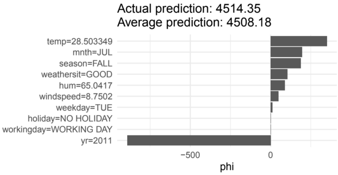
Interpretable Machine Learning

Shapley

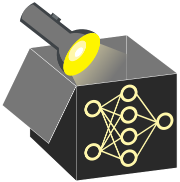
Shapley Values for Local Explanations



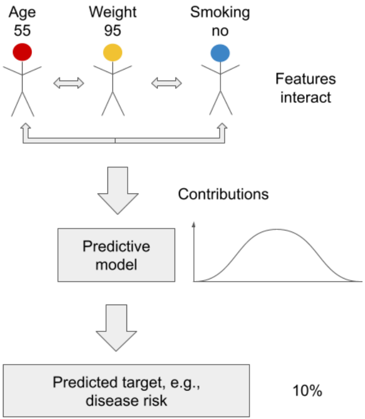
Learning goals

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

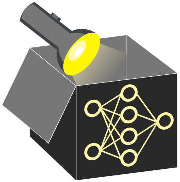
FROM GAME THEORY TO MACHINE LEARNING



- Model prediction depends on feature interactions for a specific observation
- **Goal:** Decompose prediction into **individual feature contributions**
- **Idea:** Treat features as players jointly producing a prediction
- How to fairly assign credit to features?
  $\Rightarrow$ Shapley values



FROM GAME THEORY TO MACHINE LEARNING

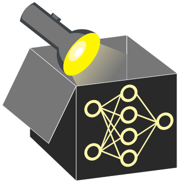


- **Game:** Predict $\hat{f}(x_1, x_2, \ldots, x_p)$ for a single observation $\mathbf{x}$
- **Players:** Features $x_j, j \in \{1, \ldots, p\}$, cooperate to produce a prediction
- **Value function:** Defines payout of coalition $S \subseteq P$ for observation $\mathbf{x}$ by

$$v(S) = \hat{f}_S(\mathbf{x}_S) - \hat{f}_\emptyset, \text{ where}$$

  - $\hat{f}_S : \mathcal{X}_S \mapsto \mathcal{Y}$ is the PD function $\hat{f}_S(\mathbf{x}_S) := \int \hat{f}(\mathbf{x}_S, X_{-S})\, d\mathbb{P}_{X_{-S}}$
    $\rightsquigarrow$ "Removes" features in $-S$ by marginalizing, keeping $\hat{f}$ fixed
  - Mean prediction $\hat{f}_\emptyset := \mathbb{E}_\mathbf{x}(\hat{f}(\mathbf{x}))$ is subtracted to ensure $v(\emptyset) = 0$
- **Goal:** Distribute total payout $v(P) = \hat{f}(\mathbf{x}) - \hat{f}_\emptyset$ fairly among features

FROM GAME THEORY TO MACHINE LEARNING



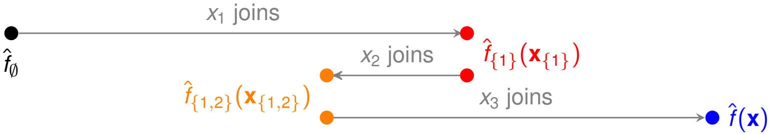
- **Game:** Predict $\hat{f}(x_1, x_2, \ldots, x_p)$ for a single observation $\mathbf{x}$
- **Players:** Features $x_j, j \in \{1, \ldots, p\}$, cooperate to produce a prediction
- **Value function:** Defines payout of coalition $S \subseteq P$ for observation $\mathbf{x}$ by

$$v(S) = \hat{f}_S(\mathbf{x}_S) - \hat{f}_\emptyset, \text{ where}$$

  - $\hat{f}_S : \mathcal{X}_S \mapsto \mathcal{Y}$ is the PD function $\hat{f}_S(\mathbf{x}_S) := \int \hat{f}(\mathbf{x}_S, X_{-S})\, d\mathbb{P}_{X_{-S}}$
    $\rightsquigarrow$ "Removes" features in $-S$ by marginalizing, keeping $\hat{f}$ fixed
  - Mean prediction $\hat{f}_\emptyset := \mathbb{E}_\mathbf{x}(\hat{f}(\mathbf{x}))$ is subtracted to ensure $v(\emptyset) = 0$
- **Goal:** Distribute total payout $v(P) = \hat{f}(\mathbf{x}) - \hat{f}_\emptyset$ fairly among features
- **Marginal contribution of feature j joining coalition S** ($\hat{f}_\emptyset$ cancels):

$$\Delta(j, S) = v(S \cup \{j\}) - v(S) = \hat{f}_{S\cup\{j\}}(\mathbf{x}_{S\cup\{j\}}) - \hat{f}_S(\mathbf{x}_S)$$

