Interpretable Machine Learning

Feature Importance

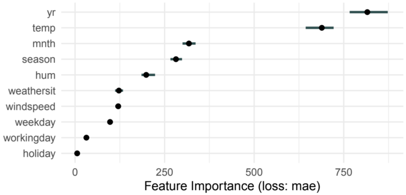


Figure: Bike Sharing Dataset

Learning goals

- Understand motivation for feature importance
- Develop an intuition for possible use-cases
- Know characteristics of feature importance methods

MOTIVATION

- **Feature effects** describe the relationship of features x with the prediction $\hat{y}$
  - requires one plot per feature
  - does not take the true target y into account

MOTIVATION

- **Feature effects** describe the relationship of features x with the prediction $\hat{y}$
  - requires one plot per feature
  - does not take the true target y into account
- **Feature importance** methods quantify the relevance of features w.r.t. prediction performance
  - condensed to one number per feature
  - provides insight into the relationship with y

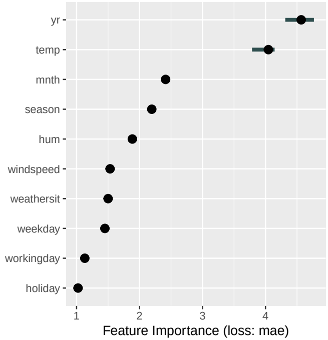
MOTIVATION

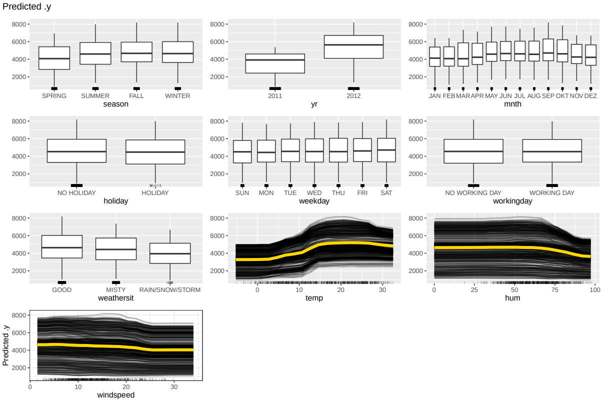
- **Feature effects** describe the relationship of features x with the prediction $\hat{y}$
  - requires one plot per feature
  - does not take the true target y into account
- **Feature importance** methods quantify the relevance of features w.r.t. prediction performance
  - condensed to one number per feature
  - provides insight into the relationship with y
- **N.B.**: Here, we use the term feature importance to describe loss-based feature importance methods. In the literature, you may find other notions of “feature importance” (e.g., variance-based methods derived from feature effect methods, see also ▸ Greenwell et al. (2020))

EXAMPLE

Feature importance offers a condensed summary of the relevance of features w.r.t. performance

- Fit random forest on bike sharing data
- Left: Feature importance ranking by permutation feature importance (PFI)
- Right: Feature effects for all features





FEATURE IMPORTANCE SCHEME

Loss-based feature importance methods are often based on two concepts

1. **Perturbation/Removal:**
   Generate predictions for which the feature of interest has been perturbed or removed
2. **Performance Comparison:**
   Compare performance under perturbation/removal with the original model performance

Depending on the type of perturbation/removal, feature importance methods provide insight into different aspects of model and data.

POTENTIAL INTERPRETATION GOALS

Feature importance methods provide condensed insights, but can only highlight certain aspects of model and data. There are different interpretation goals one might be interested in whose question of interest do not necessarily coincide (except for special cases).

For example, one may be interested in getting insight into whether the ...

(1) feature x_j is causal for the prediction?

(2) feature x_j contains prediction-relevant information about y?

(3) model requires access to x_j to achieve it's prediction performance?

POTENTIAL INTERPRETATION GOALS

Feature importance methods provide condensed insights, but can only highlight certain aspects of model and data. There are different interpretation goals one might be interested in whose question of interest do not necessarily coincide (except for special cases).

For example, one may be interested in getting insight into whether the ...

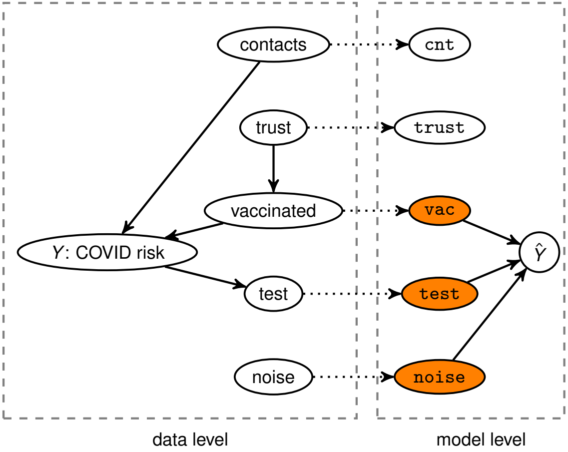
(1) feature x_j is causal for the prediction?

- Changing feature value x_j has an effect on prediction $\hat{y} = \hat{f}(x)$
- In LM: non-zero coefficient, in ML: present feature effect
- **Note:** If x_j is causal for prediction $\hat{y} \not\Rightarrow$ causal for the ground truth y, e.g.:
  - A disease symptom may be used in a model to predict disease status $\rightsquigarrow$ causal for prediction $\hat{y}$
  - But intervening on disease symptom does not have an effect on the disease $\rightsquigarrow$ not causal for the ground truth y

(2) feature x_j contains prediction-relevant information about y?

