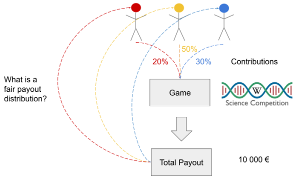
Interpretable Machine Learning

Shapley Values



Learning goals

- Learn what game theory is
- Understand the concept behind cooperative games
- Understand the Shapley value in game theory

COOPERATIVE GAMES IN GAME THEORY ▸ Shapley (1951)

- Game theory is the study of strategic games between players, “game” refers to any series of interactions between actors / agents with gains and losses of quantifiable utility value

COOPERATIVE GAMES IN GAME THEORY ▸ Shapley (1951)

- Game theory is the study of strategic games between players, “game” refers to any series of interactions between actors / agents with gains and losses of quantifiable utility value
- Cooperative games: For all possible players $P = \{1, \ldots, p\}$, each subset of players $S \subseteq P$ forms a coalition – each coalition S achieves a certain payout

COOPERATIVE GAMES IN GAME THEORY ▸ Shapley (1951)

- Game theory is the study of strategic games between players, "game" refers to any series of interactions between actors / agents with gains and losses of quantifiable utility value
- Cooperative games: For all possible players $P = \{1, \dots, p\}$, each subset of players $S \subseteq P$ forms a coalition – each coalition S achieves a certain payout
- A value function $v : 2^{|P|} \mapsto \mathbb{R}$ maps all $2^{|P|}$ possible coalitions to their payout (or gain)

COOPERATIVE GAMES IN GAME THEORY ▸ Shapley (1951)

- Game theory is the study of strategic games between players, “game” refers to any series of interactions between actors / agents with gains and losses of quantifiable utility value
- Cooperative games: For all possible players $P = \{1, \dots, p\}$, each subset of players $S \subseteq P$ forms a coalition – each coalition S achieves a certain payout
- A value function $v : 2^{|P|} \mapsto \mathbb{R}$ maps all $2^{|P|}$ possible coalitions to their payout (or gain)
- $v(S)$ is the payout of coalition $S \subseteq P$ (payout of empty coalition must be zero: $v(\emptyset) = 0$)

COOPERATIVE GAMES IN GAME THEORY ▸ Shapley (1951)

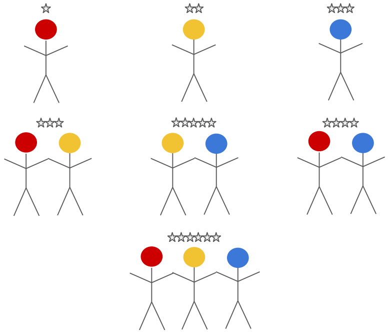
- Game theory is the study of strategic games between players, “game” refers to any series of interactions between actors / agents with gains and losses of quantifiable utility value
- Cooperative games: For all possible players $P = \{1, \ldots, p\}$, each subset of players $S \subseteq P$ forms a coalition – each coalition S achieves a certain payout
- A value function $v : 2^{|P|} \mapsto \mathbb{R}$ maps all $2^{|P|}$ possible coalitions to their payout (or gain)
- $v(S)$ is the payout of coalition $S \subseteq P$ (payout of empty coalition must be zero: $v(\emptyset) = 0$)
- As some players contribute more than others, we want to fairly divide the total achievable payout $v(P)$ among the players according to a player’s individual contribution

COOPERATIVE GAMES IN GAME THEORY ▸ Shapley (1951)

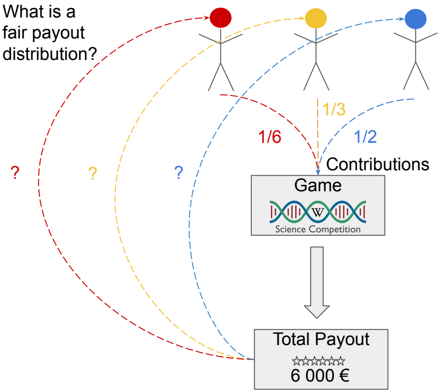
- Game theory is the study of strategic games between players, "game" refers to any series of interactions between actors / agents with gains and losses of quantifiable utility value
- Cooperative games: For all possible players $P = \{1, \dots, p\}$, each subset of players $S \subseteq P$ forms a coalition – each coalition S achieves a certain payout
- A value function $v : 2^{|P|} \mapsto \mathbb{R}$ maps all $2^{|P|}$ possible coalitions to their payout (or gain)
- $v(S)$ is the payout of coalition $S \subseteq P$ (payout of empty coalition must be zero: $v(\emptyset) = 0$)
- As some players contribute more than others, we want to fairly divide the total achievable payout $v(P)$ among the players according to a player's individual contribution
- We call the individual payout per player ϕ_j, $j \in P$ (later: Shapley value)

COOPERATIVE GAMES WITHOUT INTERACTIONS

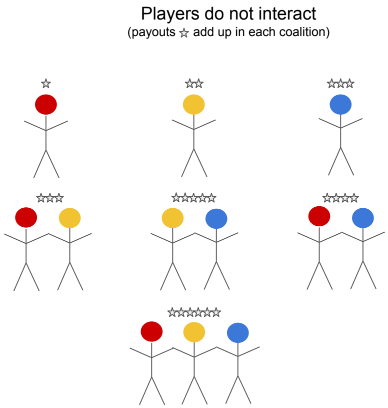
Players do not interact
(payouts ☆ add up in each coalition)

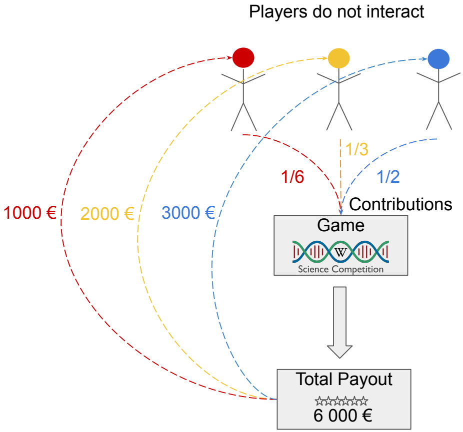


Players do not interact



COOPERATIVE GAMES WITHOUT INTERACTIONS

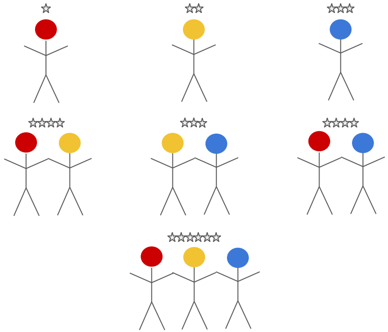


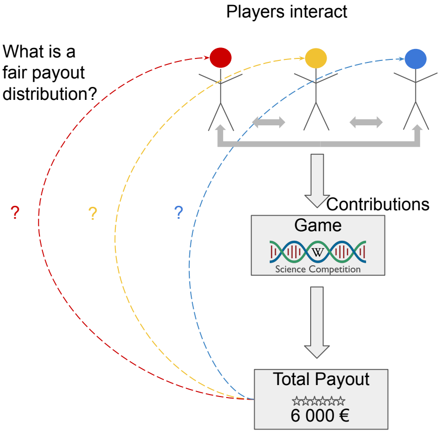


⇒ Fair Payouts are Trivial Without Interactions

COOPERATIVE GAMES WITH INTERACTIONS

Players interact
(payouts ☆ do not add up)



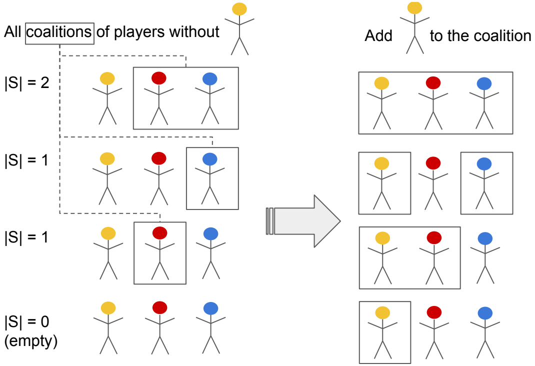


$\Rightarrow$ Unclear how to fairly distribute payouts when players interact

COOPERATIVE GAMES WITH INTERACTIONS

Question: What is a fair payout for player “yellow”?

Idea: Compute marginal contribution of the player of interest across different coalitions

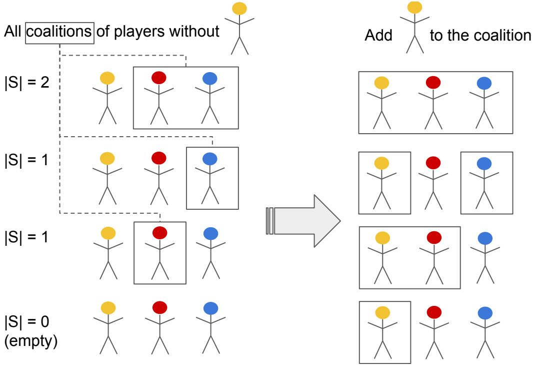


- Compute the total payout of each coalition
- Compute difference in payouts for each coalition with and without player “yellow” (= marginal contribution)
- Average marginal contributions using appropriate weights

COOPERATIVE GAMES WITH INTERACTIONS

Question: What is a fair payout for player "yellow"?

