Interpretable Machine Learning

Permutation Feature Importance (PFI)

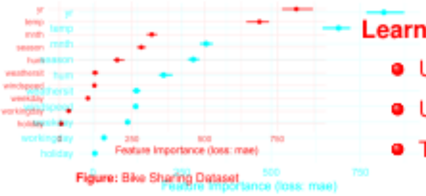


Figure: Bike Sharing Dataset

Learning goals

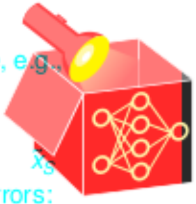
- Understand how PFI is computed
- Understanding strengths and weaknesses
- Testing Importance

Learning goals

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

Figure: Bike Sharing Dataset

PERMUTATION FEATURE IMPORTANCE (PFI) ▸ Breiman (2001)



Idea: "Destroy" feat. of interest x_j by perturbing it s.t. it becomes uninformative, e.g., randomly permute obs. in x_j (marginal distribution $\mathbb{P}(x_j)$ stays the same).

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

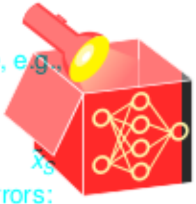
- Measure the error **without permuting feat.** and **with permuted feat. values** $\tilde{x}_S$
- Repeat permuting the feat. (e.g., m times) and avg. the difference of both errors:

$$\widehat{PFI}_S = \frac{1}{m}\sum_{k=1}^{m} \mathcal{R}_{\text{emp}}(\hat{f}, \tilde{\mathcal{D}}_{(k)}^{S}) - \mathcal{R}_{\text{emp}}(\hat{f}, \mathcal{D}),$$

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

Interpretable Machine Learning – 1 / 12

PERMUTATION FEATURE IMPORTANCE (PFI) ▸ Breiman (2001)



Idea: "Destroy" feat. of interest x_j by perturbing it s.t. it becomes uninformative, e.g., randomly permute obs. in x_j (marginal distribution $\mathbb{P}(x_j)$ stays the same).

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

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

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

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 $\tilde{\mathcal{D}}^S$.

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

$\mathcal{D}$

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

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

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

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

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

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

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

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

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

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

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

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

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

Note: The S in x_S refers to a **S**ubset 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

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

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

$\mathcal{D}$ $\mathcal{D}$

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
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 original feature with permuted feature $\tilde{x}_S$ and create data $\tilde{\mathcal{D}}^S$ containing $\tilde{x}_S$

1. **Perturbation:** Sample feature values from the distribution of x_S ($P(X_S)$).

$\Rightarrow$ Randomly permute feature x_S

$\Rightarrow$ Replace original feature with permuted feature $\tilde{x}_S$ and create data $\tilde{\mathcal{D}}^S$ containing $\tilde{x}_S$

PERMUTATION FEATURE IMPORTANCE

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

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

$\hat{f}$
0.6
0.6
0.6

$\mathcal{D}$

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

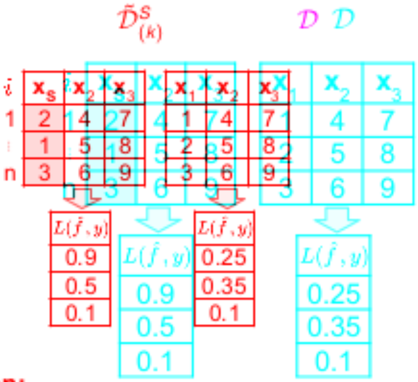
$\hat{f}$
0.4
0.8
0.6

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

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

Interpretable Machine Learning – 2 / 12

PERMUTATION FEATURE IMPORTANCE

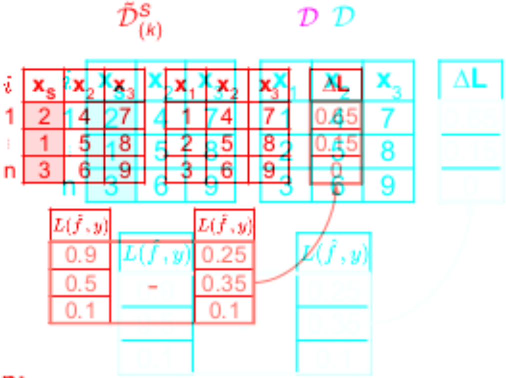


3. **Aggregation:**
   - Compute the loss for each observation in both data sets

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

Interpretable Machine Learning – 2 / 12

PERMUTATION FEATURE IMPORTANCE



3. Aggregation:

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

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

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

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: This is equivalent to computing $\mathcal{R}_{\text{emp}}$ on both data sets and taking the difference

