Global Model-Agnostic Methods

Explainable Artificial Intelligence

Dr. Stefan Heindorf

Model-agnostic explanation methods

Model-agnostic: separate explanation from the machine learning model

Model flexibility

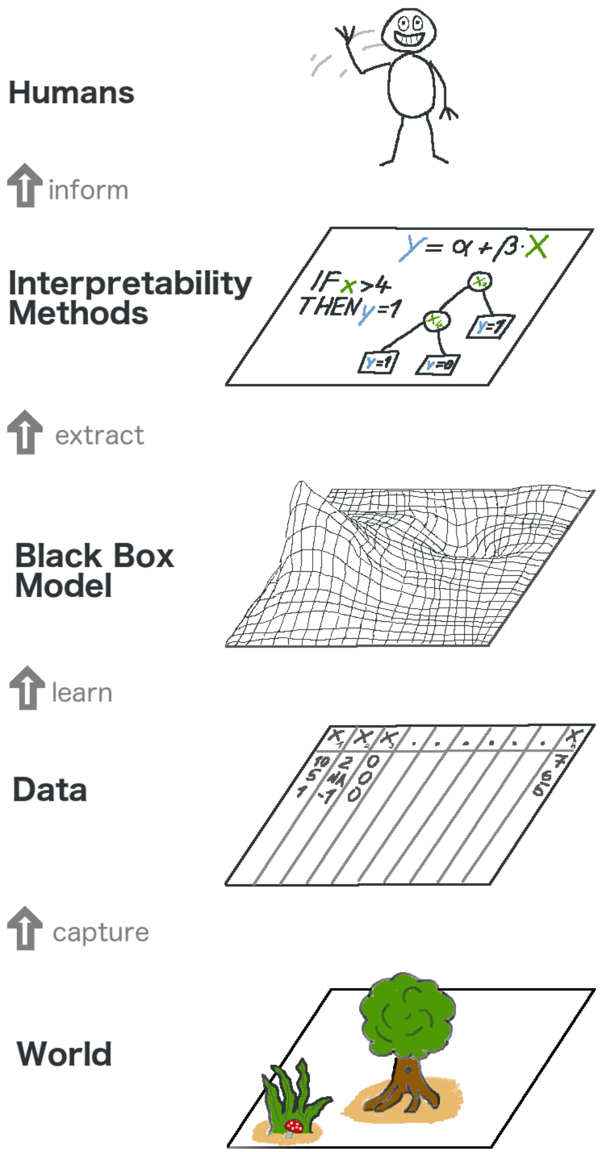
- Interpretation method can work with any machine learning model
- Example: random forests and deep NN

Explanation flexibility

- Not limited to a certain form of explanation
- Example: linear formula vs. graphic with feature importances

Representation flexibility

- Explanation system can use a different feature representation as the model being explained
- Example: text classifiers uses word embeddings, but is explained in terms of individuals words



Christoph Molnar (https://christophm.github.io/interpretable-ml-book/images/big-picture.png), https://creativecommons.org/licenses/by-nc-sa/4.0/

Global vs. local model-agnostic explanation methods

Global methods (this lecture)

- Describe the average behavior of a machine learning model
- Useful to understand the general mechanisms in the data or debug a model
- Examples
  - Partial dependence plot: expected prediction when all other features stay the same
  - Permutation feature importance measures the importance of a feature as an increase in loss when the feature is permuted
  - Global surrogate models replaces the original model with a simpler model for interpretation

Local methods (next lecture)

- Explain a single prediction
- Examples
  - LIME
  - Anchors
  - SHAP
  - Counterfactuals

