Explainable AI - Lecture 4

The Shapley value

norwegian open ai lab
NTNU

The model agnostic regime

We have looked at several model agnostic methods, and they follow the same pattern:

Given a non-interpretable model, send inputs to the model and use the outputs to produce explanations.

The model agnostic regime

We have looked at several model agnostic methods, and they follow the same pattern:

Given a non-interpretable model, send inputs to the model and use the outputs to produce explanations.

Counterfactual methods do this through search: find the best input that changes the output as desired.

LIME builds a local surrogate model for a data point by perturbing the data point.

The model agnostic regime

We have looked at several model agnostic methods, and they follow the same pattern:

Given a non-interpretable model, send inputs to the model and use the outputs to produce explanations.

We might realise that our underlying question is actually "what is the best way to probe the model to learn more about how it works?"

The model agnostic regime

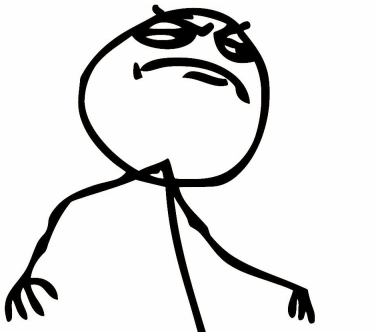
What is the best way to probe the model to learn more about how it works?

Input →  → Output

The model agnostic regime

What is the best way to probe the model to learn more about how it works?

Probably the Shapley decomposition.



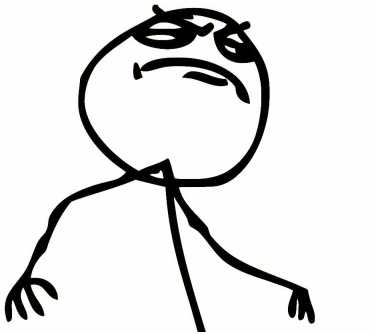
$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

What do you see here?

The model agnostic regime

What is the best way to probe the model to learn more about how it works?

Probably the Shapley decomposition.

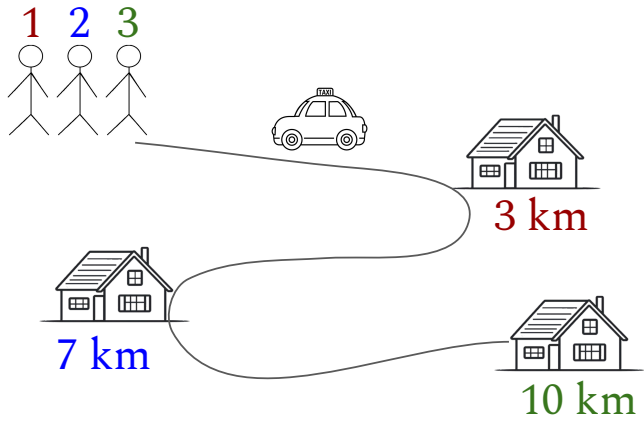


$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

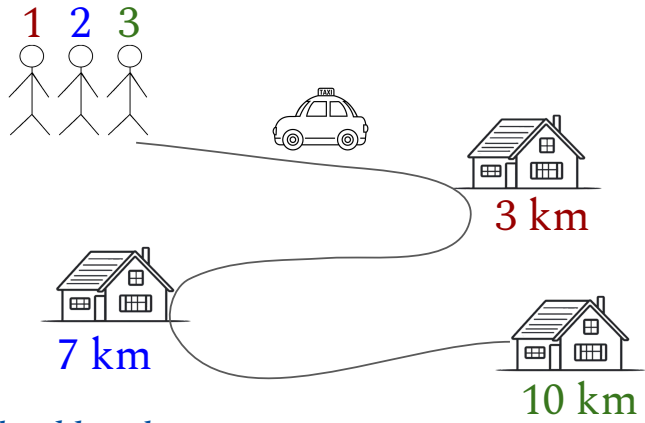
There is some set notation.
We have a sum over a subset of something.
The *v* thing is some kind of function working for sets.

The Shapley decomposition

Let's get some intuition for this equation by using a simple example



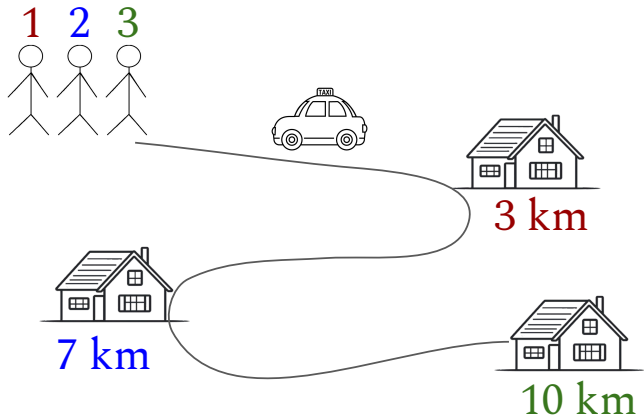
The Shapley decomposition



Three passengers (1,2,3) are sharing a cab to go (3,7,10) kilometers

What is the fair price each passenger should pay, i.e. for how many kilometers should each passenger pay?

The Shapley decomposition



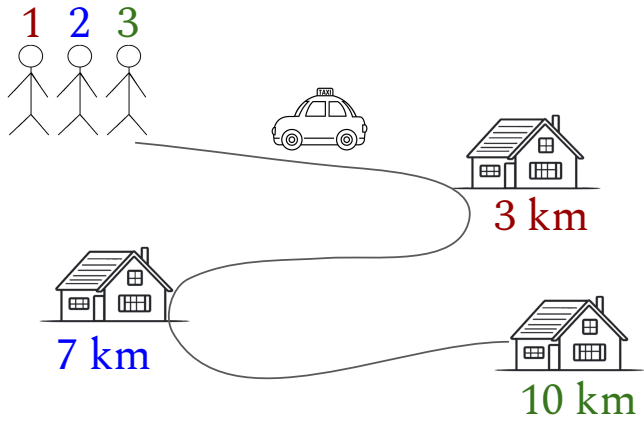
Three passengers (1,2,3) are sharing a cab to go (3,7,10) kilometers

What is the fair price each passenger should pay, i.e. for how many kilometers should each passenger pay?

We can use the Shapley decomposition to calculate this

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

The Shapley decomposition

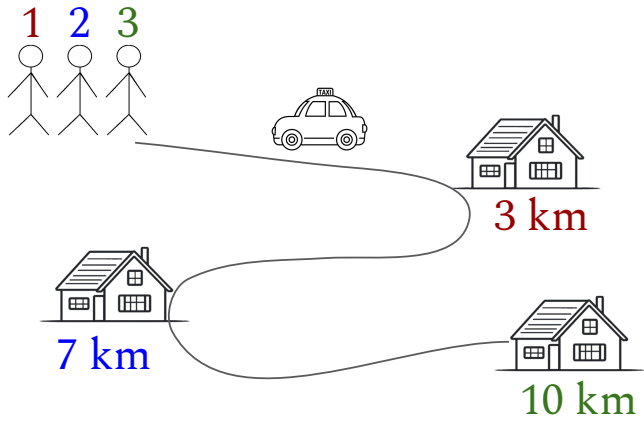


The Shapley decomposition is a solution concept from game theory.

Regard this problem as a collaborative game, where the passengers are players, and the total amount of kilometers travelled is the outcome of the game.

First, we need to characterize the game.

The Shapley decomposition



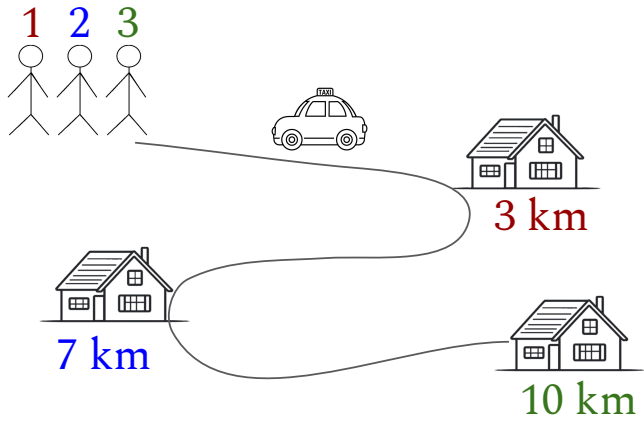
First, we need to characterize the game.

$$
\begin{aligned}
&v(\{\}) = 0 \quad \text{(no passengers costs nothing)} \\
&v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10 \\
&v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10 \\
&v(\{1,2,3\}) = 10
\end{aligned}
$$

The game is characterized using v. This is called the characteristic function, and its values are the characteristic function values.

The characteristic function is a **set function**, mapping a set to a single real number.