(3) model requires access to x_j to achieve it's prediction performance?

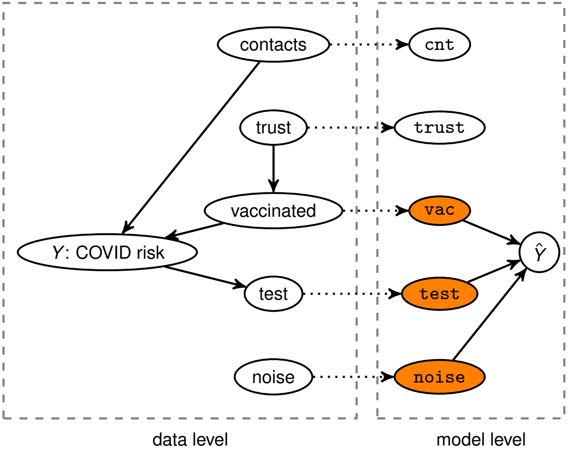
EXAMPLE: CAUSAL FOR THE PREDICTION (1)



A feature may be causal for the prediction $\hat{y}$ (1) without containing prediction-relevant information about y (2)

Examples: overfitting due noisy features

EXAMPLE: CAUSAL FOR THE PREDICTION (1)



A feature may be causal for the prediction $\hat{y}$ (1) without containing prediction-relevant information about y (2)
Examples: overfitting due noisy features

- All features used by the model are of interest
- Here: Model uses feature `noise`, although it does not contain prediction-relevant information about y (data level)

$\Rightarrow$ Overfitted models may use many noise features which are deemed relevant on model level (but not on data level)

POTENTIAL INTERPRETATION GOALS

Feature importance methods provide condensed insights, but can only highlight certain aspects of model and data. There are different interpretation goals one might be interested in whose question of interest do not necessarily coincide (except for special cases).

For example, one may be interested in getting insight into whether the ...

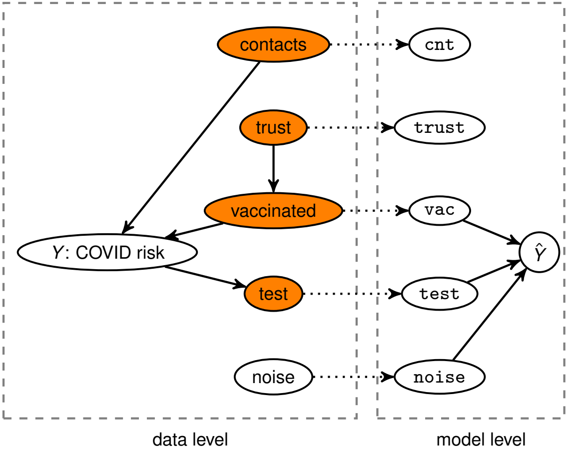
(1) feature x_j is causal for the prediction?

(2) feature x_j contains prediction-relevant information about y?

- Feature x_j helps to predict the target y (e.g., conditional expectation) w.r.t. performance
- If $x_j \perp y$ (independent) then x_j and y have zero mutual information (since $\mathbb{E}[y|x_j] = \mathbb{E}[y]$) $\rightsquigarrow$ x_j has no prediction-relevant information

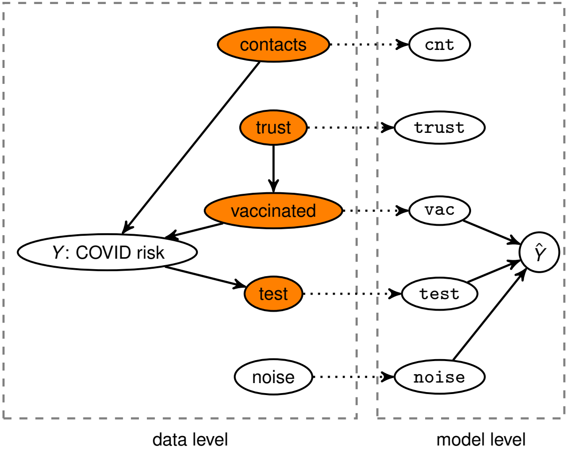
(3) model requires access to x_j to achieve it's prediction performance?

EXAMPLE: CONTAINS PREDICTION-RELEVANT INFORMATION (2)



A feature may contain prediction-relevant information (2) without causing the prediction (1)
Examples: underfitting, model multiplicity

EXAMPLE: CONTAINS PREDICTION-RELEVANT INFORMATION (2)



A feature may contain prediction-relevant information (2) without causing the prediction (1)
Examples: underfitting, model multiplicity

- All prediction-relevant features for y are of interest
- Example: All features that are directly or indirectly (i.e., via another feature) connected to y

$\Rightarrow$ Underfitted models may ignore prediction-relevant features such as `contacts` here

POTENTIAL INTERPRETATION GOALS

Feature importance methods provide condensed insights, but can only highlight certain aspects of model and data. There are different interpretation goals one might be interested in whose question of interest do not necessarily coincide (except for special cases).

