PADERBORN UNIVERSITY

Shapley Values

Explainable Artificial Intelligence

Dr. Stefan Heindorf

Outlook

- Shapley values
- SHAP

Shapley Values

Lloyd Shapley wins 2012 Nobel Prize in economics

Lloyd Shapley. Photo by Bengt Nyman (https://commons.wikimedia.org/wiki/File:Lloyd_Shapley_2_2012.jpg), „Lloyd Shapley 2 2012“, https://creativecommons.org/licenses/by/2.0/

Shapley values
Introduction

Shapley values come from cooperative game theory

A coalition game is defined as:

- There is a set {1,…,p} of p players and a characteristic function val that maps subsets of players to the real numbers: $val: 2^N \rightarrow \mathbb{R}$, with $val(\emptyset) = 0$
- If S is a coalition of players, then $val(S)$, called the value of coalition S, describes the total expected sum of payoffs the members of S can obtain by cooperation

Example: 3 Players

- $val(\{\}) \ = \ 0$
- $val(\{a\}) \ = \ 300$
- $val(\{b\}) \ = \ 300$
- $val(\{c\}) \ = \ 300$
- $val(\{a,b\}) \ = 700$
- $val(\{a,c\}) \ = 500$
- $val(\{b,c\}) \ = \ 400$
- $val(\{a,b,c\}) \ = \ 1{,}000$

Which player has the largest contribution to coalition $\{a,b,c\}$?

What is the contribution of player a to the coalition $\{a,b,c\}$?

Shapley values
Introduction

Assumptions

- Empty coalition {} has value 0
- Players join the coalition one after the other
- Permutation $[a, b, c]$ indicates that the players join in this order

Example

- $[a, b, c]$: $val(\{a\}) - val(\{\}) = 300 - 0 = 300$
- $[a, c, b]$: $val(\{a\}) - val(\{\}) = 300 - 0 = 300$
- $[c, a, b]$: $val(\{a, c\}) - val(\{c\}) = 500 - 300 = 200$
- $[b, a, c]$: $val(\{a, b\}) - val(\{b\}) = 700 - 300 = 400$
- $[b, c, a]$: $val(\{a, b, c\}) - val(\{b, c\}) = 1{,}000 - 400 = 600$
- $[c, b, a]$: $val(\{a, b, c\}) - val(\{b, c\}) = 1{,}000 - 400 = 600$

What is the contribution of player a?

- Average contribution to each permutation:
$$\phi_a = (300 + 300 + 200 + 400 + 600 + 600) / 6 = 400$$
- Let P be the set of all permutations of the set of players $\{1, \dots, p\}$
- Let S be the players joining before j in a given permutation
- Average contribution of j to each permutation: $\phi_j = \frac{1}{p!} \sum_P (val(S \cup \{j\}) - val(S))$

Shapley values
Introduction

Observation

- In many permutations a has the same contribution, e.g.,
  - $[a, b, c]$: $val(\{a\}) - val(\{\}) = 300 - 0 = 300$
  - $[a, c, b]$: $val(\{a\}) - val(\{\}) = 300 - 0 = 300$
- The difference in values $val(\{a\}) - val(\{\})$ only needs to be computed once
- However, then we need to assign different weights to this difference

Example

- The difference $val(\{a\}) - val(\{\})$ receives the weight 2 / 6
- The difference $val(\{a, b\}) - val(\{b\})$ receives the weight 1 / 6

In general

- Given a set of players $S \subseteq \{1, \dots, p\} \setminus \{j\}$ that does not contain player j. In how many permutations of the set of players {1, …, p} do the players in S join the coalition before j and the other players join the coalition after j?
  - If $|S| = 0$, i.e., player j is the first player: $0!\,(p - 1)!$
  - If $|S| = 1$, i.e., player j is the second player: $1!\,(p - 2)!$
  - If $|S| = 2$, i.e., player j is the third player: $2!\,(p - 3)!$
  - In general, if player j joins the coalition S: $|S|!\,(p - |S| - 1)!$

The Shapley value
Definition

The Shapley value of a player *j* is its contribution to the payout

$$\phi_j(val) := \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \underbrace{\frac{|S|!\,(p - |S| - 1)!}{p!}}_{\text{weight}} \underbrace{(val(S \cup \{j\}) - val(S))}_{\text{marginal contribution}}$$

Intuition

- The players enter a room in random order
- All players in the room participate in the game (= contribute to the payoff)
- The Shapley value of a player is the average change in payout that the coalition already in the room receives when the player joins them

Notation

S	subset of the players forming a coalition
p	number of players

How is the weight determined?