The Shapley decomposition



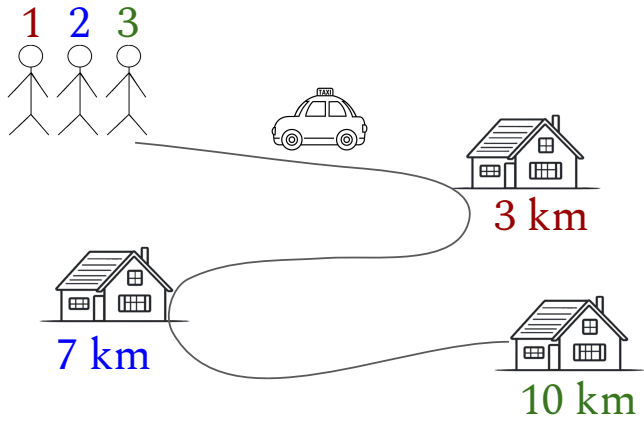
Characteristic function values:

$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

We'll use these to calculate the Shapley decomposition, where

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

The Shapley decomposition



Characteristic function values:

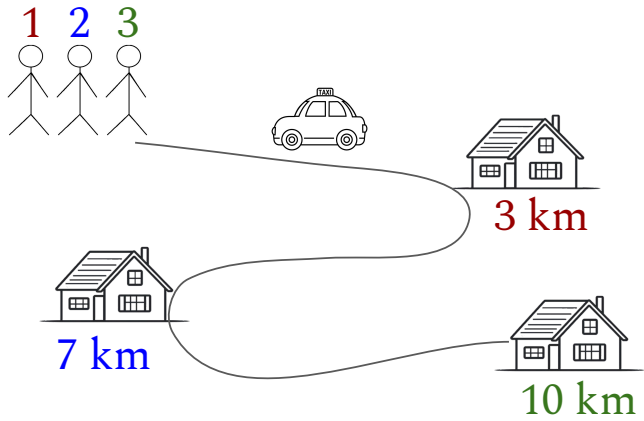
$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

We'll use these to calculate the Shapley decomposition, where

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

N = total number of passengers

The Shapley decomposition



Characteristic function values:

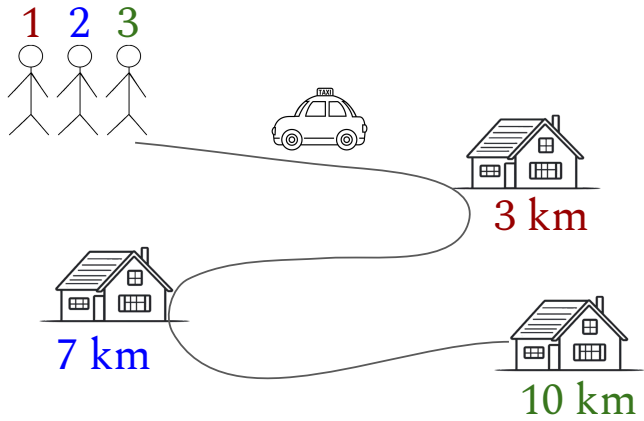
$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

We'll use these to calculate the Shapley decomposition, where

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

N = total number of passengers
S = subsets of N

The Shapley decomposition



Characteristic function values:

$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

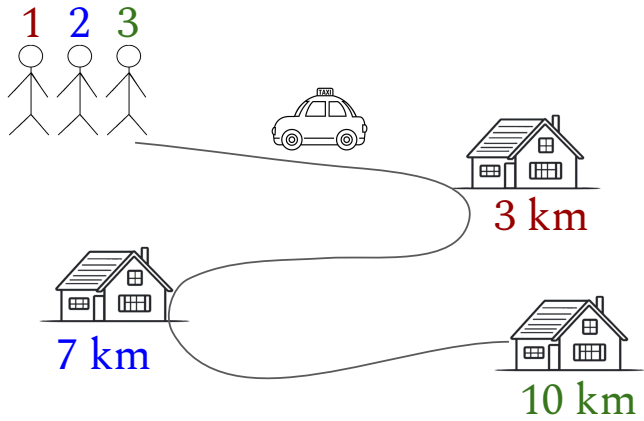
We'll use these to calculate the Shapley decomposition, where

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

N = total number of passengers
S = subsets of N

subsets S of N excluding passenger i

The Shapley decomposition



Characteristic function values:

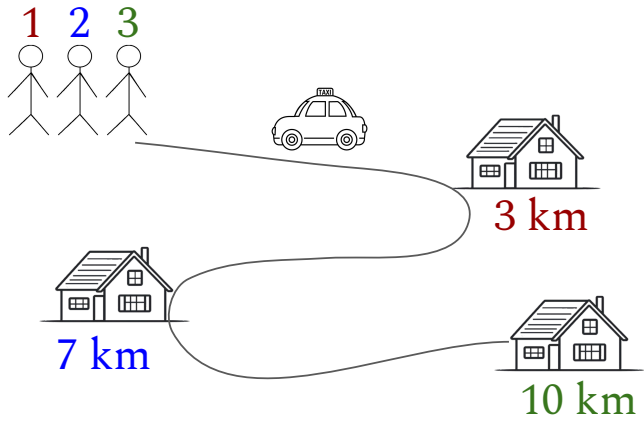
$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

Let's do this for passenger 1:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

What are the sets S excluding passenger 1?

The Shapley decomposition



Characteristic function values:

$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

Let's do this for passenger 1:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

Sets excluding passenger 1: { }, {2}, {3}, {2,3}

The Shapley decomposition

$v(\{\}) = 0 \quad (\text{no passengers costs nothing})$
$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$
$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$
$v(\{1,2,3\}) = 10$

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

Sets excluding passenger 1:

$$\phi_1 = \frac{1}{3}\left(v(\{1,2,3\}) - v(\{2,3\})\right)$$
$$+ \frac{1}{6}\left(v(\{1,2\}) - v(\{2\})\right)$$
$$+ \frac{1}{6}\left(v(\{1,3\}) - v(\{3\})\right)$$
$$+ \frac{1}{3}\left(v(\{1\}) - v(\{\})\right) = 1$$

The empty set: {}

The set containing passenger 2: {2}

The set containing passenger 3: {3}

The set containing passengers 2 and 3: {2,3}

The Shapley decomposition

$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$
$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$
$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$
$v(\{1,2,3\}) = 10$

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} (v(S \cup \{i\}) - v(S))$$

Sets excluding passenger 1: { }, {2}, {3}, {2,3}

$$\phi_1 = \frac{1(3-0-1)!}{3!}(v(\{1\}) - v(\{\})) + \frac{1(3-1-1)!}{3!}(v(\{1,2\}) - v(\{2\})) + \frac{1(3-1-1)!}{3!}(v(\{1,3\}) - v(\{3\})) + \frac{1(3-2-1)!}{3!}(v(\{1,2,3\} - v(\{2,3\})) = 1$$

ϕ_1 is the Shapley value for passenger 1.

What does it represent? What does it mean that the value is 1?

The Shapley decomposition

$v(\{\}) = 0 \quad (\text{no passengers costs nothing})$
$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$
$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$
$v(\{1,2,3\}) = 10$

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} (v(S \cup \{i\}) - v(S))$$

Sets excluding passenger 1: { }, {2}, {3}, {2,3}

$$\phi_1 = \frac{1(3-0-1)!}{3!}(v(\{1\}) - v(\{\})) + \frac{1(3-1-1)!}{3!}(v(\{1,2\}) - v(\{2\})) + \frac{1(3-1-1)!}{3!}(v(\{1,3\}) - v(\{3\})) + \frac{1(3-2-1)!}{3!}(v(\{1,2,3\} - v(\{2,3\})) = 1$$

ϕ_1 is the Shapley value for passenger 1.

It represents the number of kilometers passenger 1 should pay for, given the characteristic function values.

The Shapley decomposition

$v(\{\}) = 0 \quad (\text{no passengers costs nothing})$
$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$
$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$
$v(\{1,2,3\}) = 10$

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

Performing the calculation for all three passengers, *i*=1,2,3 in the Shapley decomposition formula yields

$$\phi_1 = \tfrac{1}{3}\left(v(\{1,2,3\} - v(\{2,3\})\right) + \tfrac{1}{6}\left(v(\{1,2\}) - v(\{2\})\right) + \tfrac{1}{6}\left(v(\{1,3\}) - v(\{3\})\right) + \tfrac{1}{3}\left(v(\{1\}) - v(\emptyset)\right) = 1$$
$$\phi_2 = \tfrac{1}{3}\left(v(\{1,2,3\} - v(\{1,3\})\right) + \tfrac{1}{6}\left(v(\{1,2\}) - v(\{1\})\right) + \tfrac{1}{6}\left(v(\{2,3\}) - v(\{3\})\right) + \tfrac{1}{3}\left(v(\{2\}) - v(\emptyset)\right) = 3$$
$$\phi_3 = \tfrac{1}{3}\left(v(\{1,2,3\} - v(\{1,2\})\right) + \tfrac{1}{6}\left(v(\{1,3\}) - v(\{1\})\right) + \tfrac{1}{6}\left(v(\{2,3\}) - v(\{2\})\right) + \tfrac{1}{3}\left(v(\{3\}) - v(\emptyset)\right) = 6$$