For example, one may be interested in getting insight into whether the ...

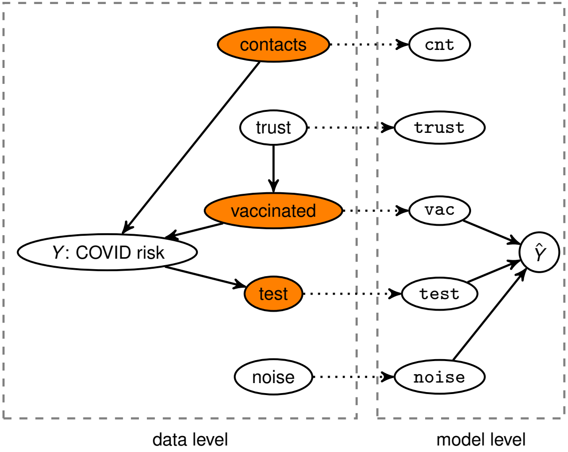
(1) feature x_j is causal for the prediction?

(2) feature x_j contains prediction-relevant information about y?

(3) model requires access to x_j to achieve it's prediction performance?

- Feature x_j helps to predict the target y w.r.t. performance, compared to using only x_{-j}
- If $x_j \perp y | x_{-j}$ (independent) then $\mathbb{E}[y|x_{-j}] = \mathbb{E}[y|x_j, x_{-j}]$
  $\leadsto$ x_j does not contribute unique prediction-relevant information about y
- **Note:** A model may rely on features that can be replaced with others, e.g., a random forest fitted on data with $\mathbb{E}[y|x_1] \neq \mathbb{E}[y]$ and $\mathbb{E}[y|x_1] = \mathbb{E}[y|x_1, x_2]$ where x_1 was not used as split variable may rely on x_2

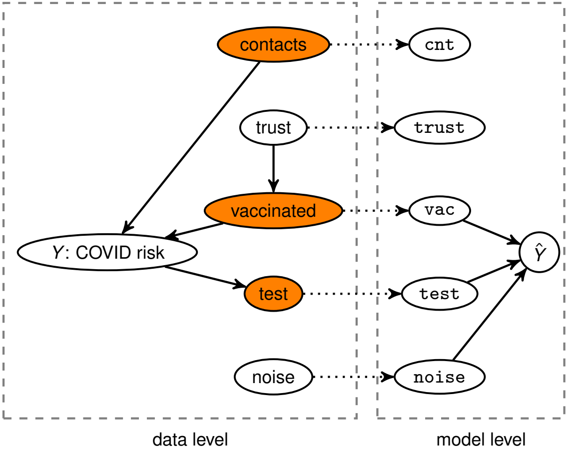
EXAMPLE: UNIQUE PREDICTION RELEVANT INFORMATION (3)



A feature may contain prediction-relevant information (2), without the model requiring access to the feature for (optimal) prediction performance (3)

Examples: correlated features, confounding

EXAMPLE: UNIQUE PREDICTION RELEVANT INFORMATION (3)



A feature may contain prediction-relevant information (2), without the model requiring access to the feature for (optimal) prediction performance (3)
Examples: correlated features, confounding

- All unique prediction-relevant features for y are of interest
- Example: All features that are directly connected to y

$\Rightarrow$ `trust` and `vaccinated` may be correlated but only `vaccinated` is directly connected to y

Interpretable Machine Learning

Permutation Feature Importance (PFI)

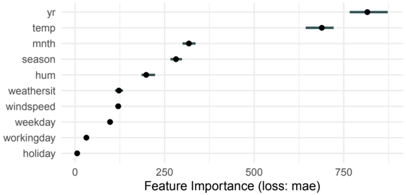


Figure: Bike Sharing Dataset

Learning goals

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

PERMUTATION FEATURE IMPORTANCE (PFI) ▸ Breiman (2001)

Idea: "Destroy" feature of interest x_j by perturbing it such that it becomes uninformative, e.g., randomly permute observations 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 features** and **with permuted feature values** $\tilde{x}_S$
- Repeat permuting the feature (e.g., m times) and average the difference of both errors:

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

PERMUTATION FEATURE IMPORTANCE (PFI) ▸ Breiman (2001)

Idea: "Destroy" feature of interest x_j by perturbing it such that it becomes uninformative, e.g., randomly permute observations 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 features** and **with permuted feature values** $\tilde{x}_S$
- Repeat permuting the feature (e.g., m times) and average the difference of both errors:

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

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

$\Rightarrow$

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

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

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

i	$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$
1	2	4	7
$\vdots$	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 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 ($P(X_S)$).
   $\Rightarrow$ Randomly permute feature x_S
   $\Rightarrow$ Replace original feature with permuted feature $\tilde{x}_S$ and create data $\tilde{\mathcal{D}}^S$ containing $\tilde{x}_S$
2. **Prediction:** Make predictions for both data, i.e., $\mathcal{D}$ and $\tilde{\mathcal{D}}^S$

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

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

$\mathcal{D}$

$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$
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

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

i	x_S	x_2	x_3
1	2	4	7
⋮	1	5	8
n	3	6	9

$\mathcal{D}$

x_1	x_2	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

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

i	x_S	x_2	x_3		x_1	x_2	x_3		ΔL
1	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