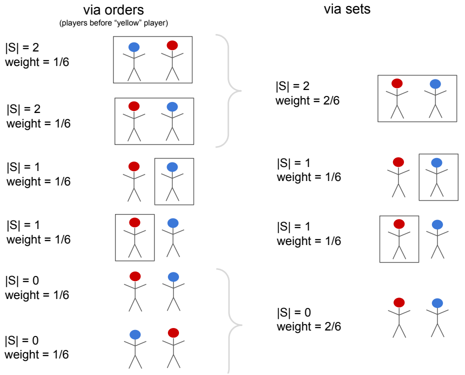
Idea: Compute marginal contribution of the player of interest across different coalitions



- Compute the total payout of each coalition
- Compute difference in payouts for each coalition with and without player "yellow" (= marginal contribution)
- Average marginal contributions using appropriate weights

Note: Each marginal contribution is weighted w.r.t. number of possible orders of its coalition

$\rightsquigarrow$ More players in $S \Rightarrow$ more orderings of S



SHAPLEY VALUE - SET DEFINITION

This idea refers to the **Shapley value** which assigns a payout value to each player according to its marginal contribution in all possible coalitions.

- Let $v(S \cup \{j\}) - v(S)$ be the marginal contribution of player j to coalition S
  $\rightsquigarrow$ measures how much a player j increases the value of a coalition S

SHAPLEY VALUE - SET DEFINITION

This idea refers to the **Shapley value** which assigns a payout value to each player according to its marginal contribution in all possible coalitions.

- Let $v(S \cup \{j\}) - v(S)$ be the marginal contribution of player j to coalition S
  $\leadsto$ measures how much a player j increases the value of a coalition S
- Average marginal contributions for all possible coalitions $S \subseteq P \setminus \{j\}$
  $\leadsto$ order of how players join the coalition matters $\Rightarrow$ different weights depending on size of S

SHAPLEY VALUE - SET DEFINITION

This idea refers to the **Shapley value** which assigns a payout value to each player according to its marginal contribution in all possible coalitions.

- Let $v(S \cup \{j\}) - v(S)$ be the marginal contribution of player j to coalition S
  $\rightsquigarrow$ measures how much a player j increases the value of a coalition S
- Average marginal contributions for all possible coalitions $S \subseteq P \setminus \{j\}$
  $\rightsquigarrow$ order of how players join the coalition matters $\Rightarrow$ different weights depending on size of S
- Shapley value via **set definition** (weighting via multinomial coefficient):

$$\phi_j = \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} (v(S \cup \{j\}) - v(S))$$

SHAPLEY VALUE - ORDER DEFINITION

The Shapley value was introduced as summation over sets $S \subseteq P \setminus \{j\}$, but it can be equivalently defined as a summation of all orders of players:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: All possible orders of players (we have $|P|!$ in total)

SHAPLEY VALUE - ORDER DEFINITION

The Shapley value was introduced as summation over sets $S \subseteq P \setminus \{j\}$, but it can be equivalently defined as a summation of all orders of players:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: All possible orders of players (we have $|P|!$ in total)
- S_j^τ: Set of players before player j in order $\tau = (\tau^{(1)}, \ldots, \tau^{(p)})$ where $\tau^{(i)}$ is i-th element
  $\Rightarrow$ Example: Players $1, 2, 3 \Rightarrow \Pi = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
  $\rightsquigarrow$ For order $\tau = (2, 1, 3)$ and player of interest $j = 3 \Rightarrow S_j^\tau = \{2, 1\}$
  $\rightsquigarrow$ For order $\tau = (3, 1, 2)$ and player of interest $j = 1 \Rightarrow S_j^\tau = \{3\}$
  $\rightsquigarrow$ For order $\tau = (3, 1, 2)$ and player of interest $j = 3 \Rightarrow S_j^\tau = \emptyset$

SHAPLEY VALUE - ORDER DEFINITION

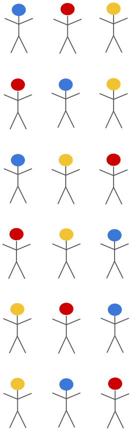
The Shapley value was introduced as summation over sets $S \subseteq P \setminus \{j\}$, but it can be equivalently defined as a summation of all orders of players:

$$\phi_j = \frac{1}{|P|!} \sum_{\tau \in \Pi} (v(S_j^\tau \cup \{j\}) - v(S_j^\tau))$$

- Π: All possible orders of players (we have $|P|!$ in total)
- S_j^τ: Set of players before player j in order $\tau = (\tau^{(1)}, \dots, \tau^{(p)})$ where $\tau^{(i)}$ is i-th element
  $\Rightarrow$ Example: Players $1, 2, 3 \Rightarrow \Pi = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$
  - $\rightsquigarrow$ For order $\tau = (2, 1, 3)$ and player of interest $j = 3 \Rightarrow S_j^\tau = \{2, 1\}$
  - $\rightsquigarrow$ For order $\tau = (3, 1, 2)$ and player of interest $j = 1 \Rightarrow S_j^\tau = \{3\}$
  - $\rightsquigarrow$ For order $\tau = (3, 1, 2)$ and player of interest $j = 3 \Rightarrow S_j^\tau = \emptyset$
- Order definition: Marginal contribution of orders that yield set $S = \{1, 2\}$ is summed twice
  $\rightsquigarrow$ In set definition, it has the weight $\frac{2!(3-2-1)!}{3!} = \frac{2 \cdot 0!}{6} = \frac{2}{6}$

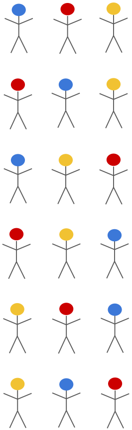
WEIGHTS FOR MARGINAL CONTRIBUTION - ILLUSTRATION

|P|! = 6 orders



WEIGHTS FOR MARGINAL CONTRIBUTION - ILLUSTRATION

|P|! = 6 orders



via orders
(players before “yellow” player)

|S| = 2
weight = 1/6

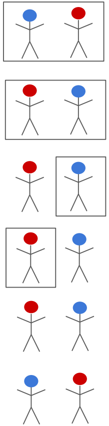
|S| = 2
weight = 1/6

|S| = 1
weight = 1/6

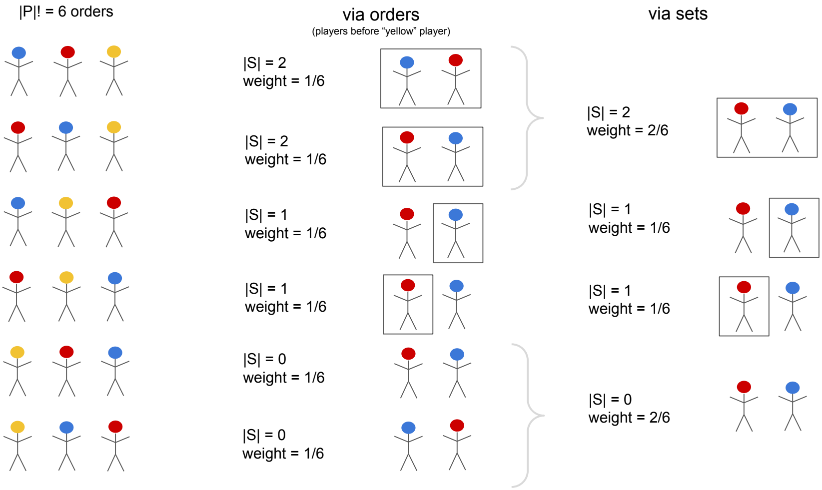
|S| = 1
weight = 1/6

|S| = 0
weight = 1/6

|S| = 0
weight = 1/6

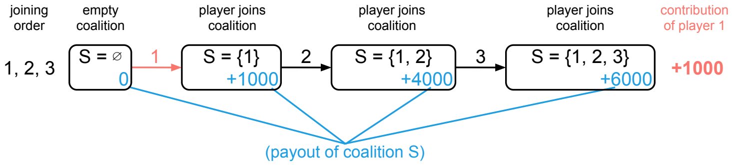


WEIGHTS FOR MARGINAL CONTRIBUTION - ILLUSTRATION



SHAPLEY VALUES - ILLUSTRATION

- Shapley value of player *j* is the marginal contribution to the value when it enters any coalition
- Produce all possible joining orders of player coalitions



SHAPLEY VALUES - ILLUSTRATION

- Shapley value of player *j* is the marginal contribution to the value when it enters any coalition
- Produce all possible joining orders of player coalitions
- Measure and average the difference in payout after player 1 enters the coalition