The Shapley decomposition

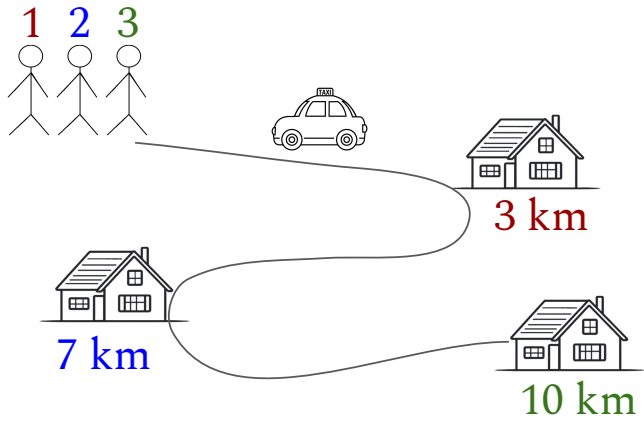
$v(\{\}) = 0 \quad (\text{no passengers costs nothing})$
$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$
$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$
$v(\{1,2,3\}) = 10$

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

What do these values mean?

$$\phi_1 = \tfrac{1}{3}\left(v(\{1,2,3\} - v(\{2,3\})\right) + \tfrac{1}{6}\left(v(\{1,2\}) - v(\{2\})\right) + \tfrac{1}{6}\left(v(\{1,3\}) - v(\{3\})\right) + \tfrac{1}{3}\left(v(\{1\}) - v(\emptyset)\right) = 1$$
$$\phi_2 = \tfrac{1}{3}\left(v(\{1,2,3\} - v(\{1,3\})\right) + \tfrac{1}{6}\left(v(\{1,2\}) - v(\{1\})\right) + \tfrac{1}{6}\left(v(\{2,3\}) - v(\{3\})\right) + \tfrac{1}{3}\left(v(\{2\}) - v(\emptyset)\right) = 3$$
$$\phi_3 = \tfrac{1}{3}\left(v(\{1,2,3\} - v(\{1,2\})\right) + \tfrac{1}{6}\left(v(\{1,3\}) - v(\{1\})\right) + \tfrac{1}{6}\left(v(\{2,3\}) - v(\{2\})\right) + \tfrac{1}{3}\left(v(\{3\}) - v(\emptyset)\right) = 6$$

The Shapley decomposition



This result tells us that the fair and unique way to distribute the cost of the journey, is when

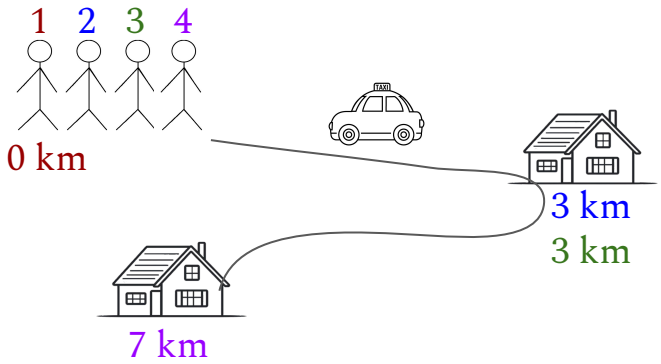
passenger 1 pays for 1 km,

passenger 2 pays for 3 km,

passenger 3 pays for 6 km.

$$\phi_1 = \frac{1}{3}\left(v(\{1,2,3\} - v(\{2,3\})\right) + \frac{1}{6}\left(v(\{1,2\}) - v(\{2\})\right) + \frac{1}{6}\left(v(\{1,3\}) - v(\{3\})\right) + \frac{1}{3}\left(v(\{1\}) - v(\emptyset)\right) = 1$$
$$\phi_2 = \frac{1}{3}\left(v(\{1,2,3\} - v(\{1,3\})\right) + \frac{1}{6}\left(v(\{1,2\}) - v(\{1\})\right) + \frac{1}{6}\left(v(\{2,3\}) - v(\{3\})\right) + \frac{1}{3}\left(v(\{2\}) - v(\emptyset)\right) = 3$$
$$\phi_3 = \frac{1}{3}\left(v(\{1,2,3\} - v(\{1,2\})\right) + \frac{1}{6}\left(v(\{1,3\}) - v(\{1\})\right) + \frac{1}{6}\left(v(\{2,3\}) - v(\{2\})\right) + \frac{1}{3}\left(v(\{3\}) - v(\emptyset)\right) = 6$$

The Shapley decomposition



New game! One more player, and

Player 1 stays behind.

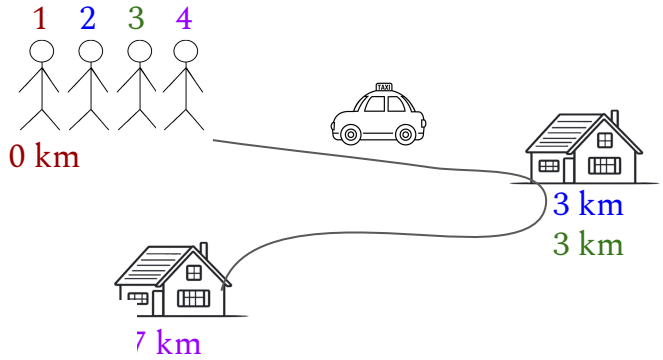
Players 2 and 3 live together.

Player 4 goes 7 km.

What are the characteristic function values?

$$v(\{\}) = ?$$
$$v(\{1\}) = ? \quad v(\{2\}) = ? \quad v(\{3\}) = ? \quad v(\{4\}) = ?$$
$$v(\{1,2\}) = ? \quad v(\{1,3\}) = ? \quad v(\{1,4\}) = ? \quad v(\{2,3\}) = ? \quad v(\{2,4\}) = ?$$
$$v(\{1,2,3\}) = ? \quad v(\{1,2,4\}) = ? \quad v(\{1,3,4\}) = ? \quad v(\{2,3,4\}) = ?$$

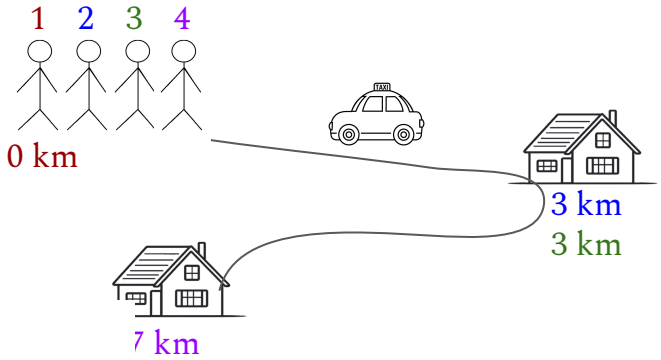
The Shapley decomposition



Characteristic function values:

$$v(\{\}) = 0$$
$$v(\{1\}) = 0 \quad v(\{2\}) = 3 \quad v(\{3\}) = 3 \quad v(\{4\}) = 7$$
$$v(\{1,2\}) = 3 \quad v(\{1,3\}) = 3 \quad v(\{1,4\}) = 7 \quad v(\{2,3\}) = 3 \quad v(\{2,4\}) = 7$$
$$v(\{1,2,3\}) = 3 \quad v(\{1,2,4\}) = 7 \quad v(\{1,3,4\}) = 7 \quad v(\{2,3,4\}) = 7$$

The Shapley decomposition



Characteristic function values:

$$v(\{\}) = 0$$
$$v(\{1\}) = 0 \quad v(\{2\}) = 3 \quad v(\{3\}) = 3 \quad v(\{4\}) = 7$$
$$v(\{1,2\}) = 3 \quad v(\{1,3\}) = 3 \quad v(\{1,4\}) = 7 \quad v(\{2,3\}) = 3 \quad v(\{2,4\}) = 7$$
$$v(\{1,2,3\}) = 3 \quad v(\{1,2,4\}) = 7 \quad v(\{1,3,4\}) = 7 \quad v(\{2,3,4\}) = 7$$

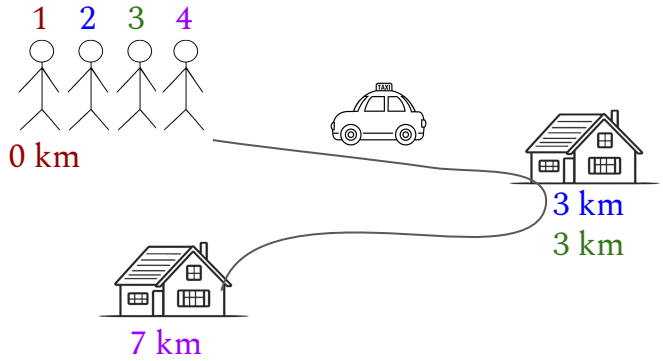
Use the formula for the Shapley decomposition to do the calculation for the different players

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

and check your answer :)

$$\phi_1 = 0 \quad \phi_2 = 1 \quad \phi_3 = 1 \quad \phi_4 = 5$$

The Shapley decomposition

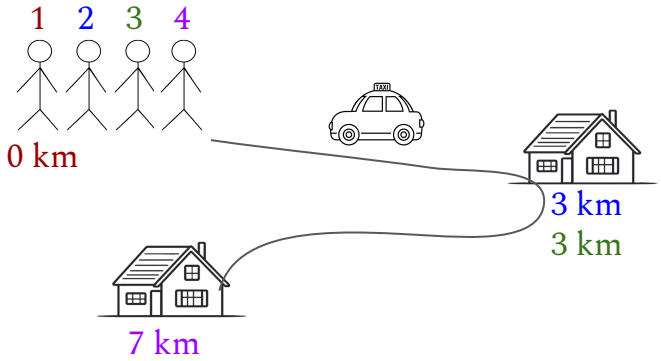


The shapley values are:

$$\phi_1 = 0 \quad \phi_2 = 1 \quad \phi_3 = 1 \quad \phi_4 = 5$$

What do we see?