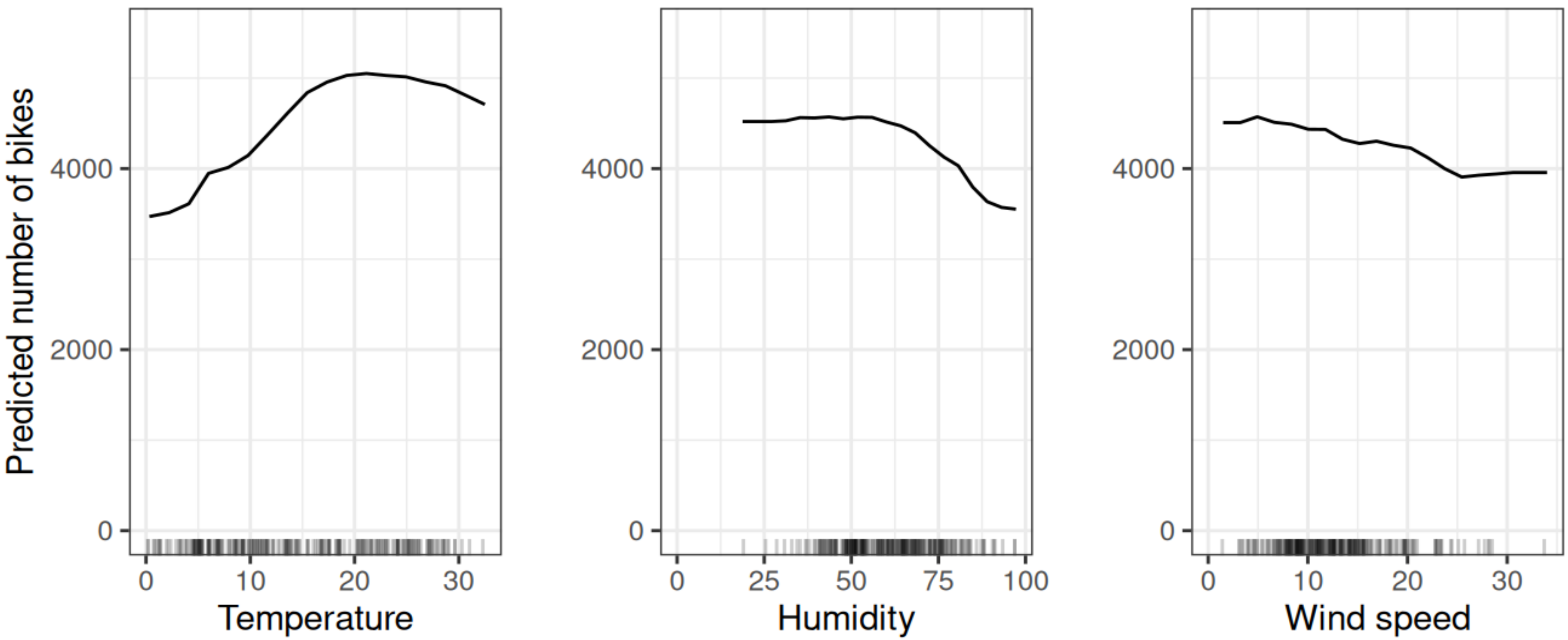
Outlook

- Partial dependence plot
- Permutation feature importance
- Global surrogate

Partial Dependence Plot

Partial dependence plot (PDP)

Example: numeric features from bike rental dataset



Partial dependence plot (PDP) [Friedman 2001]

Motivation and intuition

Shows the marginal effect: change of the prediction when varying the value of a subset of feature while other feature values are kept constant

The partial dependence function is defined as

$$\hat{f}_S(\mathbf{x}_S) = \mathbb{E}_{\mathbf{X_C}}\left[\hat{f}(\mathbf{x}_S, \mathbf{X}_\mathrm{C})\right] = \sum_{\mathbf{x}_C \in \mathbf{X}_C} \Pr(\mathbf{X}_C = \mathbf{x}_C)\, \hat{f}(\mathbf{x}_S, \mathbf{x}_C)$$

S	features we would like to know the effect on the prediction of (usually one or two)
C	other features (complement of S)
$\mathbf{x}_S$	feature values for plotting the partial dependence function
$\mathbf{X}_C$	other feature values, which are treated as random variables here (vectors $\mathbf{x}_S$ and $\mathbf{x}_C$ combined make up the total feature space $\mathbf{x}$)
$\hat{f}$	machine learning model