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 1
1, 2, 3	S = ∅ 0	1 →	S = {1} +1000	2 →	S = {1, 2} +4000	3 →	S = {1, 2, 3} +6000	**+1000**
1, 3, 2	S = ∅ 0	1 →	S = {1} +1000	3 →	S = {1, 3} +4000	2 →	S = {1, 2, 3} +6000	**+1000**
2, 1, 3	S = ∅ 0	2 →	S = {2} +2000	1 →	S = {1, 2} +4000	3 →	S = {1, 2, 3} +6000	**+2000**
2, 3, 1	S = ∅ 0	2 →	S = {2} +2000	3 →	S = {2, 3} +3000	1 →	S = {1, 2, 3} +6000	**+3000**
3, 1, 2	S = ∅ 0	3 →	S = {3} +3000	1 →	S = {1, 3} +4000	2 →	S = {1, 2, 3} +6000	**+1000**
3, 2, 1	S = ∅ 0	3 →	S = {3} +3000	2 →	S = {2, 3} +3000	1 →	S = {1, 2, 3} +6000	**+3000**

Shapley value of player 1 : +1833.33

SHAPLEY VALUES - ILLUSTRATION

- Shapley value of player *j* is the marginal contribution to the value when it enters any coalition
- Produce all possible joining orders of player coalitions
- Measure and average the difference in payout after player 2 enters the coalition

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 2
1, 2, 3	S = ∅ 0	1 →	S = {1} +1000	2 →	S = {1, 2} +4000	3 →	S = {1, 2, 3} +6000	**+3000**
1, 3, 2	S = ∅ 0	1 →	S = {1} +1000	3 →	S = {1, 3} +4000	2 →	S = {1, 2, 3} +6000	**+2000**
2, 1, 3	S = ∅ 0	2 →	S = {2} +2000	1 →	S = {1, 2} +4000	3 →	S = {1, 2, 3} +6000	**+2000**
2, 3, 1	S = ∅ 0	2 →	S = {2} +2000	3 →	S = {2, 3} +3000	1 →	S = {1, 2, 3} +6000	**+2000**
3, 1, 2	S = ∅ 0	3 →	S = {3} +3000	1 →	S = {1, 3} +4000	2 →	S = {1, 2, 3} +6000	**+2000**
3, 2, 1	S = ∅ 0	3 →	S = {3} +3000	2 →	S = {2, 3} +3000	1 →	S = {1, 2, 3} +6000	**+0**

Shapley value of player 2 : +1833.33

SHAPLEY VALUES - ILLUSTRATION

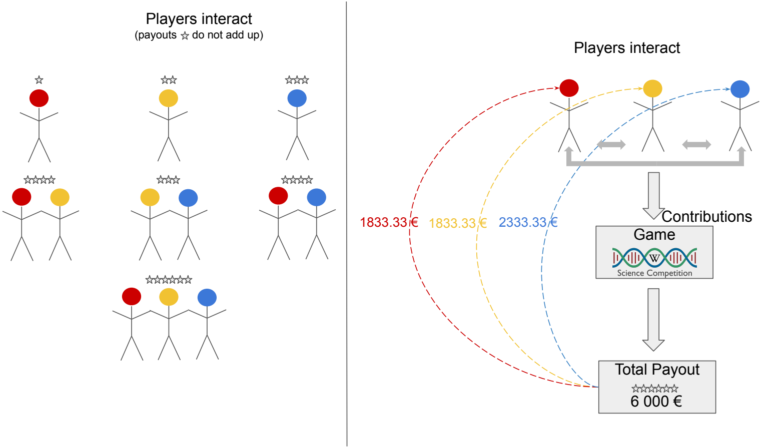
- Shapley value of player *j* is the marginal contribution to the value when it enters any coalition
- Produce all possible joining orders of player coalitions
- Measure and average the difference in payout after player 3 enters the coalition

joining order	empty coalition		player joins coalition		player joins coalition		player joins coalition	contribution of player 3
1, 2, 3	S = ∅ 0	1 →	S = {1} +1000	2 →	S = {1, 2} +4000	3 →	S = {1, 2, 3} +6000	**+2000**
1, 3, 2	S = ∅ 0	1 →	S = {1} +1000	3 →	S = {1, 3} +4000	2 →	S = {1, 2, 3} +6000	**+3000**
2, 1, 3	S = ∅ 0	2 →	S = {2} +2000	1 →	S = {1, 2} +4000	3 →	S = {1, 2, 3} +6000	**+2000**
2, 3, 1	S = ∅ 0	2 →	S = {2} +2000	3 →	S = {2, 3} +3000	1 →	S = {1, 2, 3} +6000	**+1000**
3, 1, 2	S = ∅ 0	3 →	S = {3} +3000	1 →	S = {1, 3} +4000	2 →	S = {1, 2, 3} +6000	**+3000**
3, 2, 1	S = ∅ 0	3 →	S = {3} +3000	2 →	S = {2, 3} +3000	1 →	S = {1, 2, 3} +6000	**+3000**

Shapley value of player 3: +2333.33

SHAPLEY VALUES - ILLUSTRATION

- Shapley value of player *j* is the marginal contribution to the value when it enters any coalition
- Produce all possible joining orders of player coalitions



AXIOMS OF FAIR PAYOUTS

Why is this a fair payout solution?
One possibility to define fair payouts are the following axioms for a given value function v:

- **Efficiency**: Player contributions add up to the total payout of the game: $\sum_{j=1}^{p} \phi_j = v(P)$

AXIOMS OF FAIR PAYOUTS

Why is this a fair payout solution?
One possibility to define fair payouts are the following axioms for a given value function v:

- **Efficiency**: Player contributions add up to the total payout of the game: $\sum_{j=1}^{p} \phi_j = v(P)$
- **Symmetry**: Players $j, k \in P$ who contribute the same to any coalition get the same payout:
  If $v(S \cup \{j\}) = v(S \cup \{k\})$ for all $S \subseteq P \setminus \{j, k\}$, then $\phi_j = \phi_k$

AXIOMS OF FAIR PAYOUTS

Why is this a fair payout solution?
One possibility to define fair payouts are the following axioms for a given value function v:

- **Efficiency**: Player contributions add up to the total payout of the game: $\sum_{j=1}^{p} \phi_j = v(P)$
- **Symmetry**: Players $j, k \in P$ who contribute the same to any coalition get the same payout:
  If $v(S \cup \{j\}) = v(S \cup \{k\})$ for all $S \subseteq P \setminus \{j, k\}$, then $\phi_j = \phi_k$
- **Dummy/Null Player**: Payout is 0 for players who don't contribute to the value of any coalition:
  If $v(S \cup \{j\}) = v(S) \quad \forall \quad S \subseteq P \setminus \{j\}$, then $\phi_j = 0$

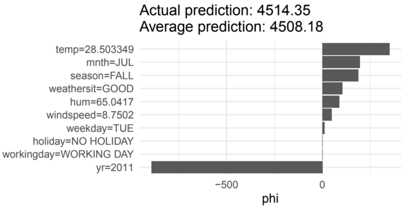
AXIOMS OF FAIR PAYOUTS

Why is this a fair payout solution?
One possibility to define fair payouts are the following axioms for a given value function v:

- **Efficiency**: Player contributions add up to the total payout of the game: $\sum_{j=1}^{p} \phi_j = v(P)$
- **Symmetry**: Players $j, k \in P$ who contribute the same to any coalition get the same payout: If $v(S \cup \{j\}) = v(S \cup \{k\})$ for all $S \subseteq P \setminus \{j, k\}$, then $\phi_j = \phi_k$
- **Dummy/Null Player**: Payout is 0 for players who don't contribute to the value of any coalition: If $v(S \cup \{j\}) = v(S) \quad \forall \quad S \subseteq P \setminus \{j\}$, then $\phi_j = 0$
- **Additivity**: For a game v with combined payouts $v(S) = v_1(S) + v_2(S)$, the payout is the sum of payouts: $\phi_{j,v} = \phi_{j,v_1} + \phi_{j,v_2}$

Interpretable Machine Learning

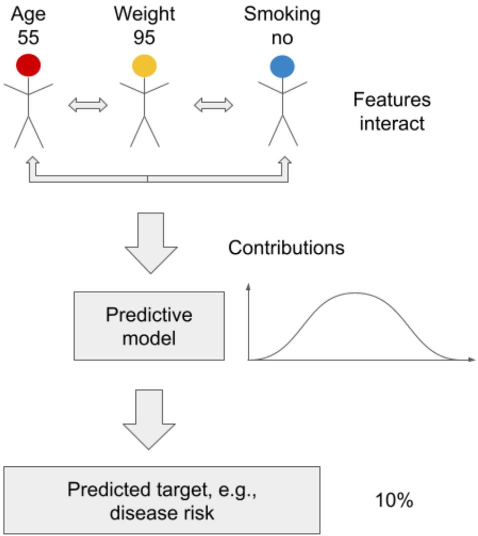
Shapley Values for Local Explanations



Learning goals

- See model predictions as a cooperative game
- Transfer the Shapley value concept from game theory to machine learning