The Shapley decomposition



The shapley values are:

$$\phi_1 = 0 \quad \phi_2 = 1 \quad \phi_3 = 1 \quad \phi_4 = 5$$

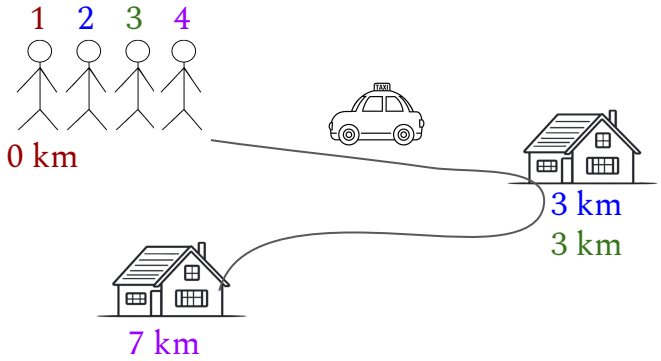
We see that:

Player 1 doesn't travel, and doesn't pay anything

Players 2 and 3 travel the same distance, and pay the same

The sum of all the Shapley values equals the value of the **grand coalition**: $\sum_{i=1}^{4} \phi_i = v(\{1,2,3,4\})$

The Shapley decomposition



The shapley values are:

$$\phi_1 = 0 \quad \phi_2 = 1 \quad \phi_3 = 1 \quad \phi_4 = 5$$

We see that:

Player 1 doesn't travel, and doesn't pay anything ← **Dummy**

Players 2 and 3 travel the same distance, and pay the same ← **Symmetry**

The sum of all the Shapley values equals the value of the grand coalition ← **Efficiency**

The four Shapley axioms

Dummy:

Symmetry:

Efficiency:

The four Shapley axioms

Dummy: The Shapley value is zero for players who don't contribute

Symmetry: Equally productive players receive the same pay

Efficiency: The entire output is shared among the players

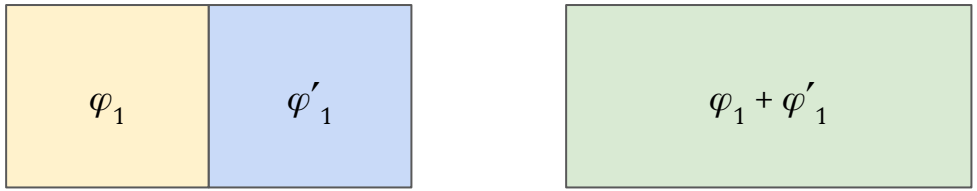
The four Shapley axioms

Dummy: The Shapley value is zero for players who don't contribute

Symmetry: Equally productive players receive the same pay

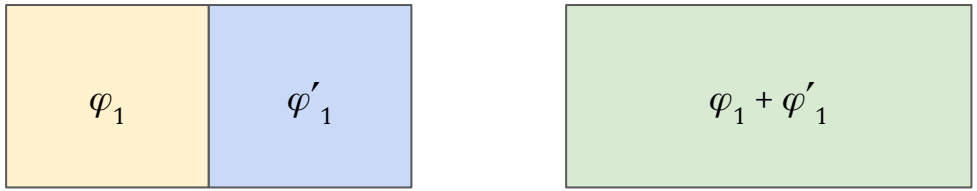
Efficiency: The entire output is shared among the players

Additivity: Given two games, the Shapley value of each player in the composite game is the same as the sum of the Shapley values for each player in the separate games.



The Shapley axioms, formally

Symmetry: If $v(S \cup \{i\}) = v(S \cup \{j\})\ \forall S \setminus \{i, j\}$ then $\phi_i = \phi_j$
Efficiency: $\sum_{i=1}^{n} \phi_i(X, v) = v(X)$
Dummy: If $v(S \cup \{i\}) - v(S) = 0\ \forall S$ then $\phi_i = 0$
Additivity: $\phi(X, v + w) = \phi(X, v) + \phi(X, w)$ for any games (X, v) and (X, w)



The Shapley axioms, formally

Symmetry: If $v(S \cup \{i\}) = v(S \cup \{j\})\ \forall S \setminus \{i, j\}$ then $\phi_i = \phi_j$
Efficiency: $\sum_{i=1}^{n} \phi_i(X, v) = v(X)$
Dummy: If $v(S \cup \{i\}) - v(S) = 0\ \forall S$ then $\phi_i = 0$
Additivity: $\phi(X, v + w) = \phi(X, v) + \phi(X, w)$ for any games (X, v) and (X, w)

Fun fact: The symmetry, efficiency and additivity can be combined into a single one:

balanced contributions by (Myerson 1977/1980):

$$\phi_i(N, v) - \phi_i(N \backslash \{j\}, v) = \phi_j(N, v) - \phi_j(N \backslash \{i\}, v)$$

The Shapley axioms, formally

Symmetry: If $v(S \cup \{i\}) = v(S \cup \{j\})\ \forall S \setminus \{i, j\}$ then $\phi_i = \phi_j$
Efficiency: $\sum_{i=1}^{n} \phi_i(X, v) = v(X)$
Dummy: If $v(S \cup \{i\}) - v(S) = 0\ \forall S$ then $\phi_i = 0$
Additivity: $\phi(X, v + w) = \phi(X, v) + \phi(X, w)$ for any games (X, v) and (X, w)

Fun fact: The symmetry, efficiency and additivity can be combined into a single one:

balanced contributions by (Myerson 1977/1980):

$$\phi_i(N, v) - \phi_i(N \backslash \{j\}, v) = \phi_j(N, v) - \phi_j(N \backslash \{i\}, v)$$

The difference in the Shapley values for player *i* when including and excluding player *j* equals the difference in the Shapley values for player *j* when including and excluding player *i*.

Homework, and strategy for getting the most out of tomorrow's lecture

You will get much more out of tomorrow's lecture if you do today's homework.

Part 1: Calculate the Shapley value by hand, on the toy problem following right away.

Part 2: Write code for calculating the Shapley value, using the toy problem for guidance.

Part 3: Adapt this code to a machine learning setting.

Homework part 1: manual calculation

Mice caught alone:

Mice caught in coalitions:

Shapley value per cat:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S) \right), \quad i = 1, \ldots, N$$

Mice caught alone:

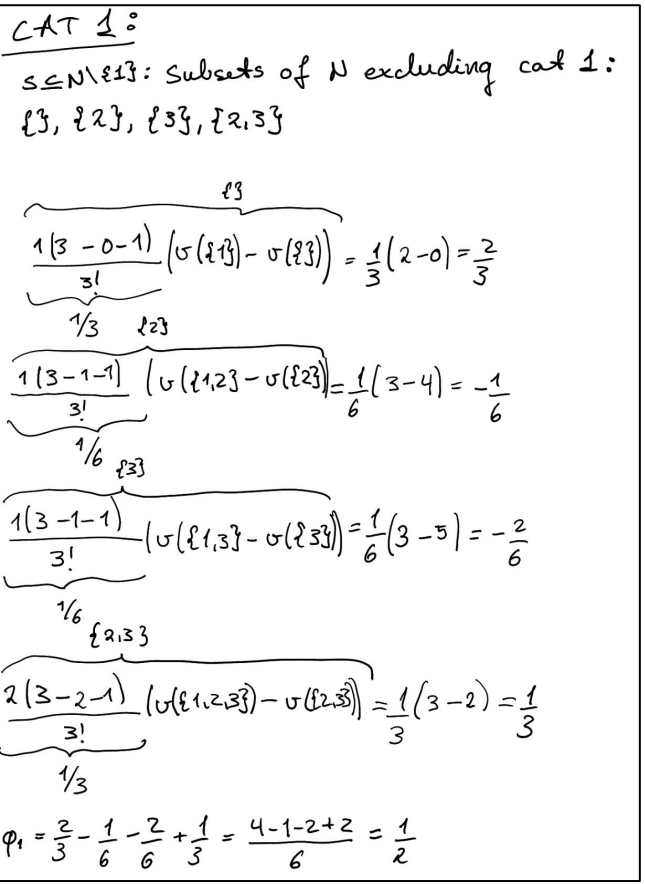
Mice caught in coalitions:

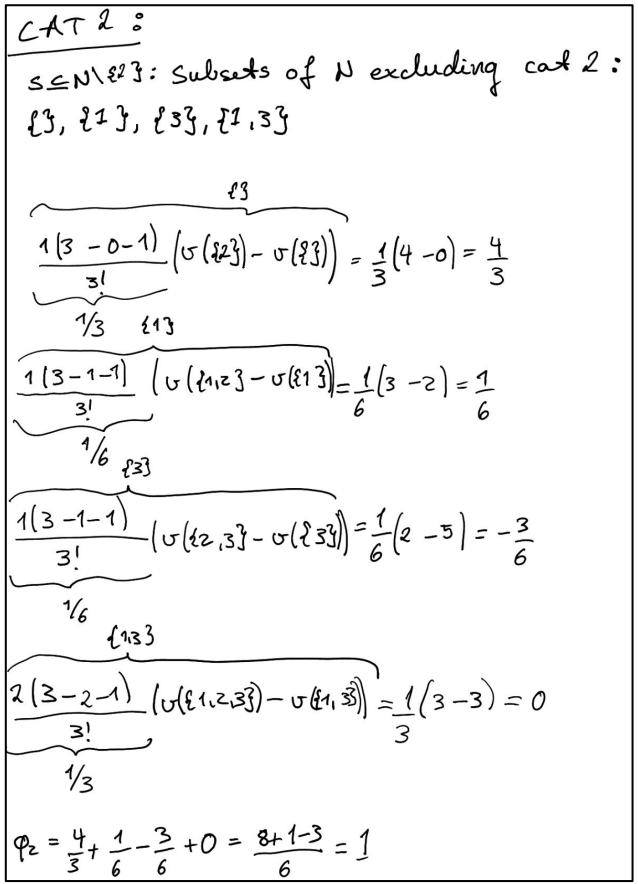
Shapley value per cat:

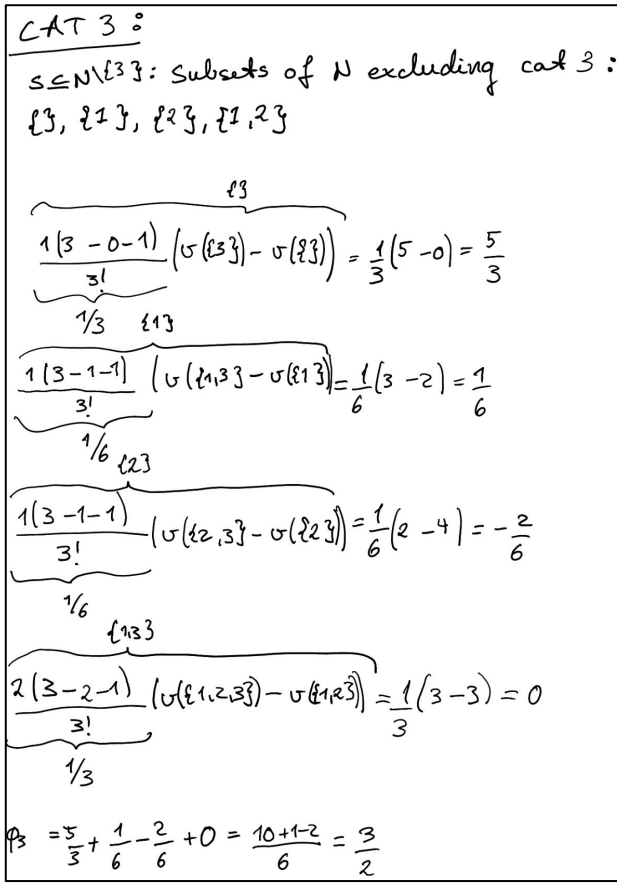
$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right), \quad i = 1, \ldots, N$$

½ 1 3/2

And here are the calculations in my glorious hand writing.







Homework part 1:

Calculate the Shapley value for each of the three cats.

While you calculate, think about how you would code this.

Are any operations repeated?

Are there any symmetries?

How many terms do you have to calculate?

Coding the Shapley decomposition

What to do before implementing an equation?

Stare it down.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

Coding the Shapley decomposition

The Shapley value for player i is

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

Coding the Shapley decomposition

The Shapley value for player i is

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

the average* of the player's

Coding the Shapley decomposition

The Shapley value for player i is

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

the average* of the player's marginal contribution in each coalition

Coding the Shapley decomposition

The Shapley value for player i is

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

the average* of the player's marginal contribution in each coalition

*over all coalitions.

Coding the Shapley decomposition

The Shapley value for player i is the average of the player's marginal contribution in each coalition, over all coalitions.

The *marginal contribution* of a player is the value with and without the player.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} (v(S \cup \{i\}) - v(S))$$

Coding the Shapley decomposition

Remember the toy problem.

The characteristic function is symmetric, i.e. yields the same value for

coalition {1,2} adding player 3,
coalition {1,3} adding player 2, and
coalition {2,3} adding player 1

⇒We should only calculate $v(\{1,2,3\})$ once, not four times.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} (v(S \cup \{i\}) - v(S))$$

Coding the Shapley decomposition

The same happens for

coalition {1} adding player 2,
coalition {2} adding player 1.

coalition {2} adding player 3,
coalition {3} adding player 2.

... etc.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} (v(S \cup \{i\}) - v(S))$$

Coding the Shapley decomposition

The characteristic function is symmetric.

Tip 1:

Since $v(\{j,i\}) = v(\{i,j\})$, the characteristic function values should be put into a dictionary

`cf_dict = {set of players : value}`

```
players = [1,2,3]
cf_dict = {():0, (1,):3, (2,):7, (3,):10,
           (1,2):7, (1,3):10, (2,3):10,
           (1,2,3):10}

print(calc_shapley_value(1, players, cf_dict))
print(calc_shapley_value(2, players, cf_dict))
print(calc_shapley_value(3, players, cf_dict))
```

```
1.0
3.0
6.0
```

Coding the Shapley decomposition

Look at the prefactor...

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \overbrace{\frac{|S|!(N - |S| - 1)!}{N!}} (v(S \cup \{i\}) - v(S))$$

We see that:

- |S| depends on coalition **sizes** (not members). *N* is constant.
- The value of the prefactor does not depend on the specific problem, meaning the characteristic function values.

Coding the Shapley decomposition

Look at the prefactor…

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \overbrace{\frac{|S|!(N - |S| - 1)!}{N!}} (v(S \cup \{i\}) - v(S))$$

We see that:

- |S| denotes coalition *size* (independent of members), and N is constant. ⇒ we can move N outside the summation.
- Summing over all coalition *sizes* means all members are included. But the order doesn't matter.

Coding the Shapley decomposition

We can thus rewrite the prefactor

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \overbrace{\frac{|S|!(N - |S| - 1)!}{N!}} (v(S \cup \{i\}) - v(S))$$

to instead summing over all coalition sizes k and including all coalitions of that size,

$$\phi_i = \frac{1}{N} \sum_{k=0}^{N-1} \frac{1}{|S_k|} \sum_{S \in S_k} \left(v(S \cup \{i\}) - v(S) \right)$$

since

$$|S_k| = \binom{N-1}{k} = \frac{(N-1)!}{k!\,(N-k-1)!}$$

Coding the Shapley decomposition

This formulation focuses on the coalition members

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S) \right)$$

while this formulation focuses on coalition sizes

$$\phi_i = \frac{1}{N} \sum_{k=0}^{N-1} \frac{1}{|S_k|} \sum_{S \in S_k} \left(v(S \cup \{i\}) - v(S) \right)$$

Tip 2: and this last one is easiest to implement with a `cf_dict`

Homework part 2: implement the Shapley value calculation

pseudocode

```
def calc_shapley_value(player_index, all_players, cf_dict):
     make sure that player_index isn't in all_players
     make a list of all coalition sizes from 0 to num_players
     initialize total value = 0
     for all coalition sizes:
          initialize coalition_value = 0
          find all coalitions of size s
          for all coalitions of size s:
               calculate the difference in value of the coalition with/without the player
               increment the coalition value
          calculate the average value of coalitions of size s
          increment the total value by the average coalition value
     calculate and return the average total value
```

This code requires that `cf_dict` exists

From game theory to data

From game theory to machine learning, step 1

The Shapley decomposition attributes gain among players.

It takes as input a set function $v: 2^N \rightarrow R$ and produces attributions φ_i for each player $i \in N$.

The attributions add up to $v(N)$.

Before going to machine learning, we need to handle data.

Let's use the Shapley decomposition for a little data analysis.