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	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	$\mathbf{x}_S$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	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

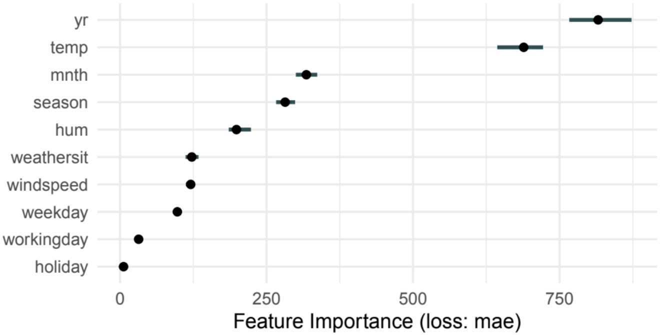
= 0.4

$\widehat{PFI}_S$ = 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 pertubation 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

COMMENTS ON PFI

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

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

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

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

COMMENTS ON PFI

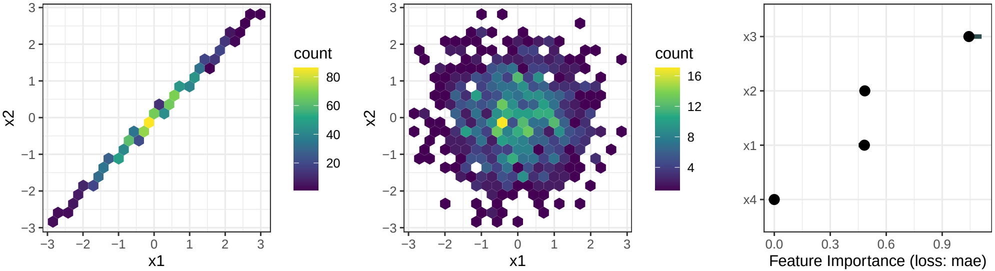
- Interpretation: PFI is the increase of model error when feature's information is destroyed
- Results can be unreliable due to random permutations
  $\Rightarrow$ Solution: Average results over multiple repetitions
- Permuting features despite correlation with other features can lead to unrealistic 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 of PFI depends on whether training or test data is used

COMMENTS ON PFI - EXTRAPOLATION

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

COMMENTS ON PFI - EXTRAPOLATION

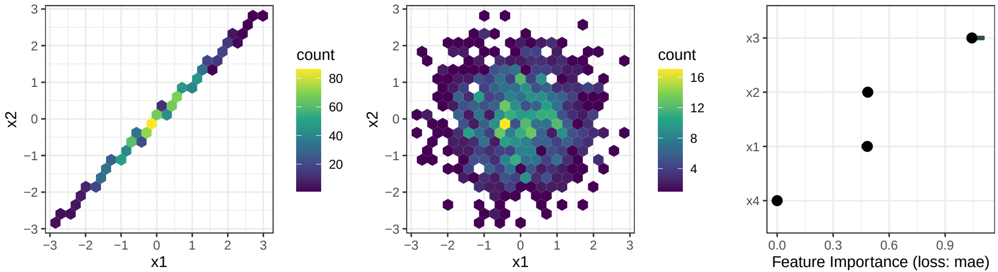
Example: Let $y = x_3 + \epsilon_y$ with $\epsilon_y \sim N(0, 0.1)$ where $x_1 := \epsilon_1$, $x_2 := x_1 + \epsilon_2$ are highly correlated ($\epsilon_1 \sim N(0,1)$, $\epsilon_2 \sim N(0, 0.01)$) and $x_3 := \epsilon_3$, $x_4 := \epsilon_4$, with $\epsilon_3, \epsilon_4 \sim N(0,1)$. All noise terms 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 $x_1, \dots, x_4$ 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

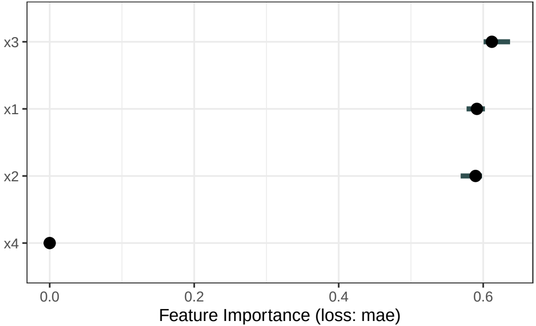
Example: Let $x_1, \ldots, x_4$ be independently and uniformly sampled from $\{-1, 1\}$ and

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

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 - 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.

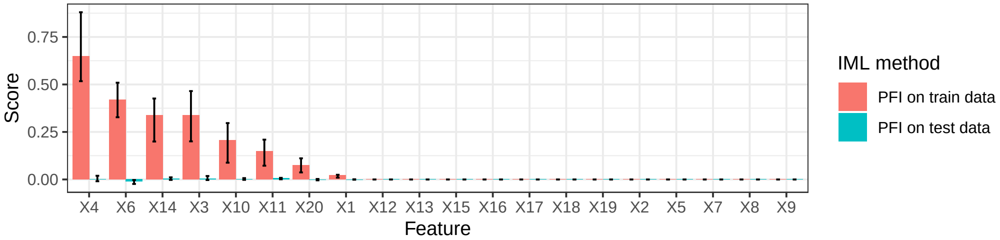


Figure: While PFI on test data considers all features to be irrelevant, PFI on train data exposes the features on which the model overfitted.