FROM GAME THEORY TO MACHINE LEARNING



FROM GAME THEORY TO MACHINE LEARNING

- Game: Make prediction $\hat{f}(x_1, x_2, \ldots, x_p)$ for a single observation $\mathbf{x}$

FROM GAME THEORY TO MACHINE LEARNING

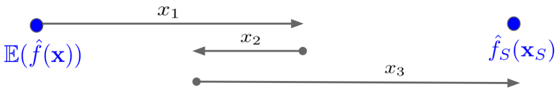
- Game: Make prediction $\hat{f}(x_1, x_2, \ldots, x_p)$ for a single observation $\mathbf{x}$
- Players: Features $x_j, j \in \{1, \ldots, p\}$ which cooperate to produce a prediction
  $\rightsquigarrow$ How can we make a prediction with a subset of features without changing the model?
  $\rightsquigarrow$ PD function: $\hat{f}_S(\mathbf{x}_S) := \int_{X_{-s}} \hat{f}(\mathbf{x}_S, X_{-S}) d\mathbb{P}_{X_{-s}}$ ("removing" by marginalizing over $-S$)

FROM GAME THEORY TO MACHINE LEARNING

- Game: Make prediction $\hat{f}(x_1, x_2, \ldots, x_p)$ for a single observation $\mathbf{x}$
- Players: Features $x_j, j \in \{1, \ldots, p\}$ which cooperate to produce a prediction
  $\leadsto$ How can we make a prediction with a subset of features without changing the model?
  $\leadsto$ PD function: $\hat{f}_S(\mathbf{x}_S) := \int_{X_{-s}} \hat{f}(\mathbf{x}_S, X_{-S}) d\mathbb{P}_{X_{-s}}$ ("removing" by marginalizing over $-S$)
- Value function / payout of coalition $S \subseteq P$ for observation $\mathbf{x}$:

$$v(S) = \hat{f}_S(\mathbf{x}_S) - \mathbb{E}_{\mathbf{x}}(\hat{f}(\mathbf{x})), \text{ where } \hat{f}_S : \mathcal{X}_S \mapsto \mathcal{Y}$$

$\leadsto$ subtraction of $\mathbb{E}_{\mathbf{x}}(\hat{f}(\mathbf{x}))$ ensures that v is a value function with $v(\emptyset) = 0$

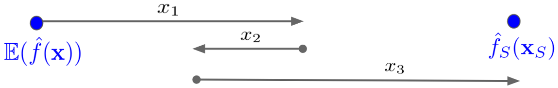


FROM GAME THEORY TO MACHINE LEARNING

- Game: Make prediction $\hat{f}(x_1, x_2, \ldots, x_p)$ for a single observation $\mathbf{x}$
- Players: Features $x_j, j \in \{1, \ldots, p\}$ which cooperate to produce a prediction
  $\leadsto$ How can we make a prediction with a subset of features without changing the model?
  $\leadsto$ PD function: $\hat{f}_S(\mathbf{x}_S) := \int_{X_{-S}} \hat{f}(\mathbf{x}_S, X_{-S}) d\mathbb{P}_{X_{-S}}$ ("removing" by marginalizing over $-S$)
- Value function / payout of coalition $S \subseteq P$ for observation $\mathbf{x}$:

$$v(S) = \hat{f}_S(\mathbf{x}_S) - \mathbb{E}_{\mathbf{x}}(\hat{f}(\mathbf{x})), \text{ where } \hat{f}_S : \mathcal{X}_S \mapsto \mathcal{Y}$$

$\leadsto$ subtraction of $\mathbb{E}_{\mathbf{x}}(\hat{f}(\mathbf{x}))$ ensures that v is a value function with $v(\emptyset) = 0$



- Marginal contribution: $v(S \cup \{j\}) - v(S) = \hat{f}_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) - \hat{f}_S(\mathbf{x}_S)$
  $\leadsto$ $\mathbb{E}_{\mathbf{x}}(\hat{f}(\mathbf{x}))$ cancels out due to the subtraction of value functions

SHAPLEY VALUE - DEFINITION ‣ Shapley (1953) ‣ Strumbelj et al. (2014)

Shapley value ϕ_j of feature j for observation $\mathbf{x}$ via **order definition**:

$$\phi_j(\mathbf{x}) = \frac{1}{|P|!} \sum_{\tau \in \Pi} \underbrace{\hat{f}_{S_j^\tau \cup \{j\}}(\mathbf{x}_{S_j^\tau \cup \{j\}}) - \hat{f}_{S_j^\tau}(\mathbf{x}_{S_j^\tau})}_{\text{marginal contribution of feature } j}$$

- Interpretation: Feature x_j contributed ϕ_j to difference between $\hat{f}(\mathbf{x})$ and average prediction
  $\rightsquigarrow$ Note: Marginal contributions and Shapley values can be negative
- For exact computation of $\phi_j(\mathbf{x})$, the PD function $\hat{f}_S(\mathbf{x}_S) = \frac{1}{n}\sum_{i=1}^n \hat{f}(\mathbf{x}_S, \mathbf{x}^{(i)}_{-S})$ for any set of features S can be used which yields

$$\phi_j(\mathbf{x}) = \frac{1}{|P|! \cdot n} \sum_{\tau \in \Pi} \sum_{i=1}^{n} \hat{f}(\mathbf{x}_{S_j^\tau \cup \{j\}}, \mathbf{x}^{(i)}_{-\{S_j^\tau \cup \{j\}\}}) - \hat{f}(\mathbf{x}_{S_j^\tau}, \mathbf{x}^{(i)}_{-S_j^\tau})$$

$\rightsquigarrow$ Note: $\hat{f}_S$ marginalizes over all other features $-S$ using all observations $i = 1, \ldots, n$

ESTIMATION: A PRACTICAL PROBLEM

- Exact Shapley value computation is problematic for high-dimensional feature spaces
$\rightsquigarrow$ For 10 features, there are already $|P|! = 10! \approx 3.6$ million possible orders of features

ESTIMATION: A PRACTICAL PROBLEM

- Exact Shapley value computation is problematic for high-dimensional feature spaces
  $\leadsto$ For 10 features, there are already $|P|! = 10! \approx 3.6$ million possible orders of features
- Additional problem due to estimation of the marginal prediction $\hat{f}_{S_j^\tau}$: Averaging over the entire data set for each coalition S_j^τ introduced by τ can be very expensive for large data sets

ESTIMATION: A PRACTICAL PROBLEM

- Exact Shapley value computation is problematic for high-dimensional feature spaces
  $\rightsquigarrow$ For 10 features, there are already $|P|! = 10! \approx 3.6$ million possible orders of features
- Additional problem due to estimation of the marginal prediction $\hat{f}_{S_j^\tau}$: Averaging over the entire data set for each coalition S_j^τ introduced by τ can be very expensive for large data sets
- Solution to both problems is sampling: Instead of averaging over $|P|! \cdot n$ terms, we approximate it using a limited amount of M random samples of τ to build coalitions S_j^τ

ESTIMATION: A PRACTICAL PROBLEM

- Exact Shapley value computation is problematic for high-dimensional feature spaces
  $\rightsquigarrow$ For 10 features, there are already $|P|! = 10! \approx 3.6$ million possible orders of features
- Additional problem due to estimation of the marginal prediction $\hat{f}_{S_j^\tau}$: Averaging over the entire data set for each coalition S_j^τ introduced by τ can be very expensive for large data sets
- Solution to both problems is sampling: Instead of averaging over $|P|! \cdot n$ terms, we approximate it using a limited amount of M random samples of τ to build coalitions S_j^τ
- M is a tradeoff between accuracy of the Shapley value and computational costs
  $\rightsquigarrow$ The higher M, the closer to the exact Shapley values, but the more costly the computation

SHAPLEY VALUE APPROXIMATION - ILLUSTRATION

Definition

x: obs. of interest

x with feature values in S_m (other are replaced)

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

x with feature values in $S_m \cup \{j\}$

	Temperature	Humidity	Windspeed	Year
x	10.66	56	11	2012
x_{+j}	10.66	56	$random : z_{windspeed}^{(m)}$	2012
x_{-j}	10.66	56	$random : z_{windspeed}^{(m)}$	$random : z_{year}^{(m)}$

j

SHAPLEY VALUE APPROXIMATION - ILLUSTRATION

Definition

$$\phi_j(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} \underbrace{\left[\hat{f}(\mathbf{x}_{+j}{}^{(m)}) - \hat{f}(\mathbf{x}_{-j}{}^{(m)})\right]}_{:=\Delta(j, S_m)}$$

Contribution of feature j to coalition S_m

- $\Delta(j, S_m) = \hat{f}(\mathbf{x}_{+j}^{(m)}) - \hat{f}(\mathbf{x}_{-j}^{(m)})$ is the marginal contribution of feature j to coalition S_m
- Here: Feature *year* contributes +700 bike rentals if it joins coalition $S_m = \{temp, hum\}$