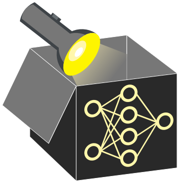
- **Example (3 features):** Feature contributions for joining order $x_1 \to x_2 \to x_3$ toward total payout $v(P) = \hat{f}(\mathbf{x}) - \hat{f}_\emptyset$, each step reflects a marginal contribution



SHAPLEY VALUE - DEFINITION

▸ "Shapley" 1953

▸ "Strumbelj et al." 2014



Order definition: Shapley value $\phi_j(\mathbf{x})$ quantifies contribution of x_j via

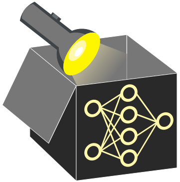
$$\phi_j(\mathbf{x}) = \frac{1}{|P|!} \sum_{\tau \in \Pi} \underbrace{\hat{f}_{S_j^\tau \cup \{j\}}(\mathbf{x}_{S_j^\tau \cup \{j\}}) - \hat{f}_{S_j^\tau}(\mathbf{x}_{S_j^\tau})}_{\Delta(j, S_j^\tau) \text{ marginal contribution of feature } j}$$

- **Interpretation:** $\phi_j(\mathbf{x})$ quantifies how much feature x_j contributes to the difference between $\hat{f}(\mathbf{x})$ and the mean prediction $\hat{f}_\emptyset$
  $\leadsto$ Marginal contributions and Shapley values can be negative
- **Exact computation of ϕ_j:** Using PD function $\hat{f}_S(\mathbf{x}_S) = \frac{1}{n}\sum_{i=1}^n \hat{f}(\mathbf{x}_S, \mathbf{x}_{-S}^{(i)})$ yields

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

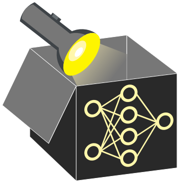
$\leadsto$ $\hat{f}_S$ marginalizes over all features not in S using all obs. $i = 1, \ldots, n$
$\leadsto$ Exact computation requires $|P|! \cdot n$ marginal contribution terms

ESTIMATION: A PRACTICAL PROBLEM



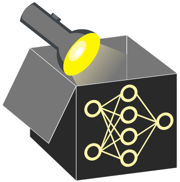
- **Exact computation is infeasible for many features:**
  For $|P| = 10$, the number of permutations is $10! \approx 3.6$ million
  $\rightsquigarrow$ Complexity grows factorially with feature count

ESTIMATION: A PRACTICAL PROBLEM



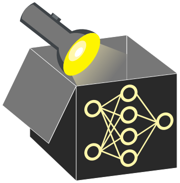
- **Exact computation is infeasible for many features:**
  For $|P| = 10$, the number of permutations is $10! \approx 3.6$ million
  $\rightsquigarrow$ Complexity grows factorially with feature count
- **Additional challenge: Estimating marginal predictions (PD funcs)**
  Each permut. τ defines a coal. S_j^τ needing its own estimate of $\hat{f}_{S_j^\tau}(\mathbf{x}_{S_j^\tau})$
  $\rightsquigarrow$ With $|P|!$ permutations and n data points, the number of such estimates grows rapidly, making marginalization costly

ESTIMATION: A PRACTICAL PROBLEM



- **Exact computation is infeasible for many features:**
  For $|P| = 10$, the number of permutations is $10! \approx 3.6$ million
  $\rightsquigarrow$ Complexity grows factorially with feature count
- **Additional challenge: Estimating marginal predictions (PD funcs)**
  Each permut. τ defines a coal. S_j^τ needing its own estimate of $\hat{f}_{S_j^\tau}(\mathbf{x}_{S_j^\tau})$
  $\rightsquigarrow$ With $|P|!$ permutations and n data points, the number of such estimates grows rapidly, making marginalization costly
- **Solution: Sampling-based approximation**
  Instead of computing $|P|! \cdot n$ terms, we approximate using M random samples of permutations τ and data points