COMMENTS ON PFI - TEST VS. TRAINING DATA

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

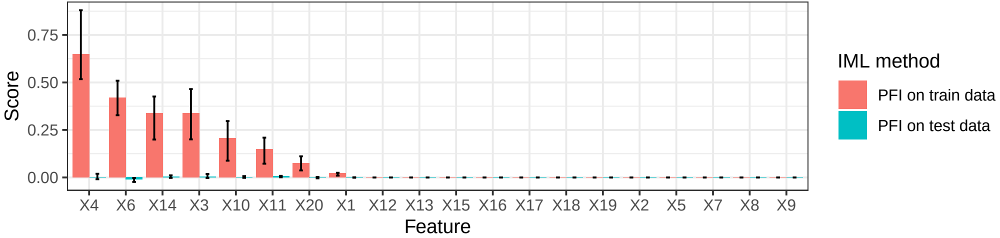


Figure: While PFI on test data considers all features to be irrelevant, PFI on train data exposes the features on which the model overfitted.

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.

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

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) $\Rightarrow PFI_j = 0$

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

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

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

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

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)

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

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

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)

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

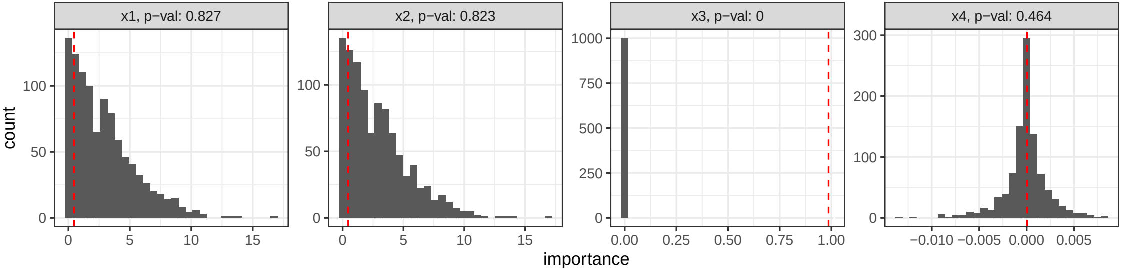
TESTING IMPORTANCE (PIMP)

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
3. For each feature $j \in \{1, \ldots, p\}$:
   - Fit probability distribution of the feature importance values PFI_j^m, $m \in \{1, \ldots, n_{repetitions}\}$ (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

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)

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

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
  $\rightsquigarrow$ **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

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
  $\rightsquigarrow$ **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)

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
  $\rightsquigarrow$ **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

Interpretable Machine Learning

Conditional Feature Importance (CFI)

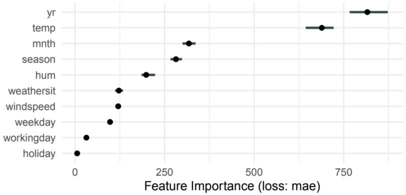


Figure: Bike Sharing Dataset

Learning goals

- Extrapolation and Conditional Sampling
- Conditional Feature Importance (CFI)
- Interpretation of CFI and difference to PFI

CONDITIONAL FEATURE IMPORTANCE IDEA

- **Permutation Feature Importance Idea:** Replace the feature of interest x_j with an independent sample from the marginal distribution $\mathbb{P}(x_j)$, e.g. by randomly permuting observations in x_j

CONDITIONAL FEATURE IMPORTANCE IDEA

- **Permutation Feature Importance Idea:** Replace the feature of interest x_j with an independent sample from the marginal distribution $\mathbb{P}(x_j)$, e.g. by randomly permuting observations in x_j
- **Problem:** Under dependent features, permutation leads to extrapolation

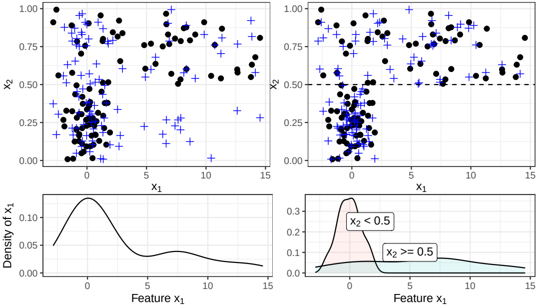
CONDITIONAL FEATURE IMPORTANCE IDEA

- **Permutation Feature Importance Idea:** Replace the feature of interest x_j with an independent sample from the marginal distribution $\mathbb{P}(x_j)$, e.g. by randomly permuting observations in x_j
- **Problem:** Under dependent features, permutation leads to extrapolation
- **Conditional Feature Importance Idea:** Resample x_j from the conditional distribution $\mathbb{P}(x_j|x_{-j})$, such that the joint distribution is preserved, i.e., $\mathbb{P}(x_j|x_{-j})\mathbb{P}(x_{-j}) = \mathbb{P}(x_j, x_{-j})$