	Temperature	Humidity	Windspeed	Year	Count
x	10.66	56	11	2012	
x_{+j}	10.66	56	$random : z_{windspeed}^{(m)}$	2012	5600
x_{-j}	10.66	56	$random : z_{windspeed}^{(m)}$	$random : z_{year}^{(m)}$	4900

j $\quad$ $\hat{f}$

700

$\Delta(j, S_m)$ marginal contribution

SHAPLEY VALUE APPROXIMATION - ILLUSTRATION

Definition

average the contributions of feature j

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

- Compute marginal contribution of feature j towards the prediction across all randomly drawn feature coalitions $S_1, \dots, S_m$
- Average all M marginal contributions of feature j
- Shapley value ϕ_j is the payout of feature j, i.e., how much feature *year* contributed to the overall prediction in bicycle counts of a specific observation $\mathbf{x}$

$m =$ 1, 2, ..., M

$$\underbrace{700 \quad 503 \quad \dots \quad 300}_{\Delta(j, S_m)}$$

Shapley value

$$\underbrace{501}_{\phi_j}$$

REVISITED: AXIOMS FOR FAIR ATTRIBUTIONS

We take the general axioms for Shapley Values and apply it to predictions:

- **Efficiency**: Shapley values add up to the (centered) prediction: $\sum_{j=1}^{p} \phi_j = \hat{f}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}}(\hat{f}(X))$

REVISITED: AXIOMS FOR FAIR ATTRIBUTIONS

We take the general axioms for Shapley Values and apply it to predictions:

- **Efficiency**: Shapley values add up to the (centered) prediction: $\sum_{j=1}^{p} \phi_j = \hat{f}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}}(\hat{f}(X))$
- **Symmetry**: Two features j and k that contribute the same to the prediction get the same payout
  $\rightsquigarrow$ interaction effects between features are fairly divided
  $\hat{f}_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) = \hat{f}_{S \cup \{k\}}(\mathbf{x}_{S \cup \{k\}})$ for all $S \subseteq P \setminus \{j, k\}$ then $\phi_j = \phi_k$

REVISITED: AXIOMS FOR FAIR ATTRIBUTIONS

We take the general axioms for Shapley Values and apply it to predictions:

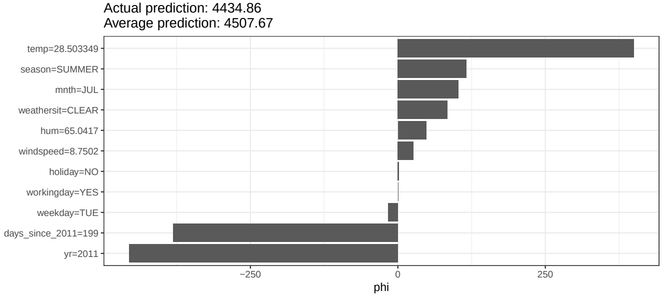
- **Efficiency**: Shapley values add up to the (centered) prediction: $\sum_{j=1}^{p} \phi_j = \hat{f}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}}(\hat{f}(X))$
- **Symmetry**: Two features j and k that contribute the same to the prediction get the same payout
  $\leadsto$ interaction effects between features are fairly divided
  $\hat{f}_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) = \hat{f}_{S \cup \{k\}}(\mathbf{x}_{S \cup \{k\}})$ for all $S \subseteq P \setminus \{j, k\}$ then $\phi_j = \phi_k$
- **Dummy / Null Player**: Shapley value of a feature that does not influence the prediction is zero
  $\leadsto$ if a feature was not selected by the model (e.g., tree or LASSO), its Shapley value is zero
  $\hat{f}_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) = \hat{f}_S(\mathbf{x}_S)$ for all $S \subseteq P$ then $\phi_j = 0$

REVISITED: AXIOMS FOR FAIR ATTRIBUTIONS

We take the general axioms for Shapley Values and apply it to predictions:

- **Efficiency**: Shapley values add up to the (centered) prediction: $\sum_{j=1}^{p} \phi_j = \hat{f}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}}(\hat{f}(X))$
- **Symmetry**: Two features j and k that contribute the same to the prediction get the same payout
  $\rightsquigarrow$ interaction effects between features are fairly divided
  $\hat{f}_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) = \hat{f}_{S \cup \{k\}}(\mathbf{x}_{S \cup \{k\}})$ for all $S \subseteq P \setminus \{j, k\}$ then $\phi_j = \phi_k$
- **Dummy** / **Null Player**: Shapley value of a feature that does not influence the prediction is zero
  $\rightsquigarrow$ if a feature was not selected by the model (e.g., tree or LASSO), its Shapley value is zero
  $\hat{f}_{S \cup \{j\}}(\mathbf{x}_{S \cup \{j\}}) = \hat{f}_S(\mathbf{x}_S)$ for all $S \subseteq P$ then $\phi_j = 0$
- **Additivity**: For a prediction with combined payouts, the payout is the sum of payouts:
  $\phi_j(v_1) + \phi_j(v_2)$ $\rightsquigarrow$ Shapley values for model ensembles can be combined

BIKE SHARING DATASET



- Shapley values of observation $i = 200$ from the bike sharing data
- Difference between model prediction of this observation and the average prediction of the data is fairly distributed among the features (i.e., $4434 - 4507 \approx -73$)
- Feature value temp = 28.5 has the most positive effect, with a contribution (increase of prediction) of about +400

ADVANTAGES AND DISADVANTAGES

Advantages:

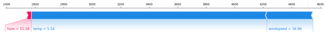
- **Solid theoretical foundation** in game theory
- Prediction is **fairly distributed** among the feature values $\rightsquigarrow$ easy to interpret for a user
- **Contrastive explanations** that compare the prediction with the average prediction

Disadvantages:

- Without sampling, Shapley values need a lot of computing time to inspect all possible coalitions
- Like many other IML methods, Shapley values suffer from the inclusion of unrealistic data observations when features are correlated

Interpretable Machine Learning

SHAP (SHapley Additive exPlanation) Values



Learning goals

- Get an intuition of additive feature attributions
- Understand the concept of Kernel SHAP
- Ability to interpret SHAP plots
- Global SHAP methods

FROM SHAPLEY TO SHAP

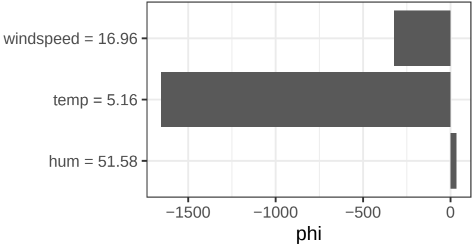
Remember: Shapley values explain the difference between actual and average prediction:

$$
\begin{aligned}
{\color{orange}2573} - {\color{blue}4515} &= \quad 34 - 1654 - 323 \quad = -1942 \\
\hat{f}(\mathbf{x}) - \mathbb{E}(\hat{f}) &= \quad \phi_{hum} + \phi_{temp} + \phi_{ws}
\end{aligned}
$$

$\leadsto$ can be rewritten to

$$\hat{f}(\mathbf{x}) = \underbrace{\mathbb{E}(\hat{f})}_{\phi_0} + \phi_{hum} + \phi_{temp} + \phi_{ws}$$

Actual prediction: 2572.67 ;
Average prediction: 4515.05



SHAP DEFINITION ▸ Lundberg et al. 2017

Aim: Find an additive combination that explains the prediction of an observation $\mathbf{x}$ by computing the contribution of each feature to the prediction using a (more efficient) estimation procedure.

Definition

- Define simplified (binary) coalition feature space $\mathbf{Z}' \in \{0, 1\}^{K \times p}$ with K rows and p columns
- Rows are referred to as $\mathbf{z}'^{(k)} = \{z_1'^{(k)}, \ldots, z_p'^{(k)}\}$ with $k \in \{1, \ldots, K\}$ (indexes k-th coalition)
- Columns are referred to as $\mathbf{z}_j$ with $j \in \{1, \ldots, p\}$ being the index of the original feature

Example:

Coalition	$\mathbf{z}'^{(k)}$	hum	temp	ws
$\emptyset$	$\mathbf{z}'^{(1)}$	0	0	0
hum	$\mathbf{z}'^{(2)}$	1	0	0
temp	$\mathbf{z}'^{(3)}$	0	1	0
ws	$\mathbf{z}'^{(4)}$	0	0	1
hum, temp	$\mathbf{z}'^{(5)}$	1	1	0
temp, ws	$\mathbf{z}'^{(6)}$	0	1	1
hum, ws	$\mathbf{z}'^{(7)}$	1	0	1
hum, temp, ws	$\mathbf{z}'^{(8)}$	1	1	1

SHAP DEFINITION ‣ Lundberg et al. 2017

Aim: Find an additive combination that explains the prediction of an observation $\mathbf{x}$ by computing the contribution of each feature to the prediction using a (more efficient) estimation procedure.

