all features to be irrelevant, PFI on train data exposes the features on which the model overfitted.

Why? PFI can only be nonzero if the permutation breaks a dependence in the data.

Why? PFI can only be nonzero if the permutation breaks a dependence in the data. Spurious correlations help the model perform well on train data but are not present in the test data.

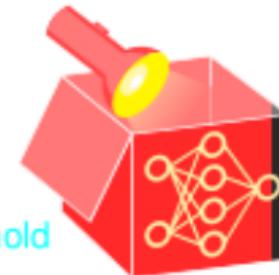
⇒ If you are interested in which features help the model to generalize, apply PFI on test data.

IMPLICATIONS OF PFI

Can we get insight into whether the ...

- feature x_j is causal to the prediction?

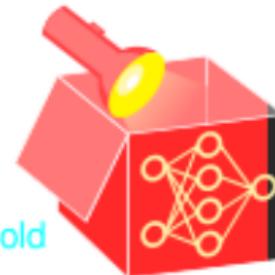
- $PFI \neq 0 \Rightarrow$ model relies on x_j
- As the training vs test example demonstrates, the converse does not hold



IMPLICATIONS OF PFI

Can we get insight into whether the the ...

- ➊ feature x_j is causal for the prediction?
 - $PFI_j \neq 0 \Rightarrow$ model relies on x_j
 - As the training vs test data example demonstrates, the converse does not hold
- ➋ feature x_j contains prediction-relevant information?
 - $PFI_j \neq 0 \Rightarrow x_j$ is dependent of y or its covariates x_{-j} or both (due to extrapolation)
 - PFI_j is not explicit, depends on y or its covariates x_{-j} or both (due to y or not) $\Rightarrow PFI_j = 0$ extrapolation)
 - x_j is not exploited by model (regardless of whether it is useful for y or not)
 $\Rightarrow PFI_j = 0$



IMPLICATIONS OF PFI

Can we get insight into whether the ...

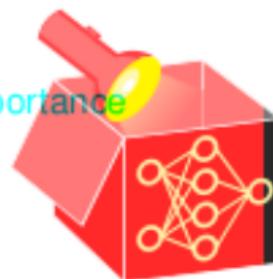
- ➊ feature x_j is causal to the prediction?
 - $PFI_j \neq 0 \Rightarrow$ model relies on x_j
 - As the training vs test data example demonstrates, the converse does not hold
- ➋ feature x_j contains prediction-relevant information?
 - $PFI_j \neq 0 \Rightarrow x_j$ is dependent of y or its covariates x_{-j} or both (due to extrapolation)
 - $PFI_j \neq 0 \Rightarrow x_j$ is dependent of y or its covariates x_{-j} or both (due to y or not) $\Rightarrow PFI_j = 0$
- ➌ model requires access to x_j to achieve it's prediction performance?
 - x_j is not exploited by model (regardless of whether it is useful for y or not)
 $\Rightarrow PFI_j = 0$
- ➍ model requires access to x_j to achieve it's prediction performance?
 - As the extrapolation example demonstrates, such insight is not possible
 $\Rightarrow PFI_j = 0$



TESTING IMPORTANCE (PIMP)

Altman et al (2010)

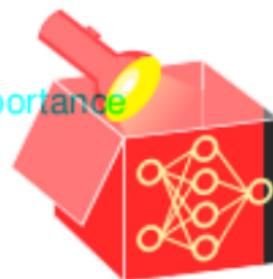
- PIMP was originally introduced for random forests' built-in permutation feature importance



TESTING IMPORTANCE (PIMP)

Altmaan et al (2010)

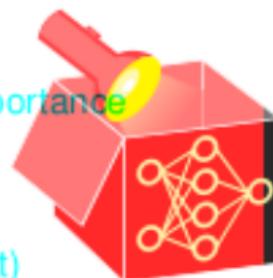
- PIMP was originally introduced for random forests built in permutation feature importance
- PIMP investigates whether the PFI score **significantly** differs from 0
- PIMP investigates whether the PFI score **significantly** differs from 0
⇒ Useful because PFI can be non-zero due to stochasticity