Note: by marginalizing over the other features, we get a function that depends only on features in S, interactions with other features included

Partial function

The partial dependence function $\hat{f}_S$ is estimated by calculating

$$\hat{f}_S(\mathbf{x}_S) = \sum_{\mathbf{x}_C \in \mathbf{X}_C} \Pr(\mathbf{X}_C = \mathbf{x}_c)\, \hat{f}(\mathbf{x}_S, \mathbf{x}_C) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})$$

n number of instances in the dataset

$\mathbf{x}_C^{(i)}$ actual feature values from the dataset

- Partial dependence function: average marginal effect on the prediction given value(s) of features S
- Assumption of the PDP: features in C are not correlated with the features in S (if violated: the averages include data points that are unlikely)
- PDP is a global method: all instances are considered

Task: Draw partial dependence plot

Training data

x_1	x_2	y
0.3	0.2	1100
0.5	0.6	2100

Learned linear model (without intercept)

$$\hat{y} = 3000x_1 + 1000x_2$$

Draw the partial dependence plot of x_1
(for $x_1 = 0$, $x_1 = 0.5$, and $x_1 = 1.0$)

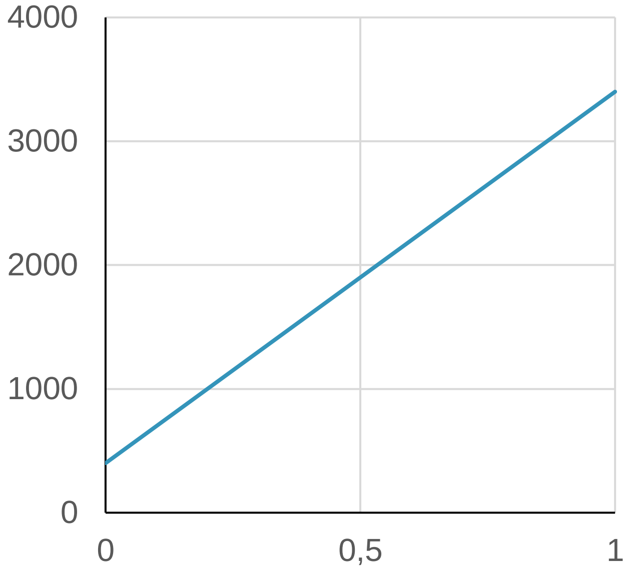
Solution: Partial dependence plot

Predictions

x_1	x_2	$\hat{y}$
0	0.2	200
0	0.6	600
0.5	0.2	1700
0.5	0.6	2100
1.0	0.2	3200
1.0	0.6	3600