CONDITIONAL FEATURE IMPORTANCE IDEA

- **Permutation Feature Importance Idea:** Replace the feature of interest x_j with an independent sample from the marginal distribution $\mathbb{P}(x_j)$, e.g. by randomly permuting observations in x_j
- **Problem:** Under dependent features, permutation leads to extrapolation
- **Conditional Feature Importance Idea:** Resample x_j from the conditional distribution $\mathbb{P}(x_j|x_{-j})$, such that the joint distribution is preserved, i.e., $\mathbb{P}(x_j|x_{-j})\mathbb{P}(x_{-j}) = \mathbb{P}(x_j, x_{-j})$

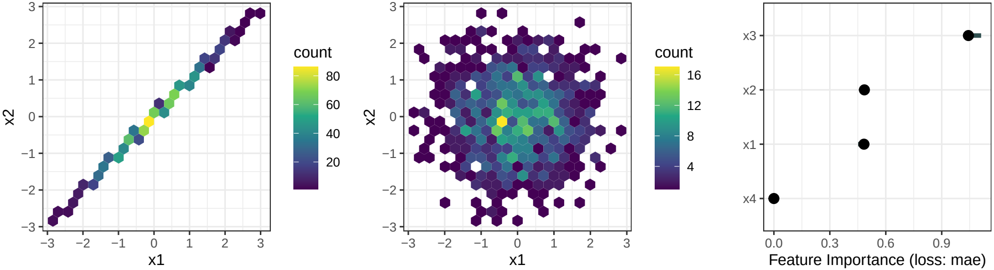
Example: Conditional permutation scheme ▸ Molnar et. al (2020)



- $X_2 \sim U(0,1)$ and $X_1 \sim N(0,1)$ if $X_2 < 0.5$, else $X_1 \sim N(4,4)$ (black dots)
- **Left:** For $X_2 < 0.5$, permuting X_1 (crosses) preserves marginal (but not joint) distribution
  $\rightsquigarrow$ Bottom: Marginal density of X_1
- **Right:** Permuting X_1 within subgroups $X_2 < 0.5$ & $X_2 \geq 0.5$ reduces extrapolation
  $\rightsquigarrow$ Bottom: Density of X_1 conditional on groups

RECALL: EXTRAPOLATION IN PFI

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 $\mathbb{P}(\mathbf{x})$ $\leadsto$ x_1 and x_2 are considered relevant

CONDITIONAL FEATURE IMPORTANCE

▸ Strobl et al. (2008) ▸ Hooker et al. (2021)

Conditional feature importance (CFI) for features x_S using test data $\mathcal{D}$:

- Measure the error **with unperturbed features**.
- Measure the error **with perturbed feature values** $\tilde{x}^{S|-S}$, where $\tilde{x}_S^{S|-S} \sim \mathbb{P}(x_S|x_{-S})$
- Repeat permuting the feature (e.g., m times) and average the difference of both errors:

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

Here, $\tilde{\mathcal{D}}^{S|-S}$ denotes the dataset where features x_S where sampled conditional on the remaining features x_{-S}.

IMPLICATIONS OF CFI ▸ König et al. (2020)

Interpretation: Due to the conditional sampling w.r.t. all other features, CFI quantifies a feature's unique contribution to the model performance.

IMPLICATIONS OF CFI ▸ König et al. (2020)

Interpretation: Due to the conditional sampling w.r.t. all other features, CFI quantifies a feature's unique contribution to the model performance.

Entanglement with data:

- If feature x_S does not contribute unique information about y, i.e., $x_S \perp y | x_{-S} \Rightarrow \text{CFI} = 0$
- Why? Under the conditional independence $\mathbb{P}(\tilde{x}^{S|-S}, y) = \mathbb{P}(x, y)$
  $\rightsquigarrow$ no prediction-relevant information is destroyed by permutation of x_S conditional on x_{-S}

IMPLICATIONS OF CFI ▸ König et al. (2020)

Interpretation: Due to the conditional sampling w.r.t. all other features, CFI quantifies a feature's unique contribution to the model performance.

Entanglement with data:

- If feature x_S does not contribute unique information about y, i.e., $x_S \perp y | x_{-S} \Rightarrow \text{CFI} = 0$
- Why? Under the conditional independence $\mathbb{P}(\tilde{x}^{S|-S}, y) = \mathbb{P}(x, y)$
  $\rightsquigarrow$ no prediction-relevant information is destroyed by permutation of x_S conditional on x_{-S}

Entanglement with model:

- If the model does not use a feature $\Rightarrow \text{CFI} = 0$
- Why? Then the prediction is not affected by any perturbation of the feature
  $\rightsquigarrow$ model performance does not change after conditional permutation

IMPLICATIONS OF CFI

Can we gain insight into whether ...

1. the feature x_j is causal for the prediction?
   - $CFI_j \neq 0 \Rightarrow$ model relies on x_j (converse does not hold, see next slide)

IMPLICATIONS OF CFI

Can we gain insight into whether ...

1. the feature x_j is causal for the prediction?
   - $CFI_j \neq 0 \Rightarrow$ model relies on x_j (converse does not hold, see next slide)
2. the variable x_j contains prediction-relevant information?
   - If $x_j \not\perp y$ but $x_j \perp y | x_{-j}$ (e.g., x_j and x_{-j} share information) $\Rightarrow CFI_j = 0$
   - x_j is not exploited by model (regardless of whether it is useful for y or not) $\Rightarrow CFI_j = 0$

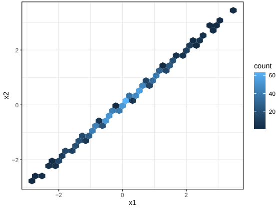
IMPLICATIONS OF CFI

Can we gain insight into whether ...