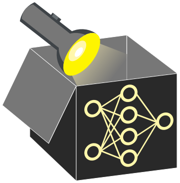
ESTIMATION: A PRACTICAL PROBLEM



- **Exact computation is infeasible for many features:**
  For $|P| = 10$, the number of permutations is $10! \approx 3.6$ million
  $\rightsquigarrow$ Complexity grows factorially with feature count
- **Additional challenge: Estimating marginal predictions (PD funcs)**
  Each permut. τ defines a coal. S_j^τ needing its own estimate of $\hat{f}_{S_j^\tau}(\mathbf{x}_{S_j^\tau})$
  $\rightsquigarrow$ With $|P|!$ permutations and n data points, the number of such estimates grows rapidly, making marginalization costly
- **Solution: Sampling-based approximation**
  Instead of computing $|P|! \cdot n$ terms, we approximate using M random samples of permutations τ and data points
- **Tradeoff: Accuracy vs. Efficiency**
  Larger M improves Shapley approximation
  $\rightsquigarrow$ Higher cost, but better fidelity to the exact value

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014

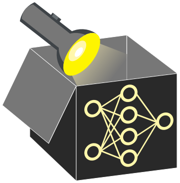


Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



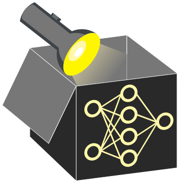
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \ldots, M$ **do**:

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



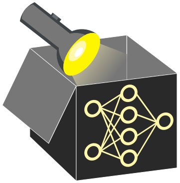
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \dots, M$ **do**:
   1. Sample random permut. $\tau = (\tau^{(1)}, \dots, \tau^{(p)}) \in \Pi$ of feature indices

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



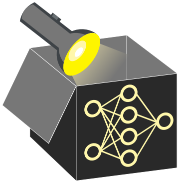
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \ldots, M$ **do**:
   1. Sample random permut. $\tau = (\tau^{(1)}, \ldots, \tau^{(p)}) \in \Pi$ of feature indices
   2. Let coalition $S_m := S_j^{\tau}$ be the set of features preceding j in τ

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



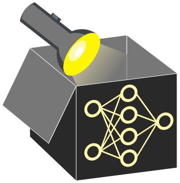
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \ldots, M$ **do**:
   1. Sample random permut. $\tau = (\tau^{(1)}, \ldots, \tau^{(p)}) \in \Pi$ of feature indices
   2. Let coalition $S_m := S_j^\tau$ be the set of features preceding j in τ
   3. Sample random data point $\mathbf{z}^{(m)} \in \mathcal{D}$ (so-called background data)

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



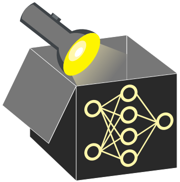
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \dots, M$ **do**:
   1. Sample random permut. $\tau = (\tau^{(1)}, \dots, \tau^{(p)}) \in \Pi$ of feature indices
   2. Let coalition $S_m := S_j^\tau$ be the set of features preceding j in τ
   3. Sample random data point $\mathbf{z}^{(m)} \in \mathcal{D}$ (so-called background data)
   4. Construct two hybrid obs. by combining values from $\mathbf{x}$ and $\mathbf{z}^{(m)}$:

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014

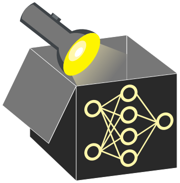


Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \dots, M$ **do**:
    1. Sample random permut. $\tau = (\tau^{(1)}, \dots, \tau^{(p)}) \in \Pi$ of feature indices
    2. Let coalition $S_m := S_j^\tau$ be the set of features preceding j in τ
    3. Sample random data point $\mathbf{z}^{(m)} \in \mathcal{D}$ (so-called background data)
    4. Construct two hybrid obs. by combining values from $\mathbf{x}$ and $\mathbf{z}^{(m)}$:
        - $\mathbf{x}_{+j}^{(m)} = (x_{\tau^{(1)}}, \dots, x_{\tau^{(|S_m|)}}, x_j, z_{\tau^{(|S_m|+2)}}^{(m)}, \dots, z_{\tau^{(p)}}^{(m)})$
          $\rightsquigarrow$ includes $\mathbf{x}_{S_m \cup \{j\}}$ (features in $S_m \cup \{j\}$ from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$