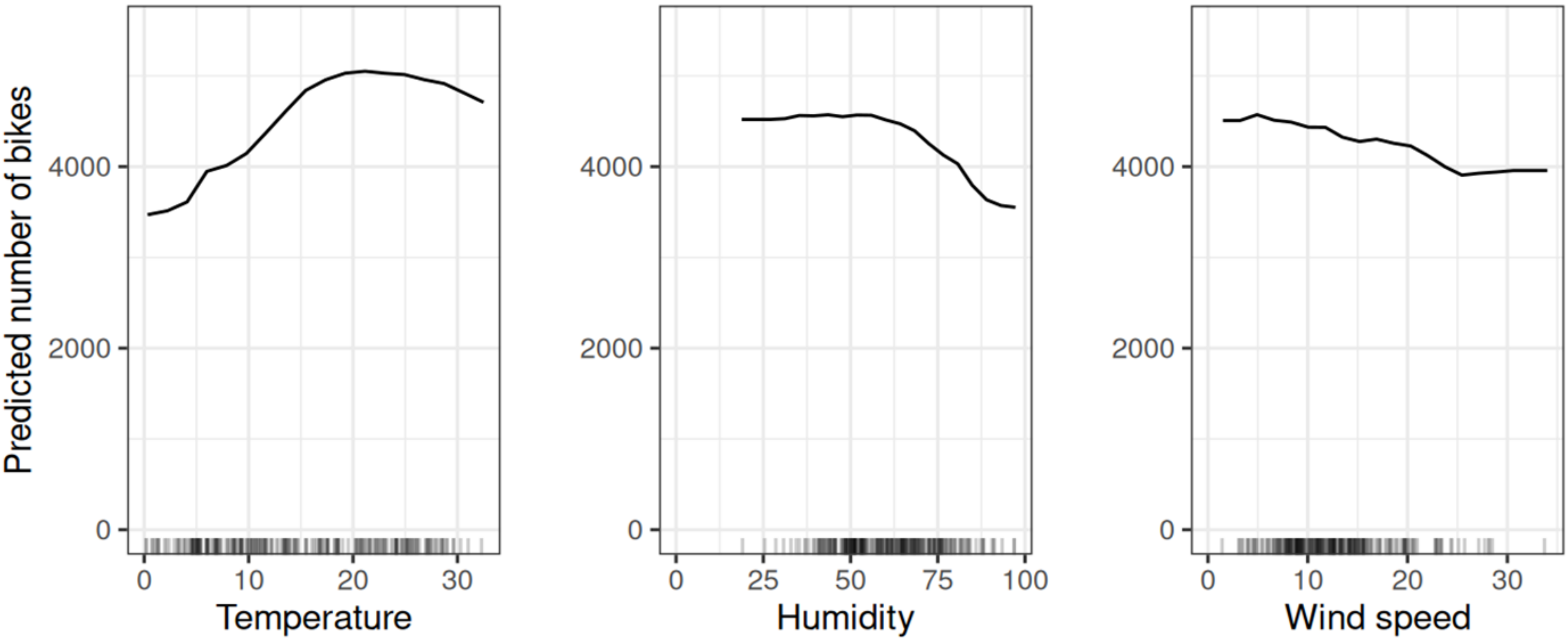
Partial dependence function

x_1	$\hat{f}_S$
0	400
0.5	1900
1.0	3400



Partial dependence plot

Example: numeric features from bike rental dataset



How to interpret the results?

Partial dependence plot

Interpretation of results

Temperature

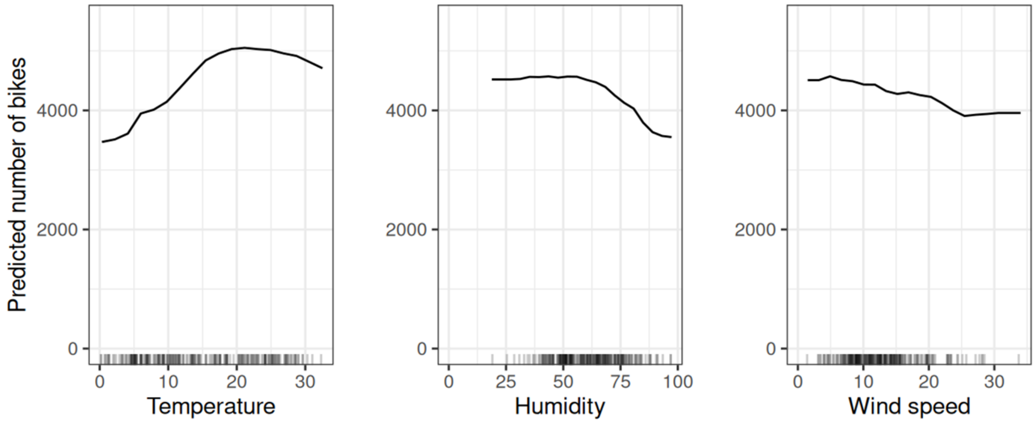
- Most bikes rented at temperatures around 15 to 25 degrees
- At lower temperatures, less bikes rented

Humidity

- As humidity goes beyond 60%, bike rentals decrease
- Humidity below 60% has little effect