The diabetes dataset

```
X_data, y_data = load_diabetes(scaled=True, as_frame=True, return_X_y=True)
features = X_data.columns.to_numpy()
features
```

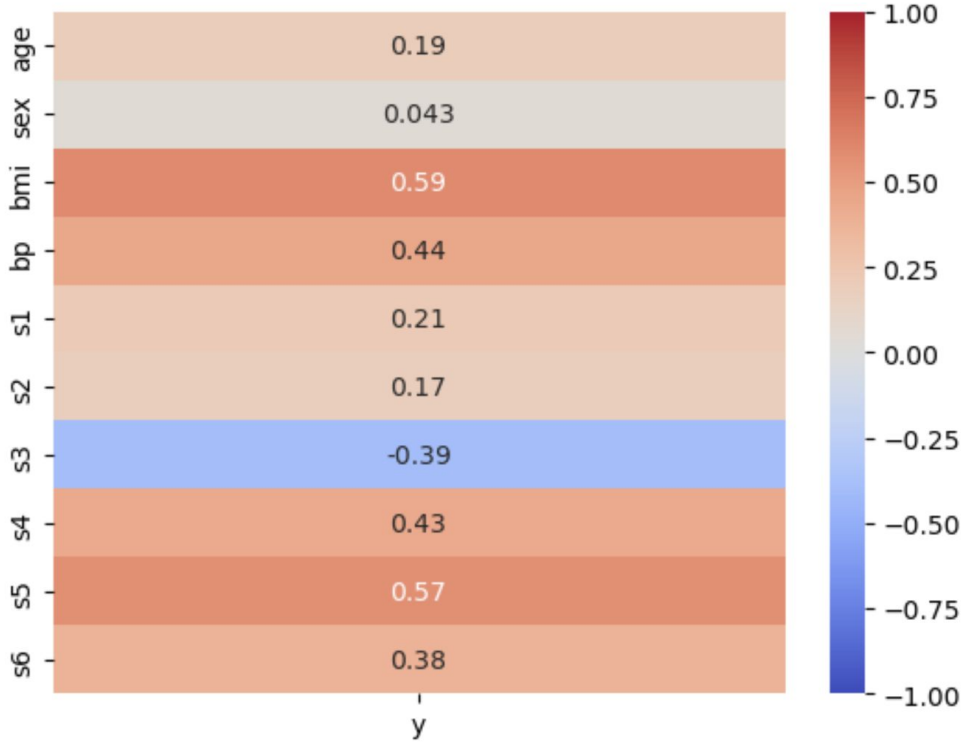
```
array(['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6'],
      dtype=object)
```

```
df = pd.concat([X_data, y_data.rename("y")], axis=1)
corr = df.corr(numeric_only=True)
sns.heatmap(corr.loc[X_data.columns, ["y"]], annot=True, vmin=-1, vmax=1, cmap="coolwarm")
plt.show()
```

Let's check out how the features correlate with the target.

.corr uses the Pearson correlation, which measures linear dependence.

The diabetes dataset



This shows how the features correlate linearly with the target.

The Shapley decomposition of features

But these are just the correlations of each single feature and the target. We care about feature interactions!

Let's use the Shapley decomposition.

First, we need a characteristic function.

Let's stick to linear correlation for starters.

The Shapley decomposition of features

We start by using Pearson's R^2, aka the coefficient of determination, as the characteristic function.

Intuitively: the R^2 tells us how much of the variance in a variable is explained by a model. 0 is none, 1 is all.

```python
def characteristic_function_r2(x, y, coalition):
    """
    Returns the coefficient of determination between the indices of x indicated in coalition and y
    Input:
        x, numpy array shape (#samples, #features)
        y, numpy array shape (#samples, )
        coalition, tuple of indices of features to include
    """
    if len(coalition)==0:
        return 0.0
    x = x[:, coalition]

    # --- Coefficient of determination, R2
    det_C_xy = np.linalg.det(np.corrcoef(x.T, y))
    if len(coalition)==1:
        det_C_x = 1
    else:
        det_C_x = np.linalg.det(np.corrcoef(x.T))

    return (1 - det_C_xy/det_C_x)
```

The Shapley decomposition of features in the diabetes dataset

Characteristic function: R^2

Function `make_cf_dict()` (from me :))

```
cf_dict_R2 = make_cf_dict(X_data.to_numpy(), y_data.to_numpy(), characteristic_function_r2)
print(len(cf_dict_R2))
```

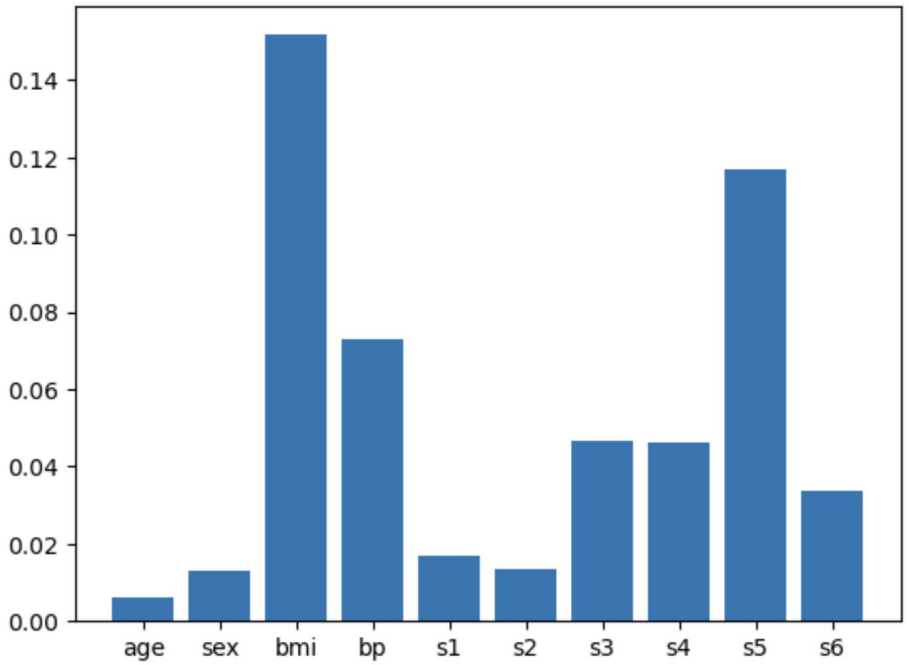
```
1024
```

That's a lot of entries...

The Shapley decomposition of features in the diabetes dataset

```
shapley_values = calc_shapley_values(X_data.to_numpy(), cf_dict_R2)
```

```
plt.bar(range(len(shapley_values)), shapley_values)
plt.xticks(range(len(shapley_values)), features)
plt.show()
```



Characteristic function: R^2

Function `make_cf_dict()` (from me :))

Plug, play & visualize.

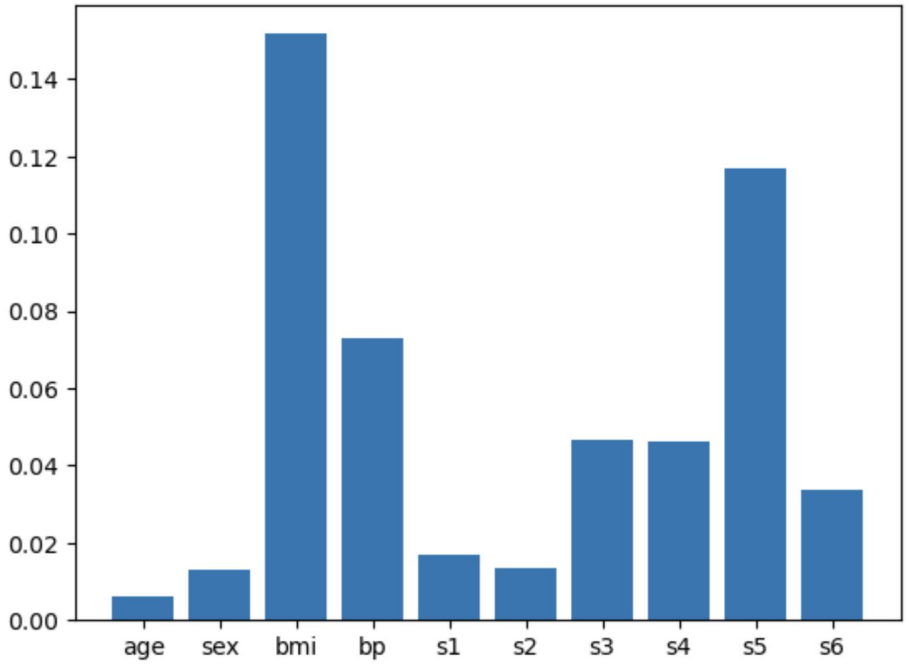
These are the Shapley values of the features in the diabetes dataset.

What do they mean?

The Shapley decomposition of features in the diabetes dataset

```
shapley_values = calc_shapley_values(X_data.to_numpy(), cf_dict_R2)
```

```
plt.bar(range(len(shapley_values)), shapley_values)
plt.xticks(range(len(shapley_values)), features)
plt.show()
```



Characteristic function: R^2

Function `make_cf_dict()` (from me :))

Plug, play & visualize.

These are the Shapley values of the features in the diabetes dataset.

They tell us each feature's fair share of ability to capture the variance of the target.

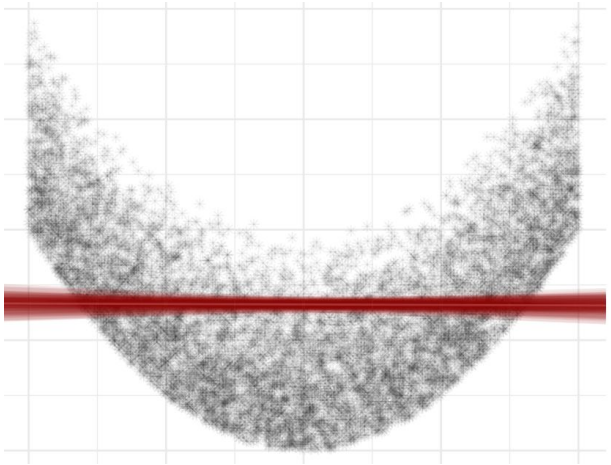
Sanity check

```
print("Sum of the Shapley values: ", sum(shapley_values))
print("Value of the grand coalition: ", cf_dict_R2[list(cf_dict_R2.keys())[-1]])
```

```
Sum of the Shapley values:  0.51774842222035
Value of the grand coalition:  0.5177484222203501
```

The Shapley decomposition of features

The coefficient of determination R^2 assumes *linear* dependence.