APPROXIMATION ALGORITHM ▸ "Strumbelj et al." 2014



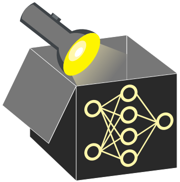
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \ldots, M$ **do**:
    1. Sample random permut. $\tau = (\tau^{(1)}, \ldots, \tau^{(p)}) \in \Pi$ of feature indices
    2. Let coalition $S_m := S_j^\tau$ be the set of features preceding j in τ
    3. Sample random data point $\mathbf{z}^{(m)} \in \mathcal{D}$ (so-called background data)
    4. Construct two hybrid obs. by combining values from $\mathbf{x}$ and $\mathbf{z}^{(m)}$:
        - $\mathbf{x}_{+j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, x_j, z_{\tau^{(|S_m|+2)}}^{(m)}, \ldots, z_{\tau^{(p)}}^{(m)})$
          $\rightsquigarrow$ includes $\mathbf{x}_{S_m \cup \{j\}}$ (features in $S_m \cup \{j\}$ from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$
        - $\mathbf{x}_{-j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, z_j^{(m)}, z_{\tau^{(|S_m|+2)}}^{(m)}, \ldots, z_{\tau^{(p)}}^{(m)})$
          $\rightsquigarrow$ includes $\mathbf{x}_{S_m}$ (features in S_m excl. x_j from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



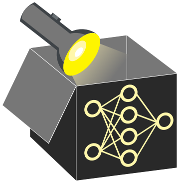
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \ldots, M$ **do**:
   1. Sample random permut. $\tau = (\tau^{(1)}, \ldots, \tau^{(p)}) \in \Pi$ of feature indices
   2. Let coalition $S_m := S_j^\tau$ be the set of features preceding j in τ
   3. Sample random data point $\mathbf{z}^{(m)} \in \mathcal{D}$ (so-called background data)
   4. Construct two hybrid obs. by combining values from $\mathbf{x}$ and $\mathbf{z}^{(m)}$:
      - $\mathbf{x}_{+j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, x_j, z_{\tau^{(|S_m|+2)}}^{(m)}, \ldots, z_{\tau^{(p)}}^{(m)})$
        $\leadsto$ includes $\mathbf{x}_{S_m \cup \{j\}}$ (features in $S_m \cup \{j\}$ from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$
      - $\mathbf{x}_{-j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, z_j^{(m)}, z_{\tau^{(|S_m|+2)}}^{(m)}, \ldots, z_{\tau^{(p)}}^{(m)})$
        $\leadsto$ includes $\mathbf{x}_{S_m}$ (features in S_m excl. x_j from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$
   5. Compute marginal contribution $\Delta(j, S_m) = \hat{f}(\mathbf{x}_{+j}^{(m)}) - \hat{f}(\mathbf{x}_{-j}^{(m)})$

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



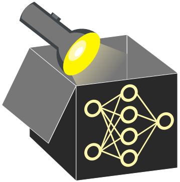
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \ldots, M$ **do**:
   1. Sample random permut. $\tau = (\tau^{(1)}, \ldots, \tau^{(p)}) \in \Pi$ of feature indices
   2. Let coalition $S_m := S_j^\tau$ be the set of features preceding j in τ
   3. Sample random data point $\mathbf{z}^{(m)} \in \mathcal{D}$ (so-called background data)
   4. Construct two hybrid obs. by combining values from $\mathbf{x}$ and $\mathbf{z}^{(m)}$:
      - $\mathbf{x}_{+j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, x_j, z_{\tau^{(|S_m|+2)}}^{(m)}, \ldots, z_{\tau^{(p)}}^{(m)})$
        $\rightsquigarrow$ includes $\mathbf{x}_{S_m \cup \{j\}}$ (features in $S_m \cup \{j\}$ from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$
      - $\mathbf{x}_{-j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, z_j^{(m)}, z_{\tau^{(|S_m|+2)}}^{(m)}, \ldots, z_{\tau^{(p)}}^{(m)})$
        $\rightsquigarrow$ includes $\mathbf{x}_{S_m}$ (features in S_m excl. x_j from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$
   5. Compute marginal contribution $\Delta(j, S_m) = \hat{f}(\mathbf{x}_{+j}^{(m)}) - \hat{f}(\mathbf{x}_{-j}^{(m)})$
2. Compute Shapley value $\phi_j = \frac{1}{M} \sum_{m=1}^{M} \Delta(j, S_m)$