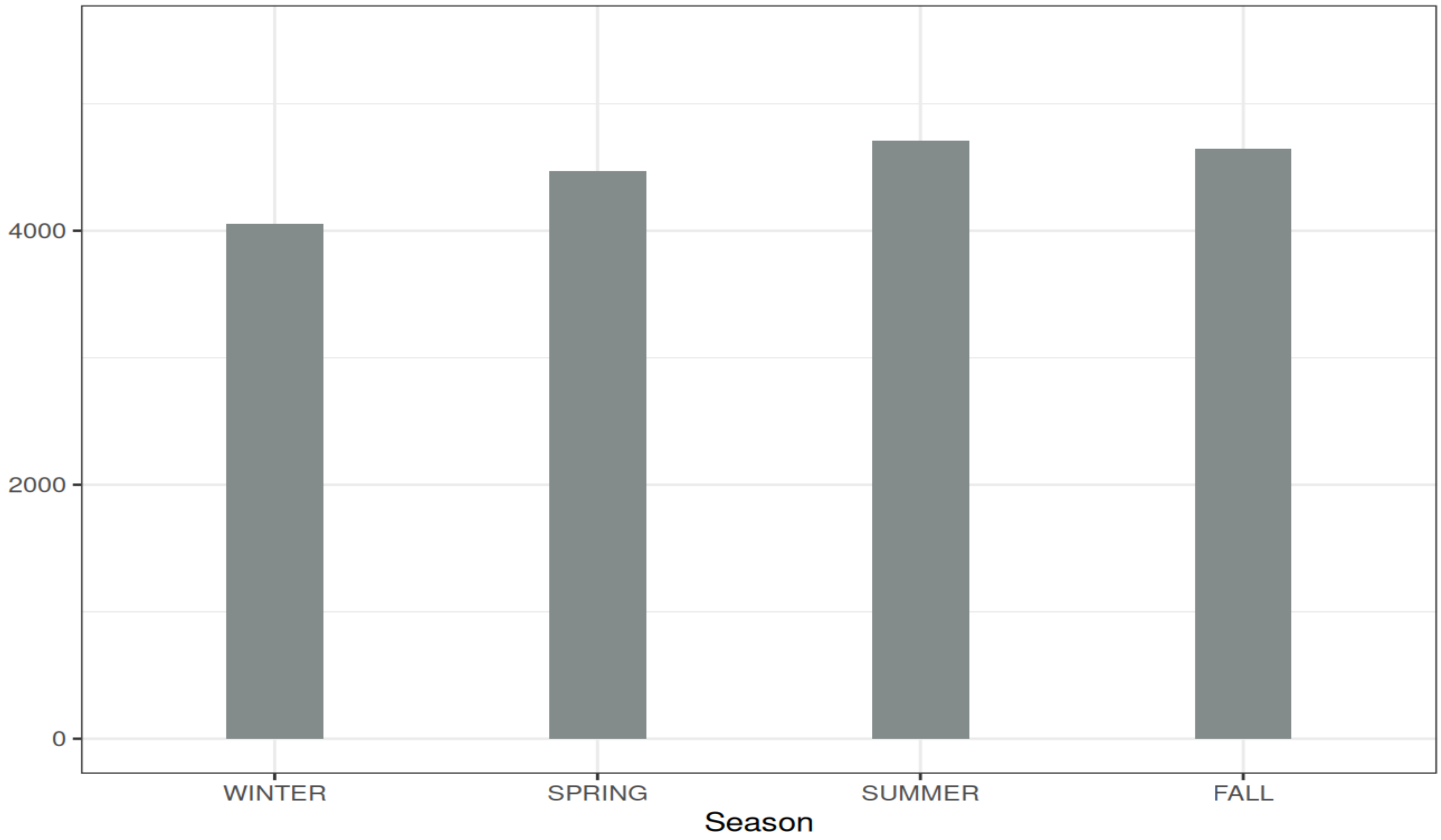
Wind speed

- As wind speed decreases, less bikes are rented
- For wind speeds above 25km/h: not much data available (see x axis)



Partial dependence plot

Example: categorical feature from bike rental dataset



PDP-based feature importance [Greenwell et al, 2018]