1. the feature x_j is causal for the prediction?
   - $CFI_j \neq 0 \Rightarrow$ model relies on x_j (converse does not hold, see next slide)
2. the variable x_j contains prediction-relevant information?
   - If $x_j \not\perp y$ but $x_j \perp y | x_{-j}$ (e.g., x_j and x_{-j} share information) $\Rightarrow CFI_j = 0$
   - x_j is not exploited by model (regardless of whether it is useful for y or not) $\Rightarrow CFI_j = 0$
3. Does the model require access to x_j to achieve its prediction performance?
   - $CFI_j \neq 0 \Rightarrow x_j$ contributes unique information (meaning $x_j \not\perp y | x_{-j}$)
   - Only uncovers the relationships that were exploited by the model

COMPARISON: PFI AND CFI

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



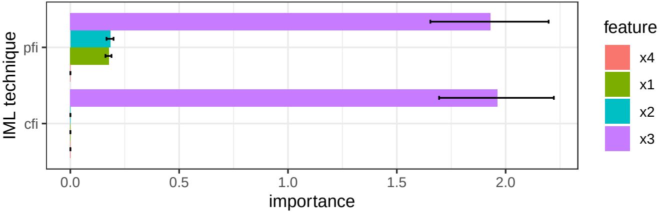


Figure: Density plot for x_1, x_2 before permuting x_1 (left). PFI and CFI (right).

$\Rightarrow$ x_1 and x_2 are 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)
$\Rightarrow$ Since x_1 can be reconstructed from x_2 and vice versa, CFI considers x_1 and x_2 to be irrelevant

Interpretable Machine Learning

Leave One Covariate Out (LOCO)

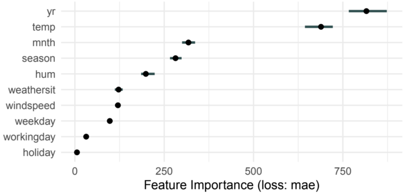


Figure: Bike Sharing Dataset

Learning goals

- Definition of LOCO
- Interpretation of LOCO

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018) ▸ Tibshirani (2018)

LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

LEAVE ONE COVARIATE OUT (LOCO) ‣ Lei et al. (2018) ‣ Tibshirani (2018)

LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \dots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train},-j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train},-j})$

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018) ▸ Tibshirani (2018)

LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \ldots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train},-j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train},-j})$
2. compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e. $\Delta_j^{(i)} = \left| y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)}) \right| - \left| y^{(i)} - \hat{f}(x^{(i)}) \right|$ with $i \in \mathcal{D}_{\text{test}}$

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018) ▸ Tibshirani (2018)

LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \ldots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train},-j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train},-j})$
2. compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e.
   $\Delta_j^{(i)} = \left|y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)})\right| - \left|y^{(i)} - \hat{f}(x^{(i)})\right|$ with $i \in \mathcal{D}_{\text{test}}$
3. yield the importance score $\text{LOCO}_j = \text{med}\left(\Delta_j\right)$

LEAVE ONE COVARIATE OUT (LOCO) ▸ Lei et al. (2018) ▸ Tibshirani (2018)

LOCO idea: Remove the feature from the dataset, refit the model on the reduced dataset, and measure the loss in performance compared to the model fitted on the complete dataset.

Definition: Given training and test datasets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}} \subseteq \mathcal{D}$, some $\mathcal{I}$ and a model $\hat{f} = \mathcal{I}(\mathcal{D}_{\text{train}})$. Then LOCO for a feature $j \in \{1, \dots, p\}$ can be computed as follows:

1. learn model on dataset $\mathcal{D}_{\text{train},-j}$ where feature x_j was removed, i.e. $\hat{f}_{-j} = \mathcal{I}(\mathcal{D}_{\text{train},-j})$
2. compute the difference in local L_1 loss for each element in $\mathcal{D}_{\text{test}}$, i.e. $\Delta_j^{(i)} = \left|y^{(i)} - \hat{f}_{-j}(x_{-j}^{(i)})\right| - \left|y^{(i)} - \hat{f}(x^{(i)})\right|$ with $i \in \mathcal{D}_{\text{test}}$
3. yield the importance score $\text{LOCO}_j = \text{med}\left(\Delta_j\right)$

The method can be generalized to other loss functions and aggregations. If we use mean instead of median we can rewrite LOCO as

$$\text{LOCO}_j = \mathcal{R}_{\text{emp}}(\hat{f}_{-j}) - \mathcal{R}_{\text{emp}}(\hat{f}).$$

BIKE SHARING EXAMPLE

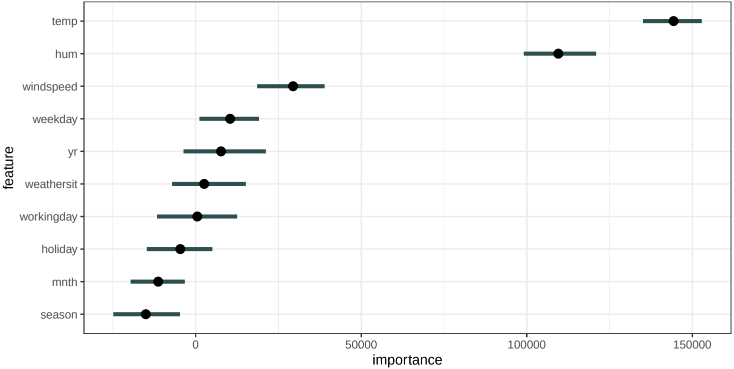


Figure: A random forest with default hyperparameters was fit on 70% of the bike sharing data (training set) to optimize MSE. Then LOCO was computed for all features on the test data. The temperature is the most important feature. Without access to `temp`, the MSE increases by approx. $140,000$.