APPROXIMATION ALGORITHM

▸ "Strumbelj et al." 2014



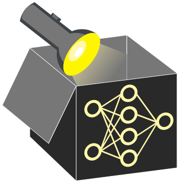
Estimate Shapley value ϕ_j of observation $\mathbf{x}$ for feature j:

- **Input:** $\mathbf{x}$ obs. of interest, j feat. of interest, $\hat{f}$ model, $\mathcal{D}$ data, M iterations

1. For $m = 1, \ldots, M$ **do**:
   1. Sample random permut. $\tau = (\tau^{(1)}, \ldots, \tau^{(p)}) \in \Pi$ of feature indices
   2. Let coalition $S_m := S_j^\tau$ be the set of features preceding j in τ
   3. Sample random data point $\mathbf{z}^{(m)} \in \mathcal{D}$ (so-called background data)
   4. Construct two hybrid obs. by combining values from $\mathbf{x}$ and $\mathbf{z}^{(m)}$:
      - $\mathbf{x}_{+j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, x_j, z^{(m)}_{\tau^{(|S_m|+2)}}, \ldots, z^{(m)}_{\tau^{(p)}})$
        $\rightsquigarrow$ includes $\mathbf{x}_{S_m \cup \{j\}}$ (features in $S_m \cup \{j\}$ from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$
      - $\mathbf{x}_{-j}^{(m)} = (x_{\tau^{(1)}}, \ldots, x_{\tau^{(|S_m|)}}, z_j^{(m)}, z^{(m)}_{\tau^{(|S_m|+2)}}, \ldots, z^{(m)}_{\tau^{(p)}})$
        $\rightsquigarrow$ includes $\mathbf{x}_{S_m}$ (features in S_m excl. x_j from $\mathbf{x}$), rest from $\mathbf{z}^{(m)}$
   5. Compute marginal contribution $\Delta(j, S_m) = \hat{f}(\mathbf{x}_{+j}^{(m)}) - \hat{f}(\mathbf{x}_{-j}^{(m)})$
2. Compute Shapley value $\phi_j = \frac{1}{M} \sum_{m=1}^{M} \Delta(j, S_m)$

$\rightsquigarrow$ Over M iterations, the PD functions $\hat{f}_{S_m}(\mathbf{x}_{S_m})$ and $\hat{f}_{S_m \cup \{j\}}(\mathbf{x}_{S_m \cup \{j\}})$ are approximated by $\hat{f}(\mathbf{x}_{-j}^{(m)})$ and $\hat{f}(\mathbf{x}_{+j}^{(m)})$, where features not in the coalition (to be marginalized) are imputed with vals from random data points $\mathbf{z}^{(m)}$

SHAPLEY VALUE APPROX. - ILLUSTRATION



Definition

x: obs. of interest

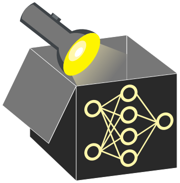
x with feature values in $\mathbf{x}_{S_m}$ (other are replaced)

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

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

	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	$random : z_{windspeed}^{(m)}$	$\underbrace{random : z_{year}^{(m)}}_{j}$

SHAPLEY VALUE APPROX. - ILLUSTRATION



Definition

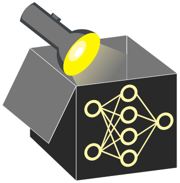
$$\phi_j(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^{M} \underbrace{\left[\hat{f}(\mathbf{x}_{+j}{}^{(m)}) - \hat{f}(\mathbf{x}_{-j}{}^{(m)}) \right]}_{:= \Delta(j, S_m)}$$

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 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	700
x_{-j}	10.66	56	$random : z_{windspeed}^{(m)}$	$random : z_{year}^{(m)}$	4900	
				j	$\hat{f}$	$\Delta(j, S_m)$ marginal contribution