TESTING IMPORTANCE (PIMP)

Altmaier et al (2010)

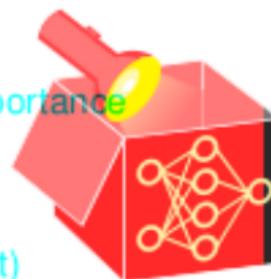
- PIMP was originally introduced for random forests built in permutation feature importance
- PIMP investigates whether the PFI score **significantly** differs from 0
- PIMP investigates whether the PFI score **significantly** differs from 0
 - ⇒ Useful because PFI can be non-zero due to stochasticity
- PIMP tests the H_0 -hypothesis: Feature is independent of the target y (unimportant)
- PIMP tests the H_0 -hypothesis: Feature is independent of the target y (unimportant)

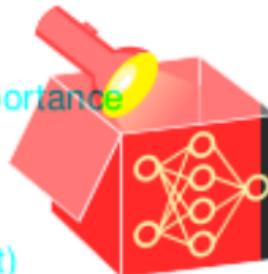


TESTING IMPORTANCE (PIMP)

Altmaier et al (2010)

- PIMP was originally introduced for random forests built in permutation feature importance
- PIMP investigates whether the PFI score significantly differs from 0
- PIMP investigates whether the PFI score significantly differs from 0
 - ⇒ Useful because PFI can be non-zero due to stochasticity
- PIMP tests the H_0 -hypothesis: Feature is independent of the target y (unimportant)
 - Sampling under H_0 : Permute target y , retrain model, compute PFI scores (repeat)
⇒ Permuting y breaks relationship to all features
- Sampling under H_0 : Permute target y , retrain model, compute PFI scores under H_0
(repeat)
 - ⇒ Permuting y breaks relationship to all features
 - ⇒ By computing PFI scores again, we obtain distribution of PFI scores under H_0





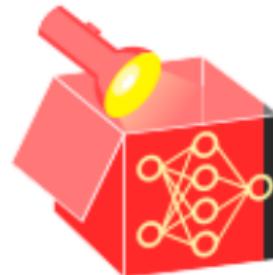
- PIMP was originally introduced for random forests built in permutation feature importance
- PIMP investigates whether the PFI score **significantly** differs from 0
- PIMP investigates whether the PFI score **significantly** differs from 0
 - Useful because PFI can be non-zero due to stochasticity
- PIMP tests the H_0 -hypothesis: Feature is independent of the target y (unimportant)
- Sampling under H_0 : Permute target y , retrain model, compute PFI scores (repeat) (unimportant)
 - Permuting y breaks relationship to all features
- Sampling under H_0 : Permute target y , retrain model, compute PFI scores under H_0 (repeat)
 - Compute p-value - the tail probability under H_0 - and use it as a new importance measure
 - Permuting y breaks relationship to all features
 - By computing PFI scores again, we obtain distribution of PFI scores under H_0
- Compute p-value - the tail probability under H_0 - and use it as a new importance measure

TESTING IMPORTANCE (PIMP)

PIMP algorithm:

- 1 **For** $m \in \{1, \dots, n_{\text{repetitions}}\}$ **do**:

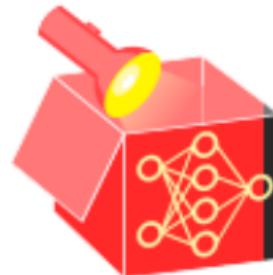
 - Permute response vector y
 - Retrain model with data \mathbf{X} and permuted y
 - Compute feature importance P_j/P_j^* for each feature j under H_0



TESTING IMPORTANCE (PIMP)

PIMP algorithm:

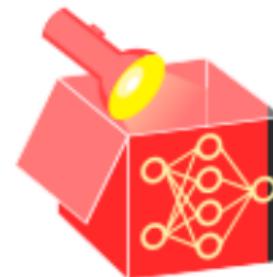
- ➊ For $m \in \{1, \dots, n_{\text{repetitions}}\}$:
 - Permute response vector y
 - Retrain model with data X and permuted y
 - Compute feature importance P_j^m / P_j^0 for each feature j under H_0
- ➋ Train model with X and unpermuted y



TESTING IMPORTANCE (PIMP)

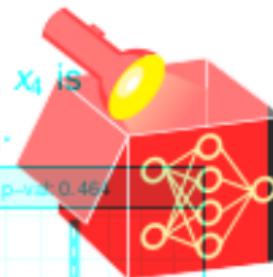
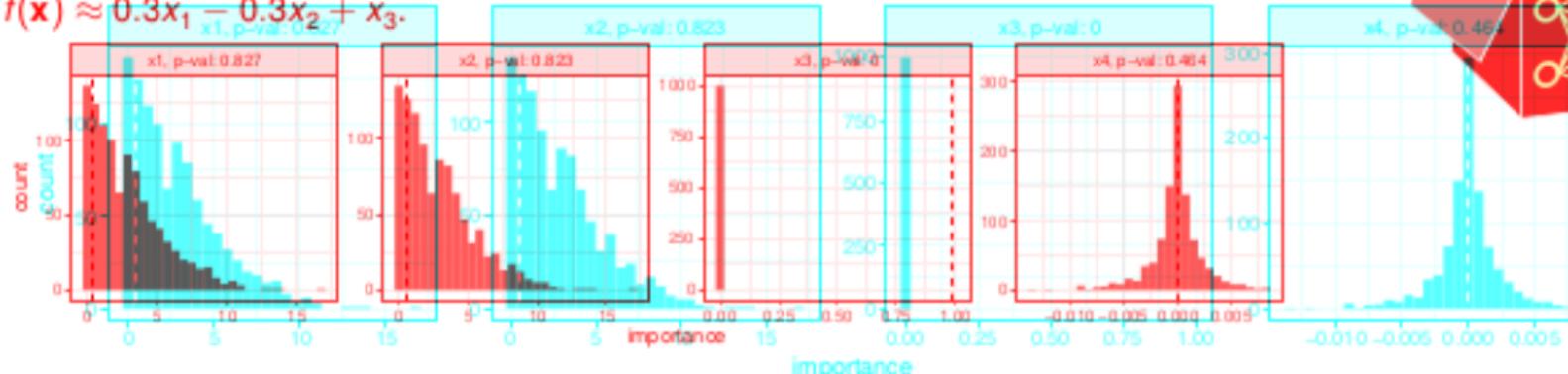
PIMP algorithm:

- ① For $m \in \{1, \dots, n_{\text{repetitions}}\}$:
 - Permute response vector y
 - Retrain model with data \mathbf{X} and permuted y
 - Compute feature importance PFI_j^m for each feature j (under H_0)
- ② Train model with \mathbf{X} and unpermuted y
- ③ For each feature $j \in \{1, \dots, p\}; p\}$:
 - Fit probability distribution of the feature importance values PFI_j^m , $m \in \{1, \dots, n_{\text{repetitions}}\}$ (m choice between $n_{\text{repetitions}}$) (choice between Gaussian, lognormal, gamma) or non-parametric
 - Compute feature importance PFI_j for the model without permutation of y (under H_1)
 - Compute the p-value of PFI_j based on the fitted distribution
 - Retrieve the p-value of PFI_j based on the fitted distribution



PIMP FOR EXTRAPOLATION EXAMPLE

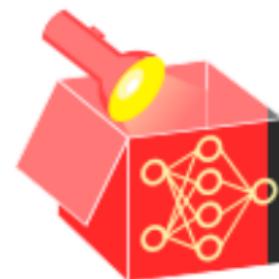
Recall: $y = x_3 x_4 + \epsilon_y$ with $\epsilon_y \sim N(0, 0.01)$, x_1, x_2 highly correlated but independent of y , x_4 is independent of y and all other variables. Fitting a Lasso yields $\hat{f}(x) \approx 0.3x_1 - 0.3x_2 + x_3$.
 $\hat{f}(x) \approx 0.3x_1 - 0.3x_2 + x_3$.