Definition

- Define simplified (binary) coalition feature space $\mathbf{Z}' \in \{0,1\}^{K \times p}$ with K rows and p columns
- Rows are referred to as $\mathbf{z}'^{(k)} = \{z_1'^{(k)}, \ldots, z_p'^{(k)}\}$ with $k \in \{1, \ldots, K\}$ (indexes k-th coalition)
- Columns are referred to as $\mathbf{z}_j$ with $j \in \{1, \ldots, p\}$ being the index of the original feature

$\mathbf{z}'^{(k)}$: **Coalition** simplified features

$$g\left(\mathbf{z}'^{(k)}\right) = \phi_0 + \sum_{j=1}^{p} \phi_j z_j'^{(k)}$$

SHAP DEFINITION ▸ Lundberg et al. 2017

Aim: Find an additive combination that explains the prediction of an observation $\mathbf{x}$ by computing the contribution of each feature to the prediction using a (more efficient) estimation procedure.

Definition

- Define simplified (binary) coalition feature space $\mathbf{Z}' \in \{0, 1\}^{K \times p}$ with K rows and p columns
- Rows are referred to as $\mathbf{z}'^{(k)} = \{z_1'^{(k)}, \ldots, z_p'^{(k)}\}$ with $k \in \{1, \ldots, K\}$ (indexes k-th coalition)
- Columns are referred to as $\mathbf{z}_j$ with $j \in \{1, \ldots, p\}$ being the index of the original feature

$\mathbf{z}'^{(k)}$: **Coalition** simplified features

$$g\left(\mathbf{z}'^{(k)}\right) = \phi_0 + \sum_{j=1}^{p} \phi_j z_j'^{(k)}$$

ϕ_0: **Null Output** Average Model Baseline ($\mathbb{E}(\hat{f})$)

SHAP DEFINITION

Lundberg et al. 2017

Aim: Find an additive combination that explains the prediction of an observation $\mathbf{x}$ by computing the contribution of each feature to the prediction using a (more efficient) estimation procedure.

Definition

- Define simplified (binary) coalition feature space $\mathbf{Z}' \in \{0,1\}^{K \times p}$ with K rows and p columns
- Rows are referred to as $\mathbf{z}'^{(k)} = \{z_1'^{(k)}, \dots, z_p'^{(k)}\}$ with $k \in \{1, \dots, K\}$ (indexes k-th coalition)
- Columns are referred to as $\mathbf{z}_j$ with $j \in \{1, \dots, p\}$ being the index of the original feature

$\mathbf{z}'^{(k)}$: **Coalition** simplified features

$$g\left(\mathbf{z}'^{(k)}\right) = \phi_0 + \sum_{j=1}^{p} \phi_j z_j'^{(k)}$$

ϕ_0: **Null Output** Average Model Baseline ($\mathbb{E}(\hat{f})$)

ϕ_j: **Attribution** How much does feature j change the output for coalition k

SHAP DEFINITION

Lundberg et al. 2017

Aim: Find an additive combination that explains the prediction of an observation **x** by computing the contribution of each feature to the prediction using a (more efficient) estimation procedure.

$g(\mathbf{z}'^{(k)})$: **Marginal Contribution**
Contribution of coalition $\mathbf{z}'^{(k)}$ to the prediction

$$\underbrace{g\left(\mathbf{z}'^{(k)}\right) = \phi_0 + \sum_{j=1}^{p} \phi_j z_j'^{(k)}}_{\textbf{Additive Feature Attribution}}$$

ϕ_j: **Shapley Values**

Problem
How do we estimate the Shapley values ϕ_j?

KERNEL SHAP - IN 5 STEPS

Definition: A kernel-based, model-agnostic method to compute Shapley values via local surrogate models (e.g. linear model)

1. Sample coalitions
2. Transfer coalitions into feature space & get predictions by applying ML model
3. Compute weights through kernel
4. Fit a weighted linear model
5. Return Shapley values

KERNEL SHAP - IN 5 STEPS

Step 1: Sample coalitions

- Sample K coalitions from the simplified feature space

$$\mathbf{z}'^{(k)} \in \{0,1\}^p, \quad k \in \{1,\ldots,K\}$$

- For our simple example, we have in total $2^p = 2^3 = 8$ coalitions (without sampling)

Coalition	$\mathbf{z}'^{(k)}$	hum	temp	ws
$\emptyset$	$\mathbf{z}'^{(1)}$	0	0	0
hum	$\mathbf{z}'^{(2)}$	1	0	0
temp	$\mathbf{z}'^{(3)}$	0	1	0
ws	$\mathbf{z}'^{(4)}$	0	0	1
hum, temp	$\mathbf{z}'^{(5)}$	1	1	0
temp, ws	$\mathbf{z}'^{(6)}$	0	1	1
hum, ws	$\mathbf{z}'^{(7)}$	1	0	1
hum, temp, ws	$\mathbf{z}'^{(8)}$	1	1	1

KERNEL SHAP - IN 5 STEPS

Step 2: Transfer Coalitions into feature space & get predictions by applying ML model

- $\mathbf{z}'^{(k)}$ is 1 if features are part of the k-th coalition, 0 if they are absent
- To calculate predictions for these coalitions, we need to define a function which maps the binary feature space back to the original feature space

Coalition	$\mathbf{z}'^{(k)}$	hum	temp	ws
$\varnothing$	$\mathbf{z}'^{(1)}$	0	0	0
hum	$\mathbf{z}'^{(2)}$	1	0	0
temp	$\mathbf{z}'^{(3)}$	0	1	0
ws	$\mathbf{z}'^{(4)}$	0	0	1
hum, temp	$\mathbf{z}'^{(5)}$	1	1	0
temp, ws	$\mathbf{z}'^{(6)}$	0	1	1
hum, ws	$\mathbf{z}'^{(7)}$	1	0	1
hum, temp, ws	$\mathbf{z}'^{(8)}$	1	1	1

$\mathbf{x}^{coalition}$	hum	temp	ws
$\mathbf{x}^{\{\varnothing\}}$	$\varnothing$	$\varnothing$	$\varnothing$
$\mathbf{x}^{\{hum\}}$	51.6	$\varnothing$	$\varnothing$
$\mathbf{x}^{\{temp\}}$	$\varnothing$	5.1	$\varnothing$
$\mathbf{x}^{\{ws\}}$	$\varnothing$	$\varnothing$	17.0
$\mathbf{x}^{\{hum,temp\}}$	51.6	5.1	$\varnothing$
$\mathbf{x}^{\{temp,ws\}}$	$\varnothing$	5.1	17.0
$\mathbf{x}^{\{hum,ws\}}$	51.6	$\varnothing$	17.0
$\mathbf{x}^{\{hum,temp,ws\}}$	51.6	5.1	17.0

KERNEL SHAP - IN 5 STEPS

Step 2: Transfer Coalitions into feature space & get predictions by applying ML model

- Define $h_x\left(\mathbf{z}'^{(k)}\right) = \mathbf{z}^{(k)}$ where $h_x : \{0, 1\}^p \to \mathbb{R}^p$ maps 1's to feature values of observation $\mathbf{x}$ for features part of the k-th coalition and 0's to feature values of a randomly sampled observation for features absent in the k-th coalition (feature values are permuted multiple times)
- Predict with ML model on this dataset $\hat{f} : \hat{f}\left(h_x\left(\mathbf{z}'^{(k)}\right)\right)$

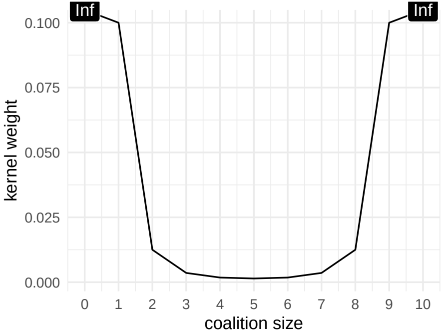
$h_x(\mathbf{z}'^{(k)})$

Coalition	$\mathbf{z}'^{(k)}$	hum	temp	ws	$\mathbf{z}^{(k)}$	hum	temp	ws	$\hat{f}\left(h_x\left(\mathbf{z}'^{(k)}\right)\right)$
$\emptyset$	$\mathbf{z}'^{(1)}$	0	0	0	$\mathbf{z}^{(1)}$	64.3	28.0	14.5	6211
hum	$\mathbf{z}'^{(2)}$	1	0	0	$\mathbf{z}^{(2)}$	51.6	28.0	14.5	5586
temp	$\mathbf{z}'^{(3)}$	0	1	0	$\mathbf{z}^{(3)}$	64.3	5.1	14.5	3295
ws	$\mathbf{z}'^{(4)}$	0	0	1	$\mathbf{z}^{(4)}$	64.3	28.0	17.0	5762
hum, temp	$\mathbf{z}'^{(5)}$	1	1	0	$\mathbf{z}^{(5)}$	51.6	5.1	14.5	2616
temp, ws	$\mathbf{z}'^{(6)}$	0	1	1	$\mathbf{z}^{(6)}$	64.3	5.1	17.0	2900
hum, ws	$\mathbf{z}'^{(7)}$	1	0	1	$\mathbf{z}^{(7)}$	51.6	28.0	17.0	5411
hum, temp, ws	$\mathbf{z}'^{(8)}$	1	1	1	$\mathbf{z}^{(8)}$	51.6	5.1	17.0	2573