$y = a\, x^2$

$R^2 \approx 0$ despite strong (non-linear) dependence.

Estimate of R^2 =0.0043 with 95% bootstrap CI (0.001, 0.013) over 100 fits.

The Shapley decomposition of features

The coefficient of determination R^2 assumes *linear* dependence.

Our features are not necessarily linearly correlated with the target.

Another option is the distance correlation (from the dcor library; `pip install dcor`)

```
def characteristic_function_dcor(x, y, coalition):
    if len(coalition)==0:
        return 0.0

    x = x[:, coalition]

    return dcor.distance_correlation(x,y)
```

The Shapley decomposition of features in the diabetes dataset

Characteristic function: dcor

Function `make_cf_dict()`

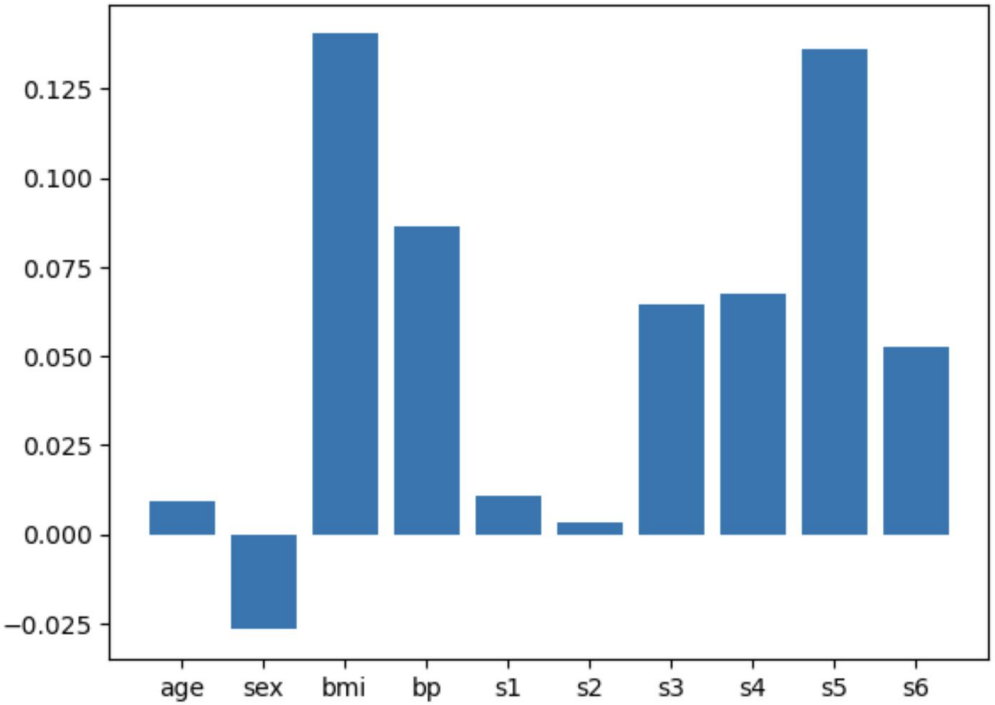
Calculate the Shapley values and visualize

```
cf_dict_dcor = make_cf_dict(X_data.to_numpy(), y_data.to_numpy(), characteristic_function_dcor)
shapley_values_dcor = calc_shapley_values(X_data, cf_dict_dcor)
```

```
plt.bar(range(len(shapley_values_dcor)), shapley_values_dcor)
plt.xticks(range(len(shapley_values_dcor)), features)
plt.show()
```

The Shapley decomposition of features in the diabetes dataset

Characteristic function: dcor

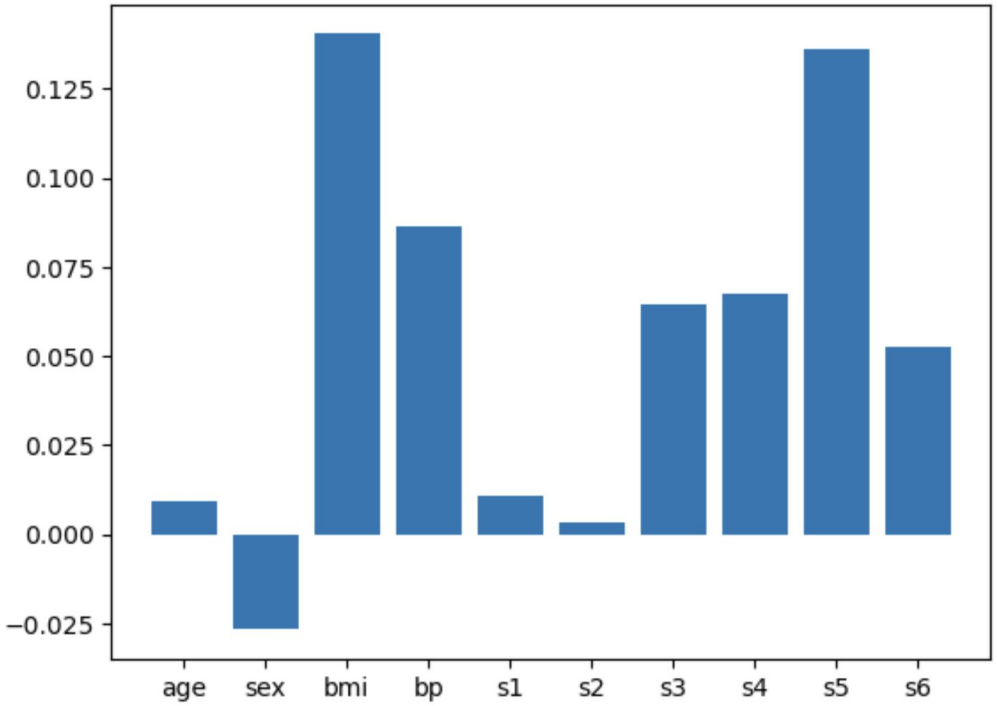


What do we see?

What are these Shapley values?

The Shapley decomposition of features in the diabetes dataset

Characteristic function: dcor



These are the Shapley values of the features in the diabetes dataset, using the distance correlation as characteristic function.

They tell us each feature's fair share of the correlation with the target.

From game theory to machine learning

From game theory to machine learning, all the way

The Shapley decomposition attributes gain among players. It takes as input a set function v: $2^N \rightarrow R$ and produces attributions φ_i for each player $i \in N$ that add up to $v(N)$

We can go from game theory to machine learning by re-interpreting…

the grand coalition N from meaning all players to meaning all data features

each i indexed player as each i indexed feature

the coalitions S from meaning coalitions of players to meaning sets of data features

the characteristic function v from characterising the game to characterising the model

From game theory to machine learning

The Shapley decomposition attributes gain among players. It takes as input a set function v: $2^N \rightarrow R$ and produces attributions φ_i for each player $i \in N$ that add up to $v(N)$

We can go from game theory to machine learning by re-interpreting...

	Game theory	**Machine learning**
N	Grand coalition	All features
S	Coalitions	Sets of features
i	Player index	Feature index
v	Characterises the game	Characterises the model

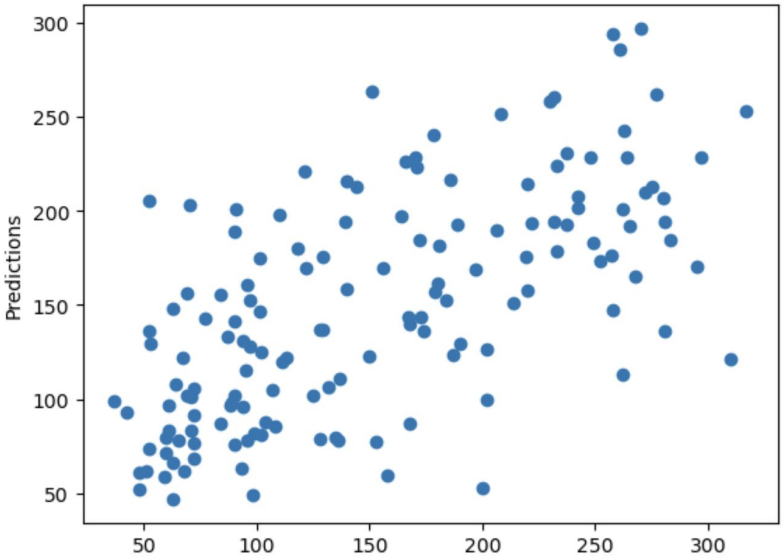
First, train a model on the data

```
x_train, x_test, y_train, y_test = train_test_split(X_data, y_data, test_size=0.33, random_state=42)

regressor = xgboost.XGBRegressor()
regressor.fit(x_train, y_train);
```

```
y_pred = regressor.predict(x_test)
```

```
plt.scatter(y_test, y_pred)
plt.xlabel("Targets")
plt.ylabel("Predictions")
plt.show()
```



Eyeballing this, do you think we have a good model?