- Histograms: H_0 distribution of PFI scores after permuting y (1000 repetitions)
- Histograms: H_0 distribution of PFI scores after permuting y (1000 repetitions)
- Red: PFI score estimated on unpermuted y (under H_1) \rightsquigarrow compare against H_0 distribution
- Red: PFI score estimated on unpermuted y (under H_1) \rightsquigarrow compare against H_0 distribution
- Results: Although PFI for x_1 and x_2 is nonzero (red), PIMP considers them not significantly relevant ($p\text{-value} > 0.05$)

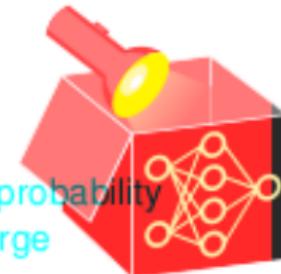
▶ Romano et al. (2010)

- When should we reject the H_0 -hypothesis for a feature?
- When should we reject the H_0 -hypothesis for a feature?



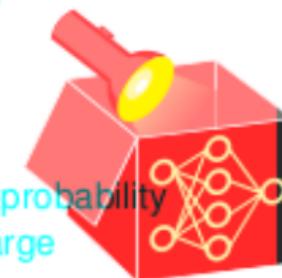
▶ Romano et al. (2010)

- When should we reject the H_0 -hypothesis for a feature?
- When should we reject the H_0 -hypothesis for a feature?
 - The larger the number of features, the more tests need to be performed by PIMP
 - The larger the number of features, if the multiple tests need to be performed by PIMP, the probability that the multiple testing problem of multiplicity of tests (is not taken) into account, the probability that some of the true H_0 -hypothesis is rejected (type-I error) by chance may be large



▶ Romano et al. (2010)

- When should we reject the H_0 -hypothesis for a feature?
- When should we reject the H_0 -hypothesis for a feature?
 - The larger the number of features, the more tests need to be performed by PIMP
 - The larger the number of features, the more tests need to be performed by PIMP, the probability that the multiple testing problem of multiplicity of tests (is not taken) into account, the probability that some of the true H_0 -hypothesis is rejected (type-I error) by Accounting for multiplicity of individual tests can be achieved by controlling an appropriate error rate, e.g., the family-wise error rate (FWE: probability of at least one type-I error)
- Accounting for multiplicity of individual tests can be achieved by controlling an appropriate error rate, e.g., the family-wise error rate (FWE: probability of at least one type-I error)



▶ Romano et al. (2010)

- When should we reject the H_0 -hypothesis for a feature?
- When should we reject the H_i -hypothesis for a feature?
- The larger the number of features, the more tests need to be performed by PIMP
- The larger the number of features, the more tests need to be performed by PIMP, the probability that the multiple testing problem of multiplicity of tests (is not taken) into account, the probability that some of the true H_i -hypothesis is rejected (type-I error) by chance may be large
- Accounting for multiplicity of individual tests can be achieved by controlling an appropriate error rate, e.g., the family-wise error rate (FWE: probability of at least one type-I error)
- Accounting for multiplicity of individual tests can be achieved by controlling an appropriate error rate, e.g., the Bonferroni correction which rejects a null hypothesis if its p-value is smaller than α/m with m as the number of performed parallel tests at least one type-I error)
- One classical method to control the FWE is the Bonferroni correction which rejects a null hypothesis if its p-value is smaller than α/m with m as the number of performed parallel tests