KERNEL SHAP - IN 5 STEPS

Step 3: Compute weights through Kernel

Intuition: We learn most about individual features if we can study their effects in isolation or at maximal interaction: Small coalitions (few 1's) and large coalitions (i.e. many 1's) get the largest weights



KERNEL SHAP - IN 5 STEPS

Step 3: Compute weights through Kernel ▸ see shapley_kernel_proof.pdf

Intuition: We learn most about individual features if we can study their effects in isolation or at maximal interaction: Small coalitions (few 1's) and large coalitions (i.e. many 1's) get the largest weights

p: Number of features in **x**

$$\pi_x\left(\mathbf{z}'^{(k)}\right) = \frac{(p-1)}{\binom{p}{|\mathbf{z}'^{(k)}|}\,|\mathbf{z}'^{(k)}|\,(p-|\mathbf{z}'^{(k)}|)}$$

$\pi_x(\mathbf{z}'^{(k)})$: kernel weight for coalition $\mathbf{z}'^{(k)}$

$|\ \mathbf{z}'^{(k)}\ |$: coalition size / sum of 1s in $\mathbf{z}'^{(k)}$

KERNEL SHAP - IN 5 STEPS

Step 3: Compute weights through Kernel

Purpose: to include this knowledge in the local surrogate model (linear regression), we calculate weights for each coalition which are the observations of the linear regression

$$\pi_x\left(\mathbf{z}'\right) = \frac{(p-1)}{\binom{p}{|\mathbf{z}'|}|\mathbf{z}'|(p-|\mathbf{z}'|)} \rightsquigarrow \pi_x\left(\mathbf{z}' = (1,0,0)\right) = \frac{(3-1)}{\binom{3}{1}1\,(3-1)} = \frac{1}{3}$$

Coalition	$\mathbf{z}'^{(k)}$	hum	temp	ws	weight
$\varnothing$	$\mathbf{z}'^{(1)}$	0	0	0	∞
hum	$\mathbf{z}'^{(2)}$	1	0	0	0.33
temp	$\mathbf{z}'^{(3)}$	0	1	0	0.33
ws	$\mathbf{z}'^{(4)}$	0	0	1	0.33
hum, temp	$\mathbf{z}'^{(5)}$	1	1	0	0.33
temp, ws	$\mathbf{z}'^{(6)}$	0	1	1	0.33
hum, ws	$\mathbf{z}'^{(7)}$	1	0	1	0.33
hum, temp, ws	$\mathbf{z}'^{(8)}$	1	1	1	∞

KERNEL SHAP - IN 5 STEPS

Step 3: Compute weights through Kernel

Purpose: to include this knowledge in the local surrogate model (linear regression), we calculate weights for each coalition which are the observations of the linear regression

Coalition	$\mathbf{z}'^{(k)}$	hum	temp	ws	weight
$\emptyset$	$\mathbf{z}'^{(1)}$	0	0	0	∞
hum	$\mathbf{z}'^{(2)}$	1	0	0	0.33
temp	$\mathbf{z}'^{(3)}$	0	1	0	0.33
ws	$\mathbf{z}'^{(4)}$	0	0	1	0.33
hum, temp	$\mathbf{z}'^{(5)}$	1	1	0	0.33
temp, ws	$\mathbf{z}'^{(6)}$	0	1	1	0.33
hum, ws	$\mathbf{z}'^{(7)}$	1	0	1	0.33
hum, temp, ws	$\mathbf{z}'^{(8)}$	1	1	1	∞

$\rightsquigarrow$ weights for empty and full set are infinity and not used as observations for the linear regression
$\rightsquigarrow$ instead constraints are used such that properties (local accuracy and missingness) are satisfied

KERNEL SHAP - IN 5 STEPS

Step 4: Fit a weighted linear model

Aim: Estimate a weighted linear model with Shapley values being the coefficients ϕ_j

$$g\left(\mathbf{z}'^{(k)}\right) = \phi_0 + \sum_{j=1}^{p} \phi_j z_j'^{(k)}$$

and minimize by WLS using the weights π_x of step 3

$$L\left(\hat{f}, g, \pi_x\right) = \sum_{k=1}^{K} \left[\hat{f}\left(h_x\left(\mathbf{z}'^{(k)}\right)\right) - g\left(\mathbf{z}'^{(k)}\right)\right]^2 \pi_x\left(\mathbf{z}'^{(k)}\right)$$

with $\phi_0 = \mathbb{E}(\hat{f})$ and $\phi_p = \hat{f}(x) - \sum_{j=0}^{p-1} \phi_j$ we receive a $p-1$ dimensional linear regression problem

KERNEL SHAP - IN 5 STEPS

Step 4: Fit a weighted linear model

Aim: Estimate a weighted linear model with Shapley values being the coefficients ϕ_j

$$g\left(\mathbf{z}'^{(k)}\right) = \phi_0 + \sum_{j=1}^{p} \phi_j z_j'^{(k)} \rightsquigarrow g\left(\mathbf{z}'^{(k)}\right) = 4515 + 34 \cdot z_1'^{(k)} - 1654 \cdot z_2'^{(k)} - 323 \cdot z_3'^{(k)}$$

$\mathbf{z}'^{(k)}$	hum	temp	ws	weight	$\hat{f}$
$\mathbf{z}'^{(2)}$	1	0	0	0.33	4635
$\mathbf{z}'^{(3)}$	0	1	0	0.33	3087
$\mathbf{z}'^{(4)}$	0	0	1	0.33	4359
$\mathbf{z}'^{(5)}$	1	1	0	0.33	3060
$\mathbf{z}'^{(6)}$	0	1	1	0.33	2623
$\mathbf{z}'^{(7)}$	1	0	1	0.33	4450

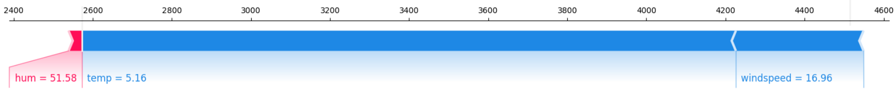
input *output*

KERNEL SHAP - IN 5 STEPS

Step 5: Return SHAP values

Intuition: Estimated Kernel SHAP values are equivalent to Shapley values

$$g(\mathbf{z}'^{(8)}) = \hat{f}(h_x(\mathbf{z}'^{(8)})) = 4515 + 34 \cdot 1 - 1654 \cdot 1 - 323 \cdot 1 = \underbrace{\mathbb{E}(\hat{f})}_{\phi_0} + \phi_{hum} + \phi_{temp} + \phi_{ws} = \hat{f}(\mathbf{x}) = 2573$$



PROPERTIES

Local Accuracy

$$f(\mathbf{x}) = g\left(\mathbf{x}'\right) = \phi_0 + \sum_{j=1}^{p} \phi_j x_j'$$

Intuition: If the coalition includes all features ($\mathbf{x}' \in \{1\}^p$), the attributions ϕ_j and the null output ϕ_0 sum up to the original model output $f(\mathbf{x})$

Local accuracy corresponds to the **axiom of efficiency** in Shapley game theory

PROPERTIES

Local Accuracy

$$f(\mathbf{x}) = g\left(\mathbf{x}'\right) = \phi_0 + \sum_{j=1}^{p} \phi_j x_j'$$

Missingness

$$x_j' = 0 \Longrightarrow \phi_j = 0$$

Intution: A missing feature gets an attribution of zero

PROPERTIES

Local Accuracy

$$f(\mathbf{x}) = g\left(\mathbf{x}'\right) = \phi_0 + \sum_{j=1}^{p} \phi_j x_j'$$

Missingness

$$x_j' = 0 \Longrightarrow \phi_j = 0$$

Consistency

$\hat{f}_x\left(\mathbf{z}'^{(k)}\right) = \hat{f}\left(h_x\left(\mathbf{z}'^{(k)}\right)\right)$ and $\mathbf{z}'^{(k)}_{-j}$ denote setting $z_j'^{(k)} = 0$. For any two models $\hat{f}$ and $\hat{f}'$, if

$$\hat{f}_x'\left(\mathbf{z}'^{(k)}\right) - \hat{f}_x'\left(\mathbf{z}'^{(k)}_{-j}\right) \geq \hat{f}_x\left(\mathbf{z}'^{(k)}\right) - \hat{f}_x\left(\mathbf{z}'^{(k)}_{-j}\right)$$

for all inputs $\mathbf{z}'^{(k)} \in \{0, 1\}^p$, then

$$\phi_j\left(\hat{f}', \mathbf{x}\right) \geq \phi_j(\hat{f}, \mathbf{x})$$

PROPERTIES

Local Accuracy

$$f(\mathbf{x}) = g\left(\mathbf{x}'\right) = \phi_0 + \sum_{j=1}^{p} \phi_j x_j'$$