$\|S\|!$	Number of ways players S can join before player j
$(p - \|S\| - 1)!$	Number of ways other players can join after $S \cup \{j\}$
$p!$	Number of ways to form coalition of p players

Shapley values
Fairness axioms and uniqueness

Efficiency: the sum of the contributions of all players give the total gain:

$$\sum_{j=1}^{p} \phi_{\mathrm{j}}(val) = val(\{1, \dots, p\})$$

Symmetry: If two players i, j are interchangeable, i.e., $val(S \cup \{i\}) = val(S \cup \{j\})$ for every coalition S not containing i and j, then

$$\phi_{\mathrm{i}}(val) = \phi_{\mathrm{j}}(val)$$

Null player: If a player i contributes no value, i.e., $val(S) = val(S \cup \{i\})$ for every coalition S not containing i, then

$$\phi_{\mathrm{i}}(val) = 0$$

Linearity: Suppose that val_1 and val_2 are two different value functions for a game. The gains for linear combinations of games behave linearly, i.e.,

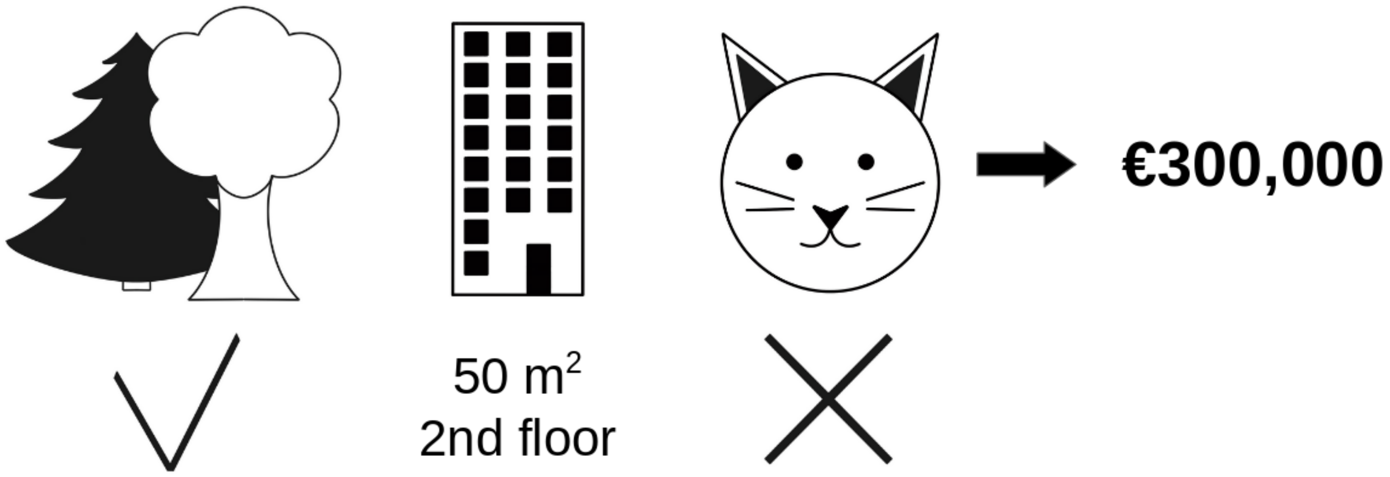
$$\phi(c_1 val_1 + c_2 val_2) = c_1 \phi(val_1) + c_2 \phi(val_2)$$

→ The Shapley value is the only function $\phi: val \rightarrow \mathbb{R}^p$ satisfying these properties [Shapley 1953]

How can Shapley values be used to quantify the importance of feature values?

Shapley values for feature importance

- Given: single instance $\mathbf{x} = (x_1, \dots, x_p)$ to explain
- Players = feature values x_j
- Shapley values indicate how much each feature value contributes to the difference between the prediction for $\mathbf{x}$ and the average prediction (remember $val(\{\}) = 0$)



`Park-nearby, area-50, floor-2nd, cat-banned`

Average price of all apartments: €310,000

→ Explain difference €300,000 - €310,000 = - €10,000

Shapley values

Shapley values come from coalitional game theory

- The “game” is the prediction task for a single instance of the dataset
- The “gain” is the actual prediction for this instance minus the average prediction for all instances
- The “players” are the feature values of the instance
- The “coalition” are the feature values of the instance that collaborate to receive the gain (=predict a certain value)

Example

- Feature values `park-nearby`, `area-50`, `floor-2nd`, and `cat-banned`, worked together to achieve the prediction of €300,000
- Goal: explain the difference between the actual prediction (€300,000) and the average prediction (€310,000): a difference of **-€10,000.** The answer could be:
  - `park-nearby` contributed €30,000
  - `area-50` contributed €10,000
  - `floor-2nd` contributed €0
  - `cat-banned` contributed -€50,000

Problem: Missing values

Example

- How to compute the value of a coalition S like {`park-nearby`, `floor-2nd`} that does not contain `area-50` and `cat-banned`?
- More generally: given 4 feature values x_1, x_2, x_3, x_4 and coalition $S = \{1, 3\}$
- What is $\hat{f}_S(\mathbf{x}_S) = \hat{f}(x_1, NA, x_3, NA)$? How to deal with missing values (NA)?