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 $\mathcal{R}_{\text{emp}}$ on both data sets and taking the difference

Interpretable Machine Learning – 2 / 12

PERMUTATION FEATURE IMPORTANCE

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

1

i	x_S	x_2	x_3	x_1	x_2	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

⋮

m

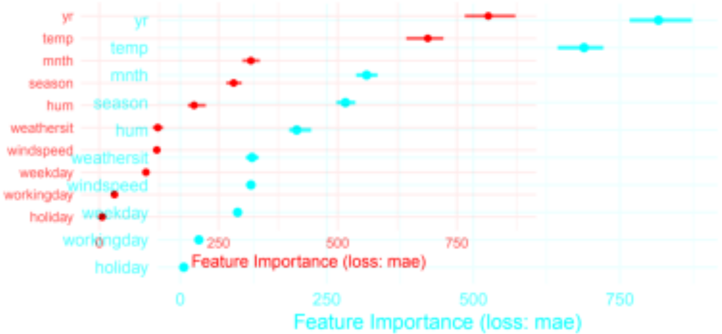
i	x_S	x_2	x_3	x_1	x_2	x_3	ΔL
1	3	4	7	1	4	7	0.85
⋮	2	5	8	2	5	8	0
n	1	6	9	3	6	9	0.35

= 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:

- 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 multiple permutations are shown as error bars

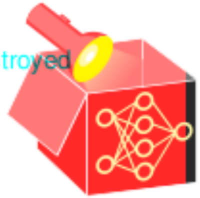
Interpretable Machine Learning – 3 / 12

COMMENTS ON PFI

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

Interpretable Machine Learning – 4 / 12

COMMENTS ON PFI

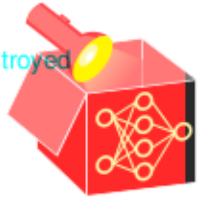


- Interpretation: PFI is the increase of model error when feature's information is destroyed
- Results can be unreliable due to random permutations

  ⇒ Solution: Average results over multiple repetitions

Interpretable Machine Learning – 4 / 12

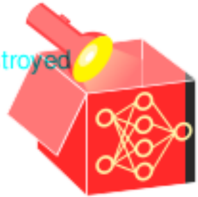
COMMENTS ON PFI



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

Interpretable Machine Learning – 4 / 12

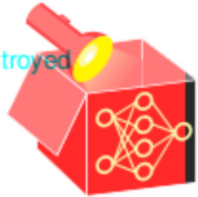
COMMENTS ON PFI



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

- Interpretation: PFI is the increase of model 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 with other features can lead to unrealistic combinations of feature values (since under dependence $\mathbb{P}(x_j, x_{-j}) \neq \mathbb{P}(x_j)\mathbb{P}(x_{-j})$) $\rightsquigarrow$ Extrapolation issue
- PFI automatically includes importance of interaction effects with other features
  $\Rightarrow$ Permutation also destroys information of interactions where permuted feature is involved
  $\Rightarrow$ Importance of all interactions with the permuted feature are contained in PFI score

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
  ⇒ Solution: Average results over multiple repetitions
- Permuting features despite correlation with other features can lead to unrealistic combinations of feature values (since under dependence $\mathbb{P}(x_j, x_{-j}) \neq \mathbb{P}(x_j)\mathbb{P}(x_{-j})$) ⇝ Extrapolation issue
- 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
- Interpretation of PFI depends on whether training or test data is used