Idea

- Flat PDP indicates that the feature is not important
- The more the PDP varies, the more important the feature is

For numeric features: PDP-based feature importance defined as the sample standard deviation ($\mathbf{x}_S^{(k)}$ are the K unique values of feature $\boldsymbol{X}_S$)

$$I(\mathbf{x}_S) = \sqrt{\frac{1}{K-1}\sum_{i=1}^{K}\left(\hat{f}_S\left(\mathbf{x}_S^{(k)}\right) - \frac{1}{K}\sum_{k=1}^{K}\hat{f}_S\left(\mathbf{x}_S^{(k)}\right)\right)^2}$$

For categorical features: feature importance defined via range rule

$$I(\mathbf{x}_S) = \frac{\max_k\left(\hat{f}_S\left(\mathbf{x}_S^{(k)}\right)\right) - \min_i\left(\hat{f}_S\left(\mathbf{x}_S^{(k)}\right)\right)}{4}$$

- Quick, rough estimate of the standard deviation
- The denominator four comes from the standard normal distribution: In the normal distribution, 95% of the data are minus two and plus two standard deviations around the mean

Partial dependence plot
Advantages

Intuitive, clear interpretation

- Partial dependence function at a particular feature value represents the average prediction if we force all data points to assume that feature value
- PDPs perfectly represent how the feature influences the prediction on average (if the feature for which you computed the PDP is not correlated with the other features)
- Lay people usually understand the idea of PDPs quickly

Easy implementation

Causal interpretation

- We intervene on a feature and measure the changes in the predictions
- We analyze the causal relationship between the feature and the prediction
- The relationship is causal for the model, but not necessarily for the real world!

Partial dependence plot

Disadvantages

Suitable for at most two features in a partial dependence function

- 1 feature → 2d representation
- 2 features → 3d representation / heat map
- More than two is hard to visualize and to imagine for humans

Assumes independence of features, example:

- Predict walking speed of person given height and weight
- For height (e.g. 200cm): we average over the marginal distribution of weight, which might include weights below 50 kg, which is unrealistic for a 2 m person
- In other words: When the features are correlated, we create new data points in areas of the feature distribution where the actual probability is very low

Heterogeneous effects might be hidden due to averaging, example:

- Half the data points have a positive association with the prediction for a feature
- Half the data points have a negative association (the smaller the feature value the larger the prediction)
- PD plot could be a horizontal line, because effects cancel each other out

Permutation Feature Importance

Permutation feature importance

Measure the importance of a feature by

- permuting the feature's values and
- calculating the increase in model's prediction error

If model error increases

→ the feature is important

If model error remains unchanged

→ feature is unimportant

Note

- a feature that is unimportant for one model, might be important for another
- In particular: feature unimportant for bad model, might be import for good model

→ Evaluate predictive performance of model before computing performances

Permutation feature importance

Algorithm

Input: Trained model $\hat{f}$, feature matrix $\mathbf{X}$, target vector $\mathbf{y}$, error measure $L(\mathbf{y}, \hat{f})$

1. Estimate the original model error $e_{orig} = L(\mathbf{y}, \hat{f}(\mathbf{X}))$
2. For each feature $j \in \{1, \dots, p\}$ do:
   - Generate feature matrix $\mathbf{X}_{perm,j}$ by permuting feature j in the data $\mathbf{X}$
   - Estimate error $e_{perm} = L(\boldsymbol{y}, \hat{f}(\mathbf{X}_{perm,j}))$ based on the predictions of the permuted data
   - Calculate permutation feature importance as quotient $FI_j = e_{perm}\ /\ e_{orig}$ or difference $FI_j = e_{perm} - e_{orig}$
3. Sort features by descending feature importance

Task: Permutation feature importance

Training data

x_1	x_2	y
0.3	0.2	1100
0.5	0.6	2100

Learned linear model (without intercept)

$$\hat{y} = 3000x_1 + 1000x_2$$

Compute the permutation feature importance for x_1 based on the mean absolute error (MAE)
$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n} |\mathbf{y}^{(i)} - \hat{\mathbf{y}}^{(i)}|$