SHAPLEY VALUE APPROX. - ILLUSTRATION



Definition

average the contributions of feature j

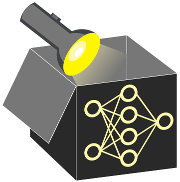
$$\phi_j(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\left[\hat{f}(\mathbf{x}_{+j}{}^{(m)}) - \hat{f}(\mathbf{x}_{-j}{}^{(m)})\right]$$

- Compute marginal contribution of feature j towards the prediction across all randomly drawn feature coalitions $S_1, \ldots, S_m$
- 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 obs. $\mathbf{x}$

$$m = \quad 1 \qquad 2 \qquad \ldots \qquad M$$

$$\underbrace{700 \quad 503 \quad \ldots \quad 300}_{\Delta(j, S_m)} \qquad \underbrace{\overset{\text{Shapley value}}{501}}_{\phi_j}$$

REVISITED: AXIOMS FOR FAIR ATTRIBUTIONS



We adapt the classic Shapley axioms to the setting of model predictions:

- **Efficiency**: Sum of Shapley values adds up to the centered prediction:

$$\sum_{j=1}^{p} \phi_j(\mathbf{x}) = \hat{f}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}}[\hat{f}(\mathbf{x})]$$

  $\rightsquigarrow$ All predictive contribution is fully distributed among features

- **Symmetry**: Identical contributors receive equal value:

$$\hat{f}_{S\cup\{j\}}(\mathbf{x}_{S\cup\{j\}}) = \hat{f}_{S\cup\{k\}}(\mathbf{x}_{S\cup\{k\}}) \; \forall S \subseteq P \setminus \{j,k\} \Rightarrow \phi_j = \phi_k$$

  $\rightsquigarrow$ Interaction effects are shared equitably

- **Dummy (Null Player)**: Irrelevant features receive zero attribution:

$$\hat{f}_{S\cup\{j\}}(\mathbf{x}_{S\cup\{j\}}) = \hat{f}_S(\mathbf{x}_S) \; \forall S \subseteq P \Rightarrow \phi_j = 0$$

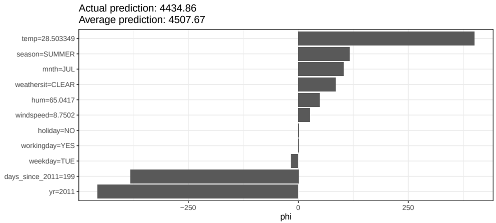
  $\rightsquigarrow$ Shapley value is zero for unused features (e.g., trees or LASSO)

- **Additivity**: Attributions are additive across models:

$$\phi_j(v_1 + v_2) = \phi_j(v_1) + \phi_j(v_2)$$

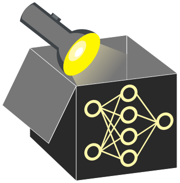
  $\rightsquigarrow$ Enables combining Shapley values for model ensembles

BIKE SHARING DATASET



- Shapley decomposition for a single prediction in bike sharing dataset
- Model pred.: $\hat{f}(\mathbf{x}^{(200)}) =$ **4434.86** vs. dataset avg.: $\mathbb{E}_{\mathbf{x}}[\hat{f}(\mathbf{x})] =$ **4507.67**
- Total feature attribution: $\sum_j \phi_j =$ **-72.81**
  $\rightsquigarrow$ Explain downward shift from mean prediction
- Temperature (with value 28.5°C) – strongest positive contributor: $+400$
- `yr = 2011` and `days_since_2011 = 199` strongly reduce prediction
  $\rightsquigarrow$ Model captures lower bike demand in 2011 compared to 2012

ADVANTAGES AND DISADVANTAGES



Advantages:

- **Strong theoretical foundation** from cooperative game theory
- **Fair attribution:** Prediction is additively distributed across features
  $\rightsquigarrow$ Easy to interpret for users
- **Contrastive explanations:** Quantify each feature's role in deviating from the average prediction

Disadvantages:

- **Comput. cost:** Exact computation scales factorially with feature count
  $\rightsquigarrow$ Without sampling, all 2^p coalitions (or $p!$ permuts) must be evaluated
- **Issue with correlated features:** Shapley values may evaluate the model on feature combinations that do not occur in the real data