Missingness

$$x_j' = 0 \Longrightarrow \phi_j = 0$$

Consistency

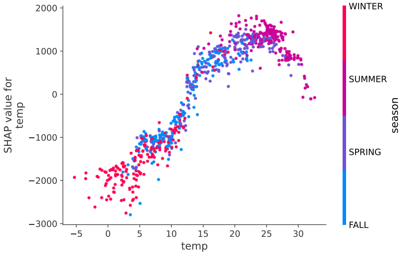
$$\hat{f}_x'\left(\mathbf{z}'^{(k)}\right) - \hat{f}_x'\left(\mathbf{z}_{-j}'^{(k)}\right) \geq \hat{f}_x\left(\mathbf{z}'^{(k)}\right) - \hat{f}_x\left(\mathbf{z}_{-j}'^{(k)}\right) \Longrightarrow \phi_j\left(\hat{f}', \mathbf{x}\right) \geq \phi_j(\hat{f}, \mathbf{x})$$

Intution: If a model changes so that the marginal contribution of a feature value increases or stays the same, the Shapley value also increases or stays the same

From **consistency** the Shapley **axioms of additivity, dummy and symmetry** follow

Interpretable Machine Learning

Global SHAP



Learning goals

- Get an intuition of additive feature attributions
- Understand the concept of Kernel SHAP
- Ability to interpret SHAP plots
- Global SHAP methods

GLOBAL SHAP ▸ Lundberg et al. 2018

Idea:

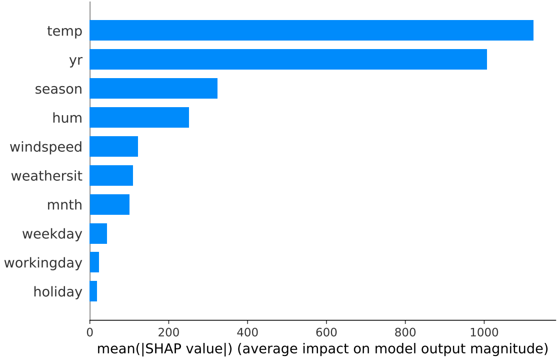
- Run SHAP for every observation and thereby get a matrix of Shapley values
- The matrix has one row per data observation and one column per feature
- We can interpret the model globally by analyzing the Shapley values in this matrix

$$\Phi = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \dots & \phi_{1p} \\ \phi_{21} & \phi_{22} & \phi_{23} & \dots & \phi_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_{n1} & \phi_{n2} & \phi_{n3} & \dots & \phi_{np} \end{bmatrix}$$

FEATURE IMPORTANCE

Idea: Average the absolute Shapley values of each feature over all observations. This corresponds to calculating averages column by column in Φ

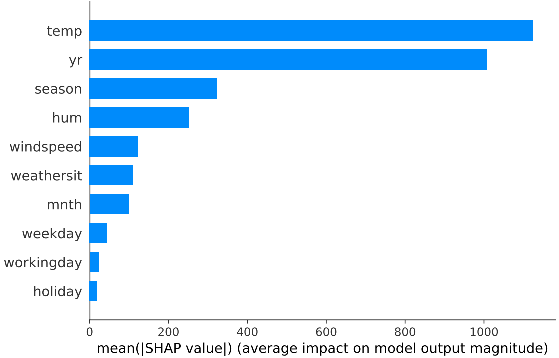
$$I_j = \frac{1}{n} \sum_{i=1}^{n} \left| \phi_j^{(i)} \right|$$



FEATURE IMPORTANCE

Interpretation:

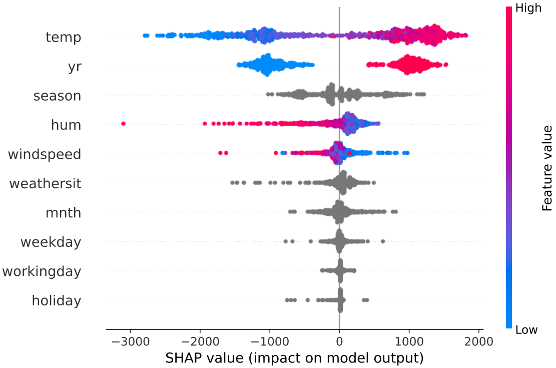
- The features temperature and year have by far the highest influence on the model's prediction
- Compared to Shapley values, no effect direction is provided, but instead a feature ranking similar to PFI
- However, Shapley FI is based on the model's predictions only while PFI is based on the model's performance (loss)



SUMMARY PLOT

Combines feature importance with feature effects

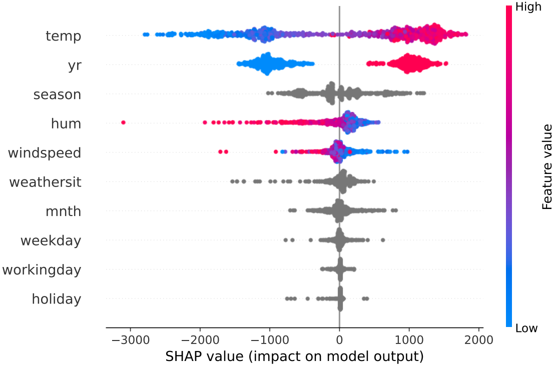
- Each point is a Shapley value for a feature and an observation
- The color represents the value of the feature from low to high
- Overlapping points are jittered in y-axis direction



SUMMARY PLOT

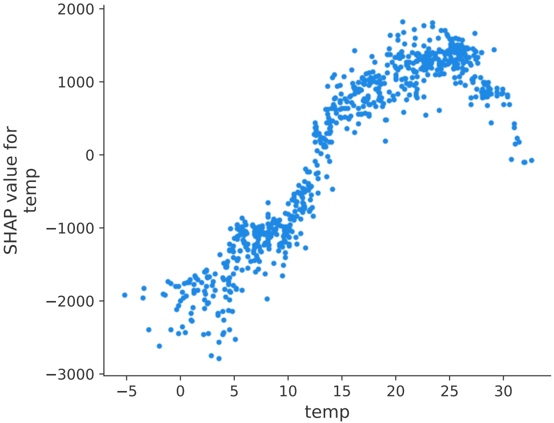
Interpretation:

- Low temperatures have a negative impact while high temperatures lead to more bike rentals
- Year: two point clouds for 2011 and 2012 (other categorical features are gray)
- A high humidity has a huge, negative impact on the bike rental, while low humidity has a rather minor positive impact on bike rentals



DEPENDENCE PLOT

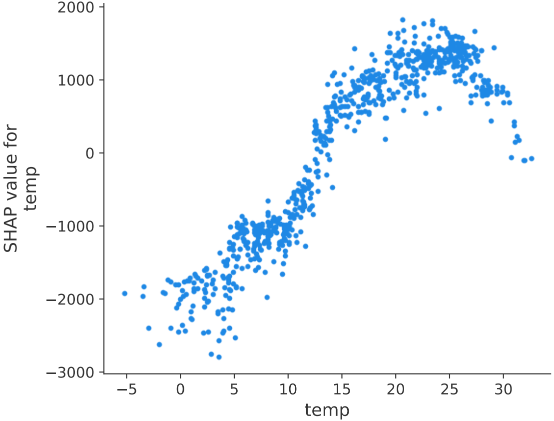
- Visualize the marginal contribution of a feature similar to the PDP
- Plot a point with the feature value on the x-axis and the corresponding Shapley value on the y-axis



DEPENDENCE PLOT

Interpretation:

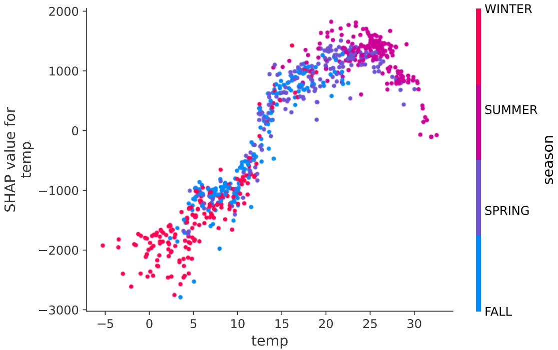
- Increasing temperatures induce increasing bike rentals until 25°C
- If it gets too hot, the bike rentals decrease



DEPENDENCE PLOT

Interpretation:

- We can colour the observations by a second feature to detect interactions
- Visibly the temperatures interaction with the season is very strong



DISCUSSION

Advantages

- All the advantages of Shapley values
- Unify the field of interpretable machine learning in the class of additive feature attribution methods
- Has a fast implementation for tree-based models
- Various global interpretation methods

Disadvantages

- Disadvantages of Shapley values also apply to SHAP
- KernelSHAP is slow (TreeSHAP can be used as a faster alternative for tree-based models ▸ Lundberg et al 2018 – and for an intuitive explanation ▸ see Sukumar: TreeSHAP)
- KernelSHAP ignores feature dependence