Task: Permutation feature importance

For x_1

x_1	x_2	y	$\hat{y}$
0.5	0.2	1100	1700
0.3	0.6	2100	1500

Feature importance (based on mean absolute error)

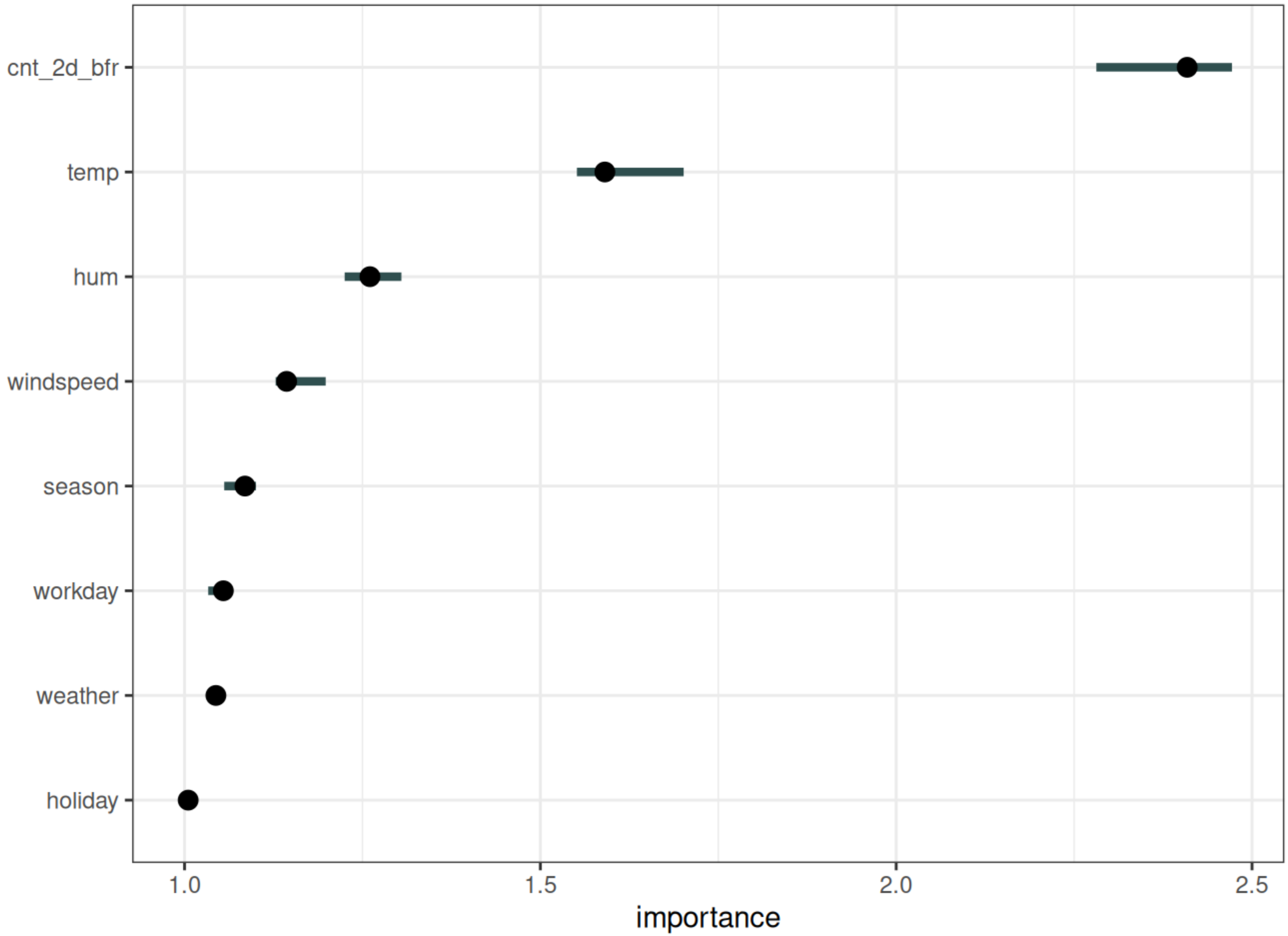
$$e_{perm} = \frac{(|1700 - 1100| + |1500 - 2100|)}{2} = 600$$