What happens if a feature j is part of the coalition S?

- We take the value of the feature from instance $\mathbf{x}$, i.e., x_j
- We make a prediction $\hat{f}(\ldots, x_j, \ldots)$

What happens if a feature j is not part of the coalition S?

- We need to make a prediction for $\hat{f}(\ldots, NA, \ldots)$
- Option 1: Use model that can cope with NA values (most models cannot)
- Option 2: replace NA by average value for j in the whole training set
- Option 3: replace NA by the average value for j in the subset of the training set where the first feature has value x_1 and the third feature has value x_3
  (there is debate in the literature whether Option 2 or Option 3 is better,
  in the following we will describe Option 3 as proposed by Lundberg 2017)

The Shapley values

Value function for tabular data, “replace NA values with conditional distribution”

For tabular data, the value function can be defined as

(via conditional distribution from tabular dataset)

$$val_{\mathbf{x}}(S) = E_{\mathbf{X}_C|\mathbf{X}_S=\mathbf{x}_s}\left[\hat{f}(\mathbf{x}_S, \mathbf{X}_C\ )\right] - E_{\mathbf{X}}\left[\hat{f}(\mathbf{X})\right]$$
$$= \sum_{\mathbf{x}_C \in \mathbf{X}_C} \Pr(\mathbf{X}_C = \mathbf{x}_C | \mathbf{X}_S = \mathbf{x}_s)\, \hat{f}(\mathbf{x}_S, \mathbf{x}_C) - \sum_{\mathbf{x}\in\mathbf{X}} \hat{f}(\mathbf{x})$$

Example: Given 4 features x_1, x_2, x_3, x_4 and coalition $S = \{1, 3\}$

$$val_{\mathbf{x}}(\{1,3\}) = \sum_{x_2 \in X_2} \sum_{x_4 \in X_4} Pr(X_2 = x_2; X_4 = x_4 \mid X_1 = x_1; X_3 = x_3)\, \hat{f}(x_1, x_2, x_3, x_4) - \sum_{\mathbf{x}\in\mathbf{X}} \hat{f}(\mathbf{x})$$

$val_{\mathbf{x}}$	Characteristic function for instance $\mathbf{x}$
$C = \{1, \dots, p\} \setminus S$	Features not selected (complement)
$\mathbf{X} = [X_1, X_2, X_3, X_4]$	Dataset: concatenation of the columns X_1, X_2, X_3, X_4
$\mathbf{X}_S$, $\mathbf{X}_C$	Subset of columns from dataset $\mathbf{X}$ induced by S and its complement C
$\mathbf{x}_S$	Feature values from instance $\mathbf{x}$ for columns S
$\mathbf{x}_C \in \mathbf{X}_C$	Feature values $\mathbf{x}_C$ from $\mathbf{X}_C$
$PR(\dots)$	Empirical probability based on dataset $\mathbf{X}$

Example with two features

Simplified example with only two features. Training data

park-nearby	cat-banned	Prediction
No	No	300,000
No	Yes	220,000
Yes	No	400,000
Yes	**Yes**	**370,000**

What is the contribution of the feature value `cat-banned` to the last instance?

- Prediction of coalition: For tabular data: Impute missing values by the average prediction of all instances in coalition
  (e.g., coalition {} corresponds to all 4 instances, coalition {cat-banned} corresponds to instances 2 and 4)
- Value of coalition: predicted price minus average price (322,500)

Coalition	park-nearby	cat-banned	Prediction	Value
{}	NA	NA	322,500	0
{cat-banned}	NA	Yes	295,000	-27,500
{park-nearby}	Yes	NA	385,000	62,500
{park-nearby, cat-banned}	Yes	Yes	370,000	47,500

Example with two features: contribution of cat-banned

Coalition	park-nearby	cat-banned	Prediction	Value
{}	NA	NA	322,500	0
{cat-banned}	NA	Yes	295,000	-27,500
{park-nearby}	Yes	NA	385,000	62,500
{park-nearby, cat-banned}	Yes	Yes	370,000	47,500

- Compute marginal contribution of cat-banned to each coalition
- Average all marginal contributions

Coalition	value w/o cat-banned	value with cat-banned	Marginal contribution	Weight
{}	0	-27,500	-27,500	1/2
{park-nearby}	62,500	47,500	-15,000	1/2
Average			**-21,250**	

→ Shapley value of feature `cat-banned` is **-21,250**

What is the contribution of the feature park-nearby to the last instance?