Interpretable Machine Learning – 4 / 12

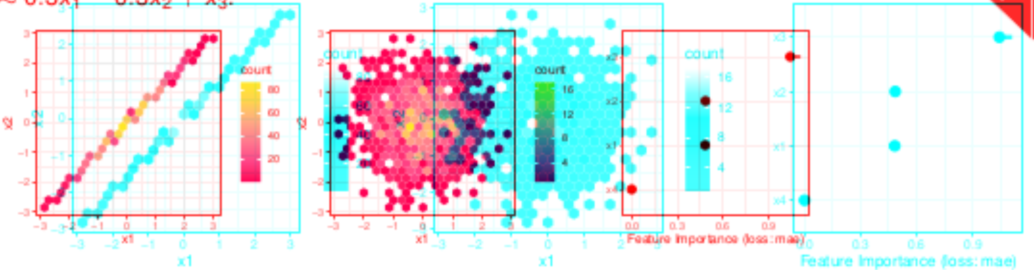
COMMENTS ON PFI - EXTRAPOLATION

Example: Let $y = x_3 + \epsilon_y$ with $\epsilon_y \sim N(0, 0.1)$ where $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$ 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 $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$.

Interpretable Machine Learning – 5 / 12

COMMENTS ON PFI - EXTRAPOLATION

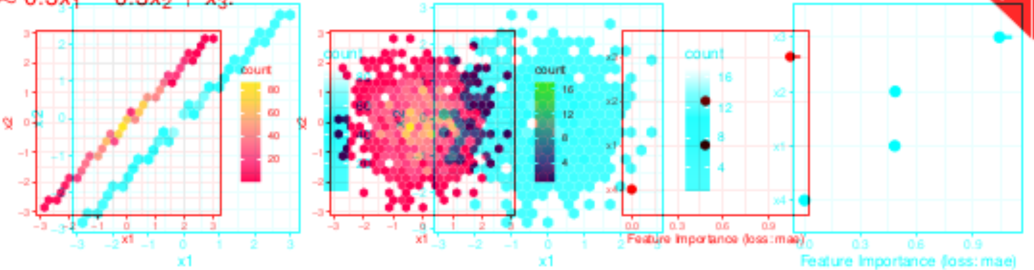
Example: Let $y = x_3 + \epsilon_y$ with $\epsilon_y \sim N(0, 0.1)$ where $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$ are highly correlated ($\epsilon_1 \sim N(0, 1)$, $\epsilon_2 \sim N(0, 0.01)$) and $x_3 := \epsilon_3$, $x_4 := \epsilon_4$, with $\epsilon_3, \epsilon_4 \sim N(0, 1)$. All noise terms are independent. Fitting a LM yields $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$.



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

COMMENTS ON PFI - EXTRAPOLATION

Example: Let $y = x_3 + \epsilon_y$ with $\epsilon_y \sim N(0, 0.1)$ where $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$ 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 $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$.



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 $\hat{f}(\mathbf{x})$ for $\{\mathbf{x} : \mathbb{P}(\mathbf{x}) > 0\}$ as $0.3x_1 - 0.3x_2 \approx 0$

$\Rightarrow$ PFI evaluates model on unrealistic obs. outside $\mathbb{P}(\mathbf{x})$ $\rightsquigarrow$ x_1, x_2 are considered relevant (PFI > 0)

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 $\hat{f}(x) \approx x_1 x_2 + x_3$.

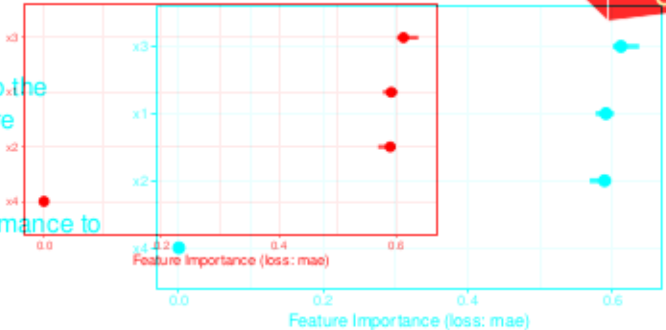
Interpretable Machine Learning – 6 / 12

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.

Interpretable Machine Learning – 6 / 12

COMMENTS ON PFI - TEST VS. TRAINING DATA

Example: $x_1, \ldots, x_{20}, y$ 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.