$$e_{orig} = 0$$

$$FI_j = e_{perm} - e_{orig} = 600$$

Permutation feature importance

Example: Bike sharing dataset



Permutation Feature Importance
Advantages

Intuitive interpretation
- Increase in model error when the feature’s information is destroyed

Does not require retraining the model
- In contrast to some feature selection techniques retraining the model

Can be computed on unseen data
- In contrast to impurity measures such as Gini Loss, ….

Permutation feature importance

Disadvantages

Requires access to the true outcome

- The model alone plus unlabeled data are not sufficient to compute the error

Varies from run to run

- Permutation feature importance depends on shuffling the feature

→ different result for every run

- Repeating the permutation and averaging the importance measures stabilizes the measure, but adds runtime

Assumes independence of features

- Just like partial dependence plots
- Permutation produces unlikely data instances when features are correlated
- Adding a correlated feature can decrease the importance of the associated feature by splitting the importance between both features

Global surrogate model

Global surrogate model

Global surrogate model

- Train interpretable model that approximates a black box model
- Interpret surrogate model instead of black box model
  (e.g., linear model or decision tree)

Steps to obtain a surrogate model

1. Select a dataset $\mathbf{X}$
2. For the selected dataset $\mathbf{X}$, get the predictions $\hat{\mathbf{y}}$ of the black box model
3. Select an interpretable model type (e.g., linear model, decision tree)
4. Train the interpretable surrogate model on the dataset $\mathbf{X}$ and its predictions $\hat{\mathbf{y}}$
5. Measure how well the surrogate model replicates the predictions of the black box model (e.g., use R-squared measure, see next slide)
6. Interpret the surrogate model

R-squared measure

Measure how well the surrogate replicates the black box model:

$$R^2 = 1 - \frac{\mathrm{SSE}}{\mathrm{SST}} = 1 - \frac{\sum_{i=1}^{n}\left(\hat{y}_*^{(i)} - \hat{y}^{(i)}\right)^2}{\sum_{i=1}^{n}\left(\hat{y}^{(i)} - \bar{\hat{y}}\right)^2}$$

$\hat{y}_*^{(i)}$ prediction for the i-th instance of the surrogate model

$\hat{y}^{(i)}$ prediction for the i-th of the black box model

$\bar{\hat{y}}$ mean of the black box model predictions

SSE sum of squares error

SST sum of squares total

R-squared measure: "percentage of variance that is captured by the surrogate model"

- If close to 1 (= low SSE) → interpretable model approximates the black box model well
- If close to 0 (= high SSE) → interpretable model fails to approximate the black box

Global surrogate model
Advantages

Flexible

- Any interpretable model can be used (e.g., linear model, decision tree / rules)
- Black box model / interpretable model can easily be exchanged
- Multiple surrogate models can be used on top of the same black box model (for different audiences)

Intuitive

- Easy to implement
- Easy to explain to people

R-squared measure

- Measure how good surrogate models approximates the black box model

Global surrogate model
Disadvantages

Approximates the model (and not the data)

- Draws conclusions about the model and not about the data
(since the surrogate model never sees the real outcome)
- Unclear what the best cut-off for R-squared is
- Surrogate model might approximate the original model well on one subset of the data but not for another subset

Surrogate model comes with all its own advantages and disadvantages

- See chapter on interpretable models

References

- **Friedman**, Jerome H. "Greedy function approximation: a gradient boosting machine." *Annals of statistics* (**2001**): 1189-1232.
- **Greenwell**, Brandon M., Bradley C. Boehmke, and Andrew J. McCarthy. "A simple and effective model-based variable importance measure." *arXiv preprint arXiv:1805.04755* (**2018**).