Solution: contribution of park-nearby

Coalition	park-nearby	cat-banned	Prediction	Value
{}	NA	NA	322,500	0
{cat-banned}	NA	Yes	295,000	-27,500
{park-nearby}	Yes	NA	385,000	62,500
{park-nearby, cat-banned}	Yes	Yes	370,000	47,500

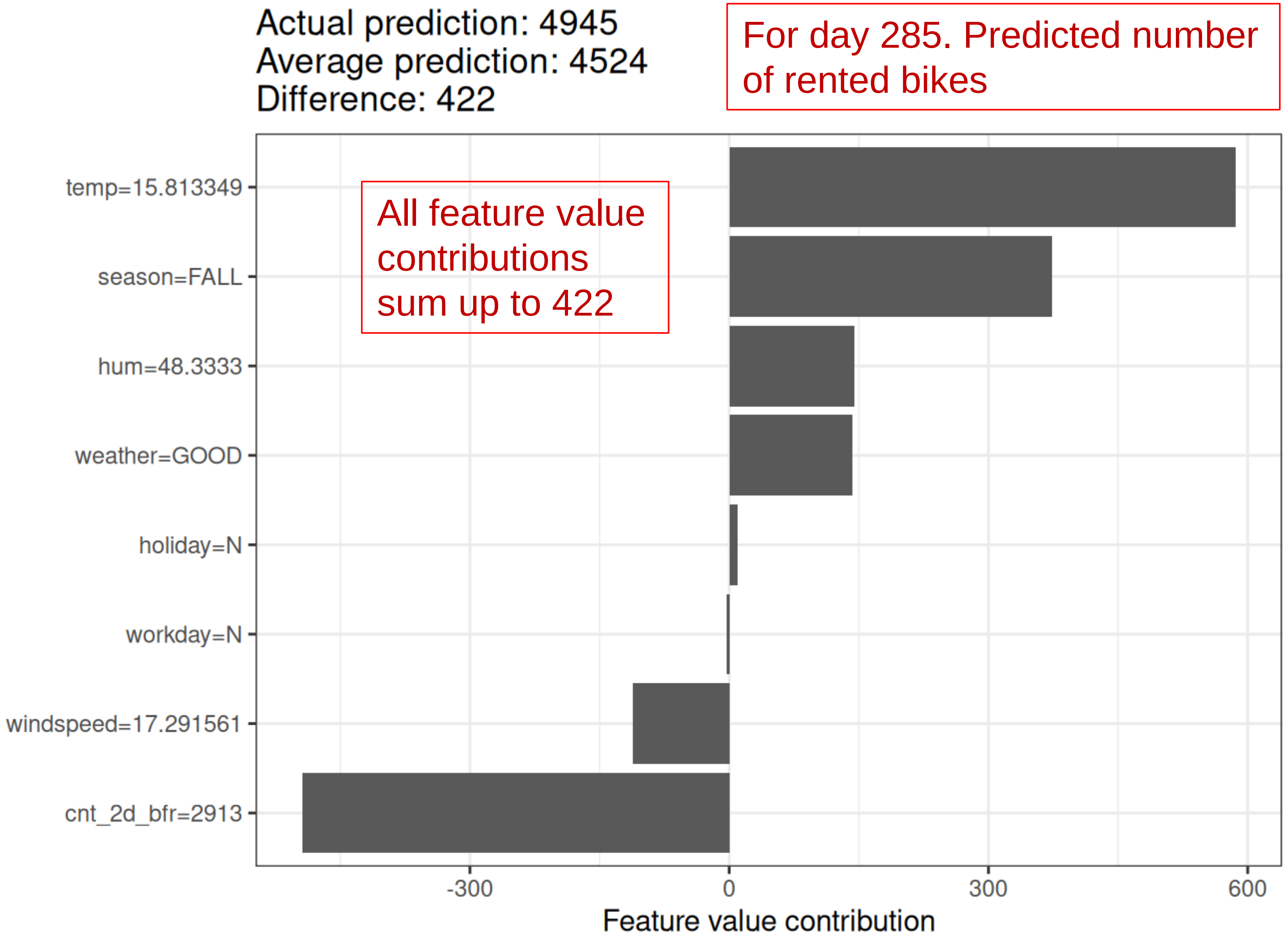
Coalition	value w/o park-nearby	value with park-nearby	Marginal contribution	Weight
{}	0	62,500	62,500	1/2
{cat-banned}	-27,500	47,500	75,000	1/2
Average			**68,750**	

To double check

- Prediction 370,000
- Average prediction: 322,500
- Contribution of feature value `cat-banned` -21,250
- Contribution of feature value `park-nearby` 68,750
- 322,500 - 21,250 + 68,750 = 370,000

Examples and interpretation

Bike rental dataset



Christoph Molnar (https://christophm.github.io/interpretable-ml-book/shapley_files/figure-html/fig-shapley-bike-plot-1.png), https://creativecommons.org/licenses/by-nc-sa/4.0/

Estimating the Shapley value

Method by Strumbelj, 2014

Exact calculation:

- All possible coalitions (sets) of feature values have to be evaluated with and without the j-th feature to calculate the exact Shapley value
- Problem: the number of possible coalitions exponentially increases as more features are added

Strumbelj et al. (2014) - approximation with Monte-Carlo sampling:

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \hat{f}\left(\mathbf{x}_{+j}^{m}\right) - \hat{f}\left(\mathbf{x}_{-j}^{m}\right)$$

where

$\mathbf{x}_{+j}^{m}$	Instance $\mathbf{x}$, but with a random number of feature values replaced by feature values from a random data point $\mathbf{z}$, except for the respective value of feature j
$\mathbf{x}_{-j}^{m}$	Identical to $\mathbf{x}_{+j}^{m}$, but the value $\mathbf{x}_{+j}^{m}$ is also taken from the sampled $\mathbf{z}$

Estimating the Shapley value

Method by Strumbelj, 2014

Output: Shapley value for the value of the j-th feature

Required: Number of iterations M, instance of interest $\mathbf{x}$, feature index j, data matrix $\mathbf{X}$, and machine learning model $\hat{f}$

For all $m = 1, \dots, M$**:**

- Draw random instance $\mathbf{z}$ from the data matrix $\mathbf{X}$
- Choose a random permutation o of the feature values
- Order instance $\mathbf{x}$: $\mathbf{x}_o = \left(x_{(1)}, \dots, x_{(j)}, \dots, x_{(p)}\right)$
- Order instance $\mathbf{z}$: $\mathbf{z}_o = \left(z_{(1)}, \dots, z_{(j)}, \dots, z_{(p)}\right)$
- Construct two new instances
  - With j: $\mathbf{x}_{+j} = \left(x_{(1)}, \dots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \dots, z_{(p)}\right)$
  - Without j: $\mathbf{x}_{-j} = \left(x_{(1)}, \dots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \dots, z_{(p)}\right)$
- Compute marginal contribution: $\phi_j^m = \hat{f}\left(\mathbf{x}_{+j}^m\right) - \hat{f}\left(\mathbf{x}_{-j}^m\right)$

Compute estimated Shapley value as the average: $\hat{\phi}_j(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$

Shapley values

Advantages

- Complete attribution. The difference between the actual prediction and the average prediction is fully explained by summing up all Shapley values of the values of the instances (efficiency property)
- Solid theory. The Shapley value is the only explanation method with a **solid theory**. The axioms – efficiency, symmetry, dummy, additivity – give the explanation a reasonable foundation.

Shapley values
Disadvantages

- Exact computation of Shapley values requires a lot of computing time
- Explanations created with the Shapley value method always use all the features (no sparse explanations)
- The Shapley value method returns a simple value per feature, but no prediction model like LIME. (This means it cannot be used to make statements about changes in prediction for changes in the input, such as: “If I were to earn €300 more a year, my credit score would increase by 5 points.”)
- You need access to the data to replace parts of the instance of interest with values from randomly drawn instances of the data. This can only be avoided if you can create data instances that look like real data instances but are not actual instances from the training data.
- The Shapley value method suffers from inclusion of unrealistic data instances when features are correlated (like many other permutation-based interpretation methods)

SHAP

SHapley Additive exPlanations

SHAP

Problem: Computing Shapley values is computationally expensive

Solution: Different approximation methods (with different trade-offs between speed, accuracy, and models they can be applied to)

- KernelSHAP (model-agnostic)
- TreeSHAP (model-specific for trees, not part of lecture)
- LinearSHAP (model-specific for linear models, not part of lecture)
- And many more

Additive feature attribution method

Definition 1 (additive feature attribution method):

$$g(\mathbf{z}') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

g	explanation model for original model $\hat{f}$ (such that $g(\mathbf{z}') \approx \hat{f}(\mathbf{z})$ when $\mathbf{z}' \approx \mathbf{z} = h_{\mathbf{x}}(\mathbf{z}')$)
$\mathbf{z}' \in \{0,1\}^M$	coalition vector (also called “simplified features”)
M	maximum coalition size
$\phi_j \in \mathbb{R}$	feature attribution for feature j, the Shapley values

Recap: LIME

$$g(\mathbf{z}') = \beta_0 + \sum_{j=1}^{M} \beta_j z'_j = \mathbf{z}'^{\mathrm{T}} \boldsymbol{\beta}$$

Idea

- Compute Shapley values similar to LIME

KernelSHAP

[Lundberg et al. 2017, Bagheri et al. 2022]

Preliminaries

- $\mathbf{x}' \in \{1\}^M$ special coalition vector of $\mathbf{x}$ where all elements are 1
- $h_{\mathbf{x}}: \{0,1\}^M \to F^M$ mapping from coalition vector to feature vector (that may contain NA values)