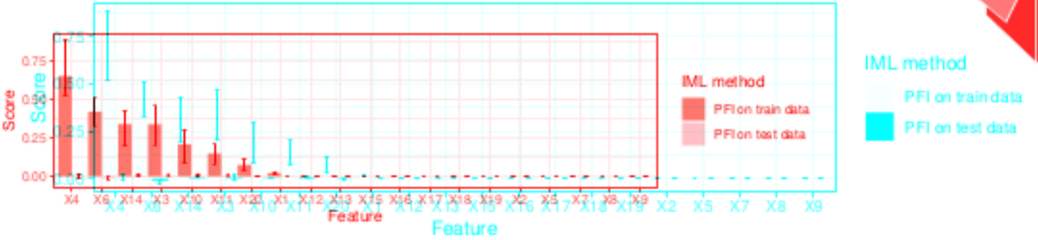
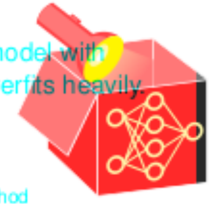


Figure: While PFI on test data considers all features to be irrelevant, PFI on train data exposes the features on which the model overfitted.

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.

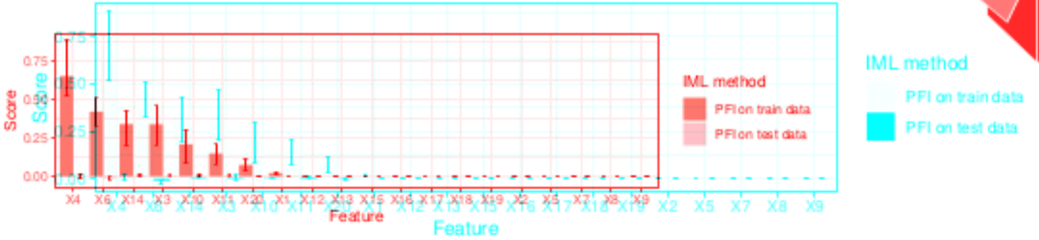
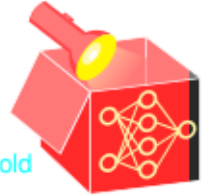


Figure: While PFI on test data considers all features to be irrelevant, PFI on train data exposes the features on which the model overfitted.

Why? PFI can only be nonzero if the permutation breaks a dependence in the data. Spurious correlations help the model perform well on train data but are not present in the test data.
$\Rightarrow$ If you are interested in which features help the model to generalize, apply PFI on test data.

Interpretable Machine Learning – 7 / 12

IMPLICATIONS OF PFI

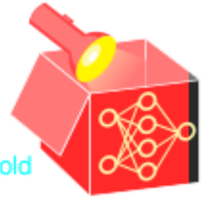


Can we get insight into whether the ...

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

Interpretable Machine Learning – 8 / 12

IMPLICATIONS OF PFI

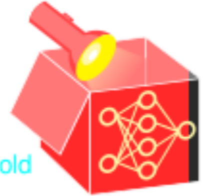


Can we get insight into whether the ...

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

Interpretable Machine Learning – 8 / 12

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

Interpretable Machine Learning – 8 / 12

TESTING IMPORTANCE (PIMP) Altmann et al. (2010)

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

Interpretable Machine Learning – 9 / 12

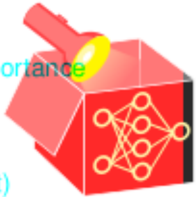
TESTING IMPORTANCE (PIMP) Altmann et al. (2010)

- PIMP was originally introduced for random forest's built-in permutation feature importance
- PIMP investigates whether the PFI score **significantly** differs from 0

  $\Rightarrow$ Useful because PFI can be non-zero due to stochasticity

Interpretable Machine Learning – 9 / 12

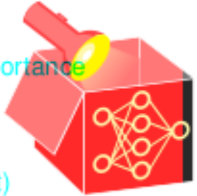
TESTING IMPORTANCE (PIMP) Altmann et al. (2010)