INTERPRETATION OF LOCO

Interpretation: LOCO estimates the generalization error of the learner on a reduced dataset $\mathcal{D}_{-j}$.

Can we get insight into whether the ...

1. feature x_j is causal for the prediction $\hat{y}$?
   - In general, no also because we refit the model (counterexample on the next slide)
2. feature x_j contains prediction-relevant information?
   - In general, no (counterexample on the next slide)
3. model requires access to x_j to achieve its prediction performance?
   - Approximately, it provides insight into whether the *learner* requires access to x_j

INTERPRETATION OF LOCO

Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

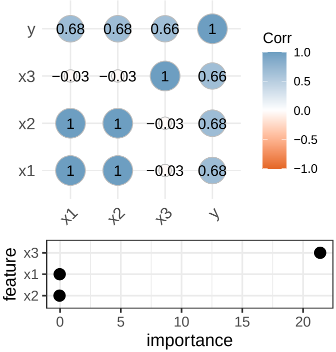
$\Rightarrow$ Fitting a LM yields $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$

INTERPRETATION OF LOCO

Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

$\Rightarrow$ Fitting a LM yields $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



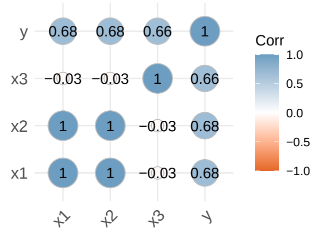
Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations

INTERPRETATION OF LOCO

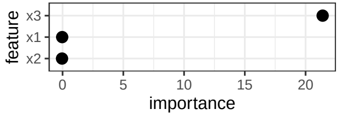
Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

$\Rightarrow$ Fitting a LM yields $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations



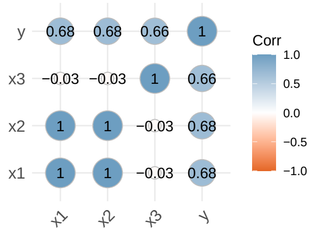
$\Rightarrow$ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coefficient of x_2 is 2.05)

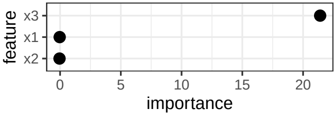
INTERPRETATION OF LOCO

Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

$\Rightarrow$ Fitting a LM yields $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$

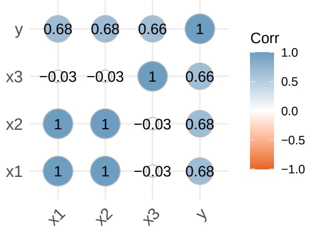




Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations

$\Rightarrow$ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coefficient of x_2 is 2.05)
$\Rightarrow$ We also can't infer (2), e.g., $Cor(x_2, y) = 0.68$ but $\text{LOCO}_2 \approx 0$

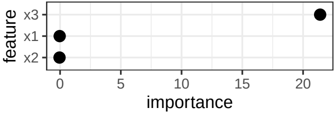
INTERPRETATION OF LOCO



Example: Sample 1000 observations with

- $x_1, x_3 \sim N(0, 5)$
- $x_2 = x_1 + \epsilon_2$ with $\epsilon_2 \sim N(0, 0.1)$
- $y = x_2 + x_3 + \epsilon$ with $\epsilon \sim N(0, 2)$

$\Rightarrow$ Fitting a LM yields $\hat{f}(x) = -0.02 - 1.02x_1 + 2.05x_2 + 0.98x_3$



Top: Correlation matrix
Bottom: LOCO importance of LM fitted on 70% of the data computed on 30% remaining observations

$\Rightarrow$ We cannot infer (1) from LOCO (e.g. $\text{LOCO}_2 \approx 0$ but coefficient of x_2 is 2.05)
$\Rightarrow$ We also can't infer (2), e.g., $Cor(x_2, y) = 0.68$ but $\text{LOCO}_2 \approx 0$
$\Rightarrow$ We can get insight into (3): x_2 and x_1 highly correlated with $\text{LOCO}_1 = \text{LOCO}_2 \approx 0$
$\rightsquigarrow$ x_2 and x_1 can take each others place if one of them is left out (not the case for x_3)

PROS AND CONS

Pros:

- Requires (only?) one refitting step per feature for evaluation
- Easy to implement
- Testing framework available in ▸ Lei et al. (2018)

Cons:

- Does not provide insight into a specific model, but rather a learner on a specific dataset
- Model training is a random process, so estimates can be noisy (which is problematic for inference about model and data)
- Requires re-fitting the learner for each feature $\rightarrow$ computationally intensive compared to PFI