Theorem: If we set

$$\Omega(g) = 0$$

$$\pi_{\mathbf{x}}(\mathbf{z}') = \frac{M-1}{\binom{M}{|\mathbf{z}'|}|\mathbf{z}'|(M-|\mathbf{z}'|)}$$

$$L(\hat{f}, g, \pi_{\mathbf{x}}) = \sum_{\mathbf{z}' \subseteq \mathbf{x}'} [\hat{f}(h_{\mathbf{x}}(\mathbf{z}') - g(\mathbf{z}')]^2 \pi_{\mathbf{x}}(\mathbf{z}')$$

then the solution g to the LIME equation (g is a linear model)

$$\underset{g \in G}{arg\,min}\, L(\hat{f}, g, \pi_{\mathbf{x}}) + \Omega(g)$$

has coefficients that estimate the Shapley values.

Proof (will **not** be relevant for exam):

- Lundberg et al. 2017: Supplementary Material of Lundberg et al. 2017. https://papers.nips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Supplemental.zip
- Bagheri et al. 2022 : see Medium blog post in Panda

KernelSHAP
Model-agnostic approximation

Overview of Algorithm

1. Sample coalitions $\mathbf{z}'_k \in \{0,1\}^M, \;\; k \in \{1, \ldots, K\}$ (1=feature present in coalition, 0=feature absent)
2. Get prediction for each $\mathbf{z}'_k$ by first converting $\mathbf{z}'_k$ to the original feature space and then applying model $\hat{f}$: $\hat{f}(h_{\mathbf{x}}(\mathbf{z}'_k))$
3. Compute the weight $\pi_{\mathbf{x}}(\mathbf{z}')$ for each $\mathbf{z}'_k$ with the SHAP kernel
4. Fit weighted linear model
5. Return Shapley values ϕ_k, the coefficients from the linear model

SHAP kernel

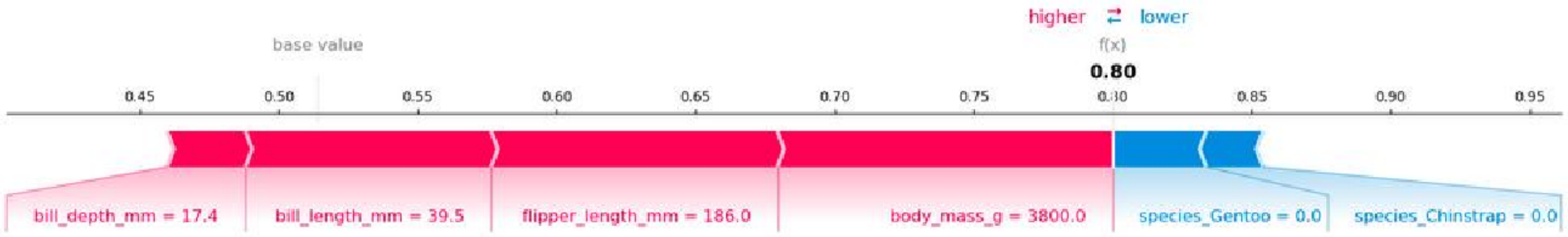
$$\pi_x(\mathbf{z}') = \frac{M-1}{\binom{M}{|\mathbf{z}'|}|\mathbf{z}'|(M-|\mathbf{z}'|)}$$

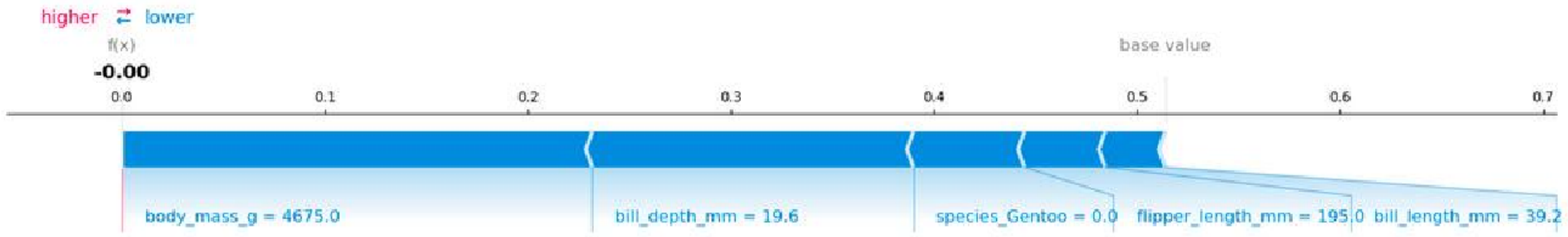
Advantage of KernelSHAP (over estimation method by Strumbelj, 2014)

- Less evaluations of black box model $\hat{f}$ necessary

Force plots

Example: Prediction of Penguin P(Adelie)