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

Interpretable Machine Learning – 9 / 12

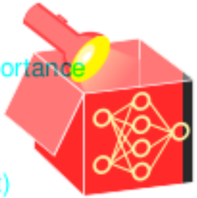
TESTING IMPORTANCE (PIMP) Altmann et al. (2010)



- PIMP was originally introduced for random forest's built-in permutation feature importance
- 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
  ⇒ By computing PFI scores again, we obtain distribution of PFI scores under H_0

Interpretable Machine Learning – 9 / 12

TESTING IMPORTANCE (PIMP) ▸ Altmann et al. (2010)



- PIMP was originally introduced for random forest's built-in permutation feature importance
- PIMP investigates whether the PFI score **significantly** differs from 0
  $\Rightarrow$ 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)
  $\Rightarrow$ Permuting y breaks relationship to all features
  $\Rightarrow$ 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

Interpretable Machine Learning – 9 / 12

TESTING IMPORTANCE (PIMP)

PIMP algorithm:

1. For $m \in \{1, \dots, n_{repetitions}\}$:
    - 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)

Interpretable Machine Learning – 10 / 12

TESTING IMPORTANCE (PIMP)

PIMP algorithm:

1. For $m \in \{1, \ldots, n_{repetitions}\}$:
   - 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

Interpretable Machine Learning – 10 / 12

TESTING IMPORTANCE (PIMP)

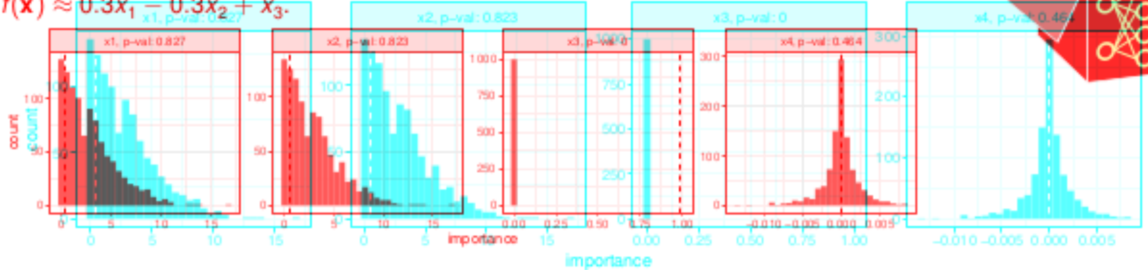
PIMP algorithm:

1. For $m \in \{1, \dots, n_{repetitions}\}$:
   - 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 $j \in \{1, \dots, p\}$:
   - Fit probability distribution of the feature importance values PFI_j^m, $m \in \{1, \dots, n_{repetitions}\}$ (choice between Gaussian, lognormal, gamma or non-parametric)
   - Compute feature importance PFI_j for the model without permutation of y (under H_1)
   - Retrieve the p-value of PFI_j based on the fitted distribution

Interpretable Machine Learning – 10 / 12

PIMP FOR EXTRAPOLATION EXAMPLE

Recall: $y = x_3 + \epsilon_y$ with $\epsilon_y \sim N(0, 0.1)$. x_1, x_2 highly correlated but independent of y, x_4 is independent of y and all other variables. Fitting a LM yields $\hat{f}(\mathbf{x}) \approx 0.3x_1 - 0.3x_2 + x_3$.



- 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
- Results: Although PFI for x_1 and x_2 is nonzero (red), PIMP considers them not significantly relevant (p-value > 0.05)

DIGRESSION: MULTIPLE TESTING PROBLEM

▸ Romano et al. (2010)

▸ 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?

Interpretable Machine Learning – 12 / 12

DIGRESSION: MULTIPLE TESTING PROBLEM

▸ Romano et al. (2010)

- 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

  $\leadsto$ **Multiple testing problem:** If 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

Interpretable Machine Learning – 12 / 12

DIGRESSION: MULTIPLE TESTING PROBLEM

Romano et al. (2010)

- 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
  $\leadsto$ **Multiple testing problem:** If 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
- 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)

Interpretable Machine Learning – 12 / 12

DIGRESSION: MULTIPLE TESTING PROBLEM

Romano et al. (2010)

- 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
  $\leadsto$ **Multiple testing problem**: If 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
- 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)
- 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