Next, calculate Shapley values

Calculate the Shapley decomposition of the correlation between the features and the model prediction.

Compare the Shapley values

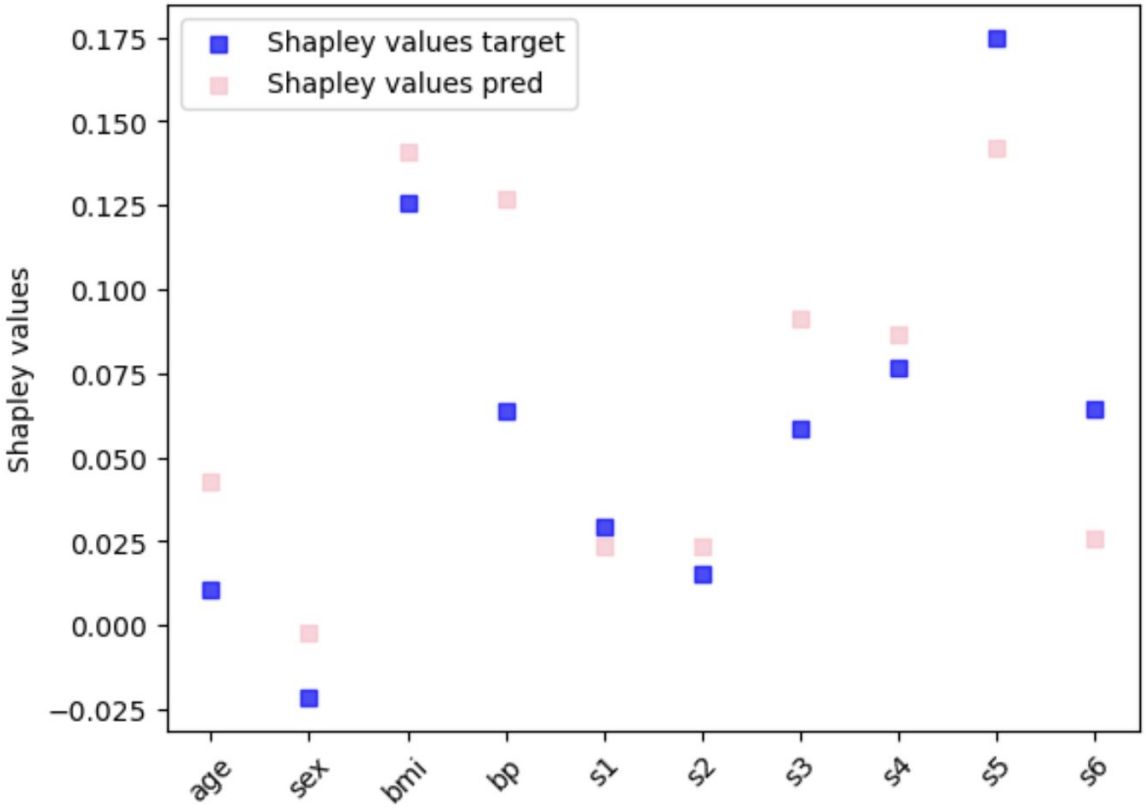
Calculate the Shapley decomposition of the correlation between the features and the model prediction

Compare the resulting Shapley values to the Shapley values from the decomposition of the correlation between the features and the data targets.

```
cf_dict_dcor_targets = make_cf_dict(x_test.to_numpy(), y_test.to_numpy(), characteristic_function_dcor)
shapley_values_targets = calc_shapley_values(x_test.to_numpy(), cf_dict_dcor_targets)
```

```
cf_dict_dcor_preds = make_cf_dict(x_test.to_numpy(), np.array(y_pred), characteristic_function_dcor)
shapley_values_preds = calc_shapley_values(x_test.to_numpy(), cf_dict_dcor_preds)
```

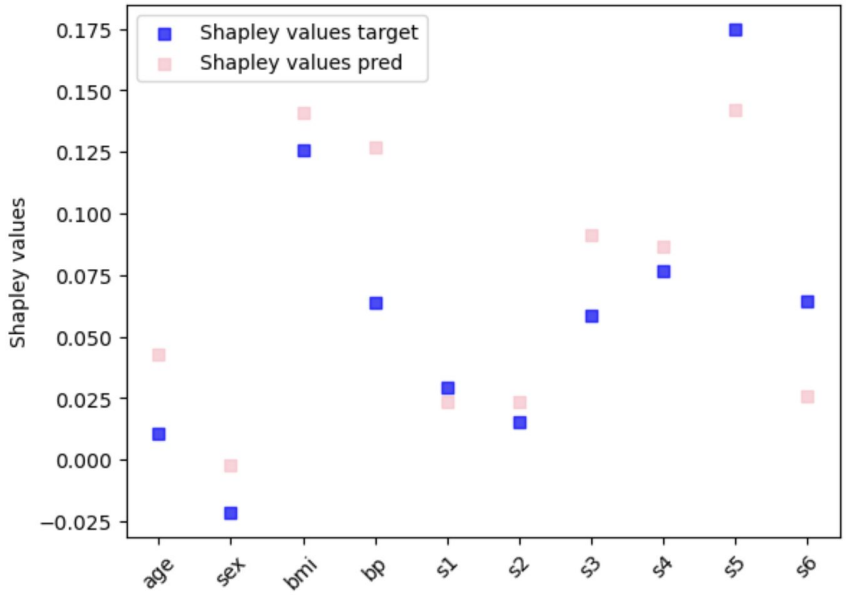
Compare the Shapley values (cf = dcor)

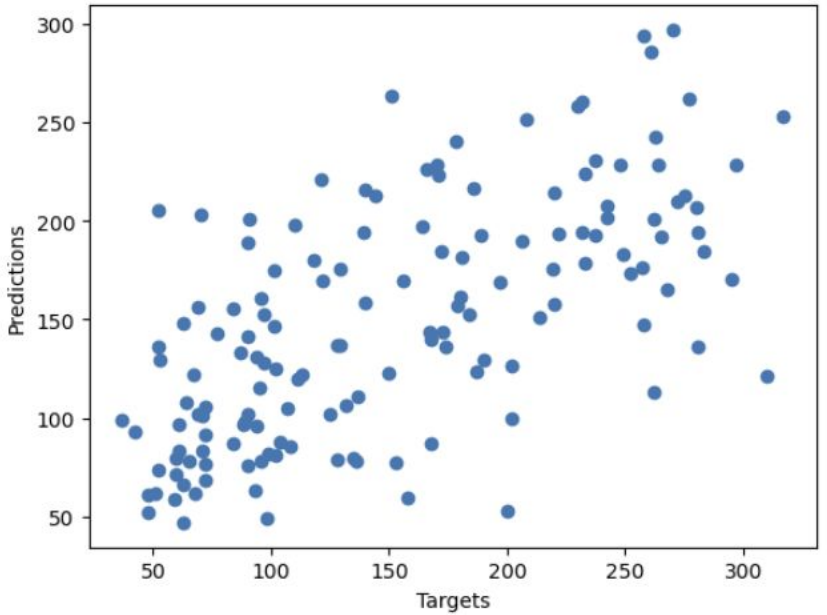


What do we see?

Has the XGBoost regressor modelled the dependence structure in the data?

Compare the Shapley values





Do we have a good model?

Can we use the Shapley values to understand which features are not well modelled?

Does it assign more/less dependence to some features than the correlation structure in the data?

Summary and way ahead

The Shapley decomposition distributes the outcome of a game fairly among the players.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

This is done by calculating the value of the game with and without each player in every coalition of players.

How many calculations does this require for N players?

Summary and way ahead

The Shapley decomposition distributes the outcome of a game fairly among the players.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

This is done by calculating the value of the game with and without each player in every coalition of players.

This requires 2^N calculations for N players.

Summary and way ahead

The Shapley decomposition distributes the outcome of a game fairly among the players.

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v(S \cup \{i\}) - v(S)\right)$$

This is done by calculating the value of the game with and without each player in every coalition of players.

This requires 2^N calculations for N players.

In a machine learning context, the players are the data features.

$\Rightarrow$ 20 features requires 2^{20} = 1,048,576 calculations.

The computational cost scales exponentially with the number of features.

Summary and way ahead

Tomorrow, we will use the Shapley decomposition for XAI.

The lecture will be more accessible to you if you do the calculations by hand and understand the code in the notebook.

Also, think about the following:

1) Most machine learning problems involve more than 20 features. The Shapley decomposition has exponential complexity in the number of features. What to do, for example for image data?

2) If interpreting a model as the game, how can we remove features as the Shapley decomposition requires?

Homework, and strategy for getting the most out of tomorrow's lecture

You will get much more out of tomorrow's lecture if you do today's homework.

Part 1: Calculate the Shapley value by hand, on the toy problem following right away.

Part 2: Write code for calculating the Shapley value, using the toy problem for guidance.

Part 3: Adapt this code to a machine learning setting.

Get familiar with the Shapley decomposition by hand and code, and meditate on it for at least 5 minutes.

See you tomorrow!