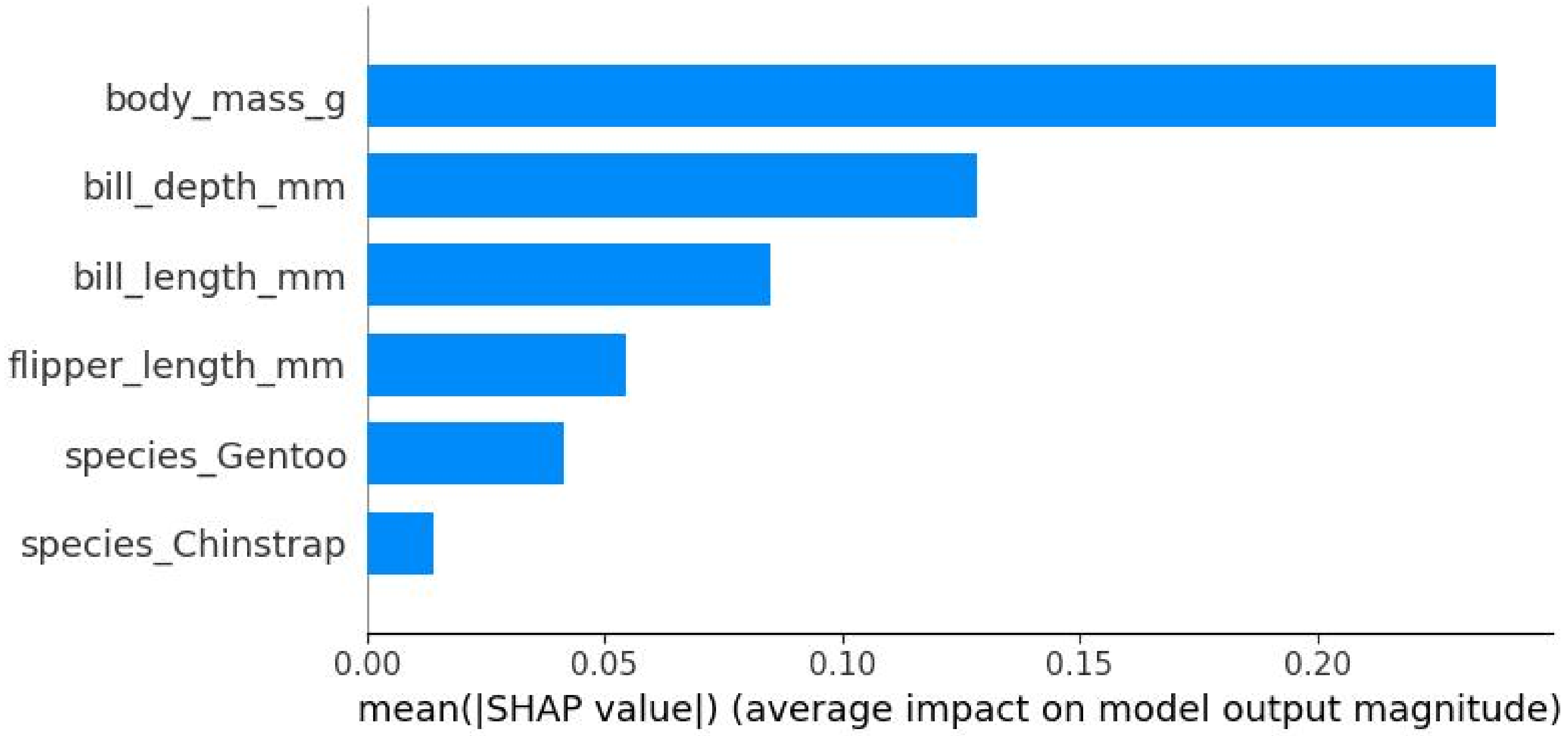




Christoph Molnar (https://christophm.github.io/interpretable-ml-book/shap_files/figure-html/fig-force-plots-1.png), https://creativecommons.org/licenses/by-nc-sa/4.0/

SHAP feature importance

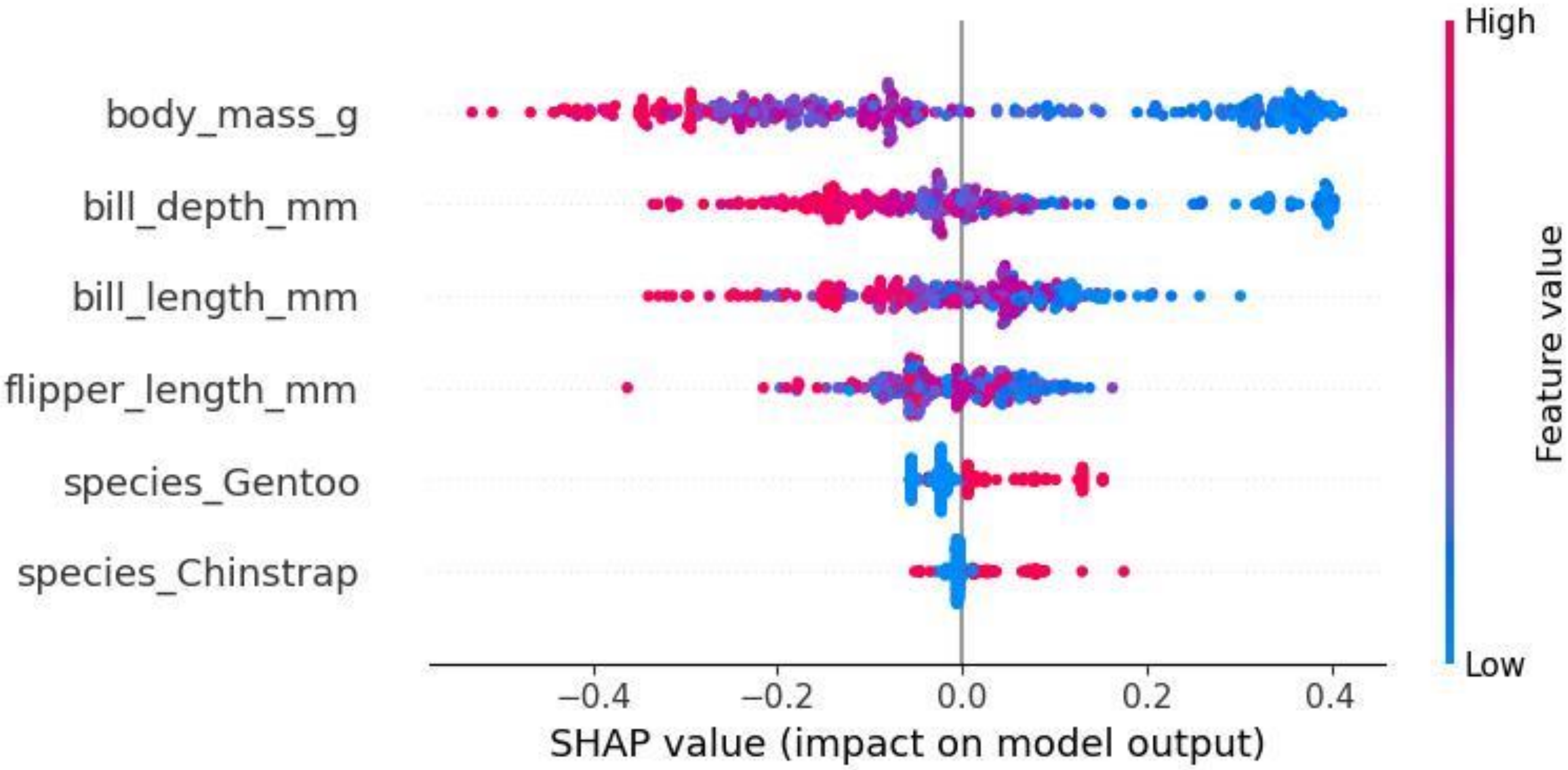
Global importance is obtained by averaging local importances



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

SHAP summary plot

point: Shapley value for a feature and an instance, color: value of the feature



- Each point is a Shapley value for a feature and an instance
- Points are jittered in y-axis direction to visualize distribution

SHAP
Advantages

SHAP has solid theoretical foundation in game theory

- Prediction is fairly distributed among the feature values
- Contrastive explanations comparing prediction with average prediction

SHAP connects LIME and Shapley values

- Useful to better understand both methods
- Helps to unify the field of interpretable machine learning

SHAP has a fast implementation for tree-based models

SHAP allows global model interpretations

- By computing SHAP value for each instance and aggregating results
- Global interpretations are consistent with local explanations
- If you use LIME for local explanations and partial dependence plots plus permutation feature importance for global explanations, you lack a common foundation

SHAP
Disadvantages

KernelSHAP is slow (compared to TreeSHAP)

- Impractical to use for many instances

Access to data is needed to compute Shapley values
(except for TreeSHAP)

References

- **Bagheri,** Reza**.** “Introduction to SHAP Values and their Application in Machine Learning.” Medium (**2022**). https://towardsdatascience.com/introduction-to-shap-values-and-their-application-in-machine-learning-8003718e6827
- **Lundberg**, Scott M., and Su-In Lee. “A unified approach to interpreting model predictions.” *Advances in neural information processing systems* 30 (**2017**).
- **Lundberg**, Scott M., Gabriel G. Erion, and Su-In Lee. "Consistent individualized feature attribution for tree ensembles." *arXiv preprint arXiv:1802.03888* (**2018**).
- **Molnar,** Christoph. “Interpretable machine learning.” (**2020**).
- **Shapley**, Lloyd S. "A value for n-person games." (**1953**): 307-317.