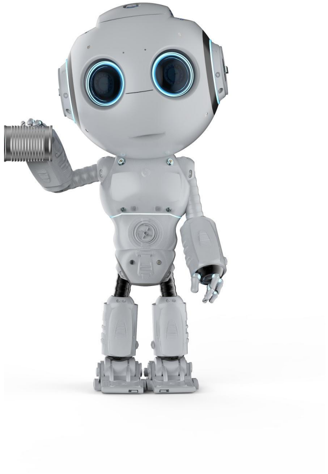
Shapley

From game theory with love

day 1



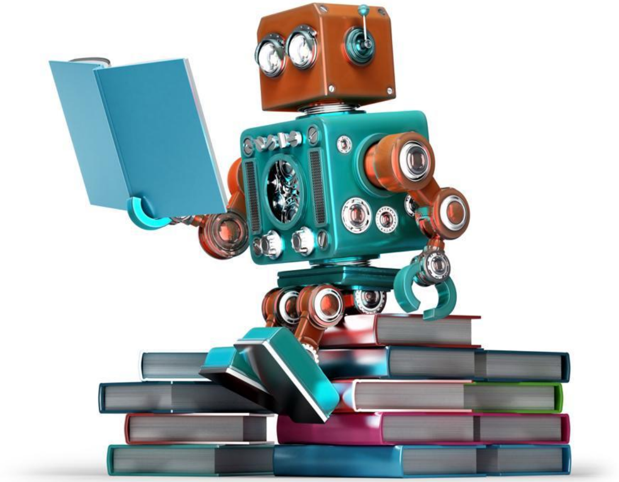
Inga Strümke

inga@simula.no / inga@strumke.com

The menu - day 1, part 1

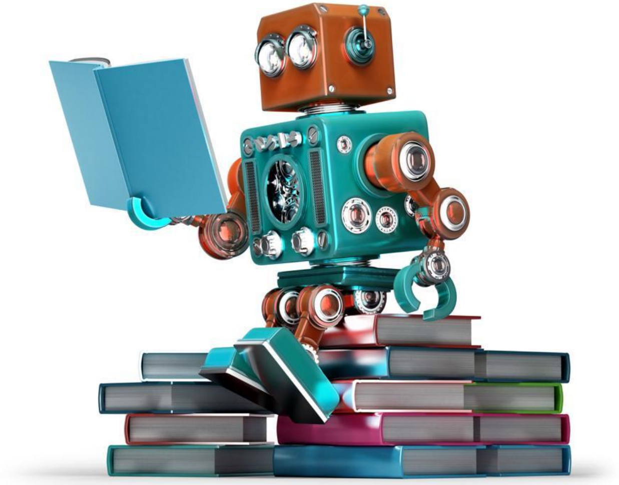
- *I can't get no explanations*
- *How does homo sapiens usually do it?*
- *Shapley values: Axioms and intuition*
- *Example: Cab sharing*
- *Hands on: A cats and mice game*
- *Shapley values for data: Dependence attribution*
- *Hands on: Jupyter (Calculate Shapley values. Boston Housing.)*

Break



The menu - day 1, part 2

- *The characteristic function*
- *Hands on: Jupyter (Characteristic functions)*
- *Explaining a model, take 1*
- *Hands on: Jupyter (Shapley values of model output)*
- *Summary*



Jupyter Notebooks: gitlab.com/Strumke/shapley_lectures

I can't get no explanations

Machine learning works like 🤯

How do machine learning methods do feature extraction and compression?

(How) do they solve the curse of dimensionality?

Are the solutions even stable?? 🧐

Nobody knows.

What can we do in the meantime?

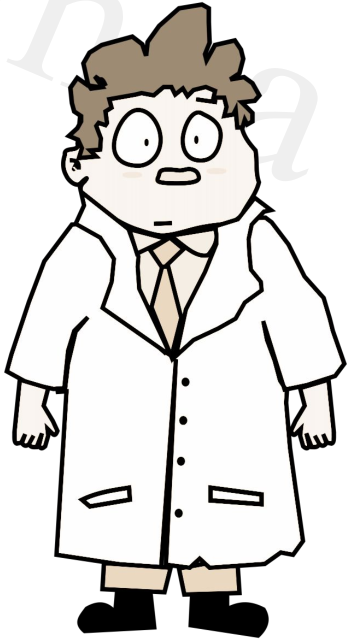
A mystery box fell from the sky

A mystery box fell from the sky

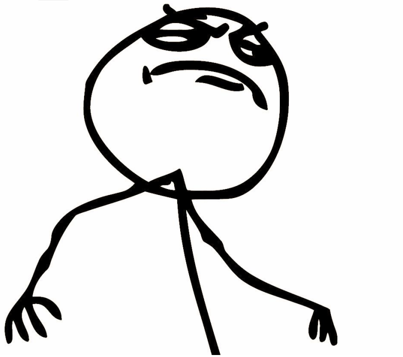
Information →

→ Predictions

A mystery box fell from the sky

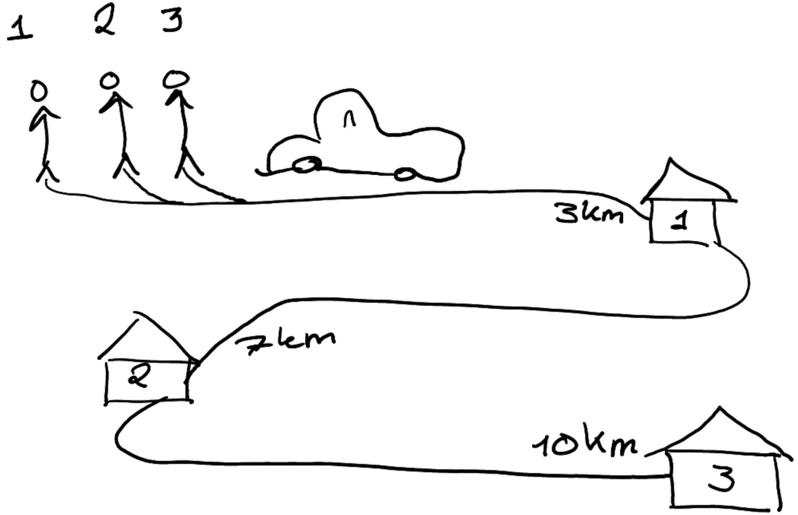


Poking the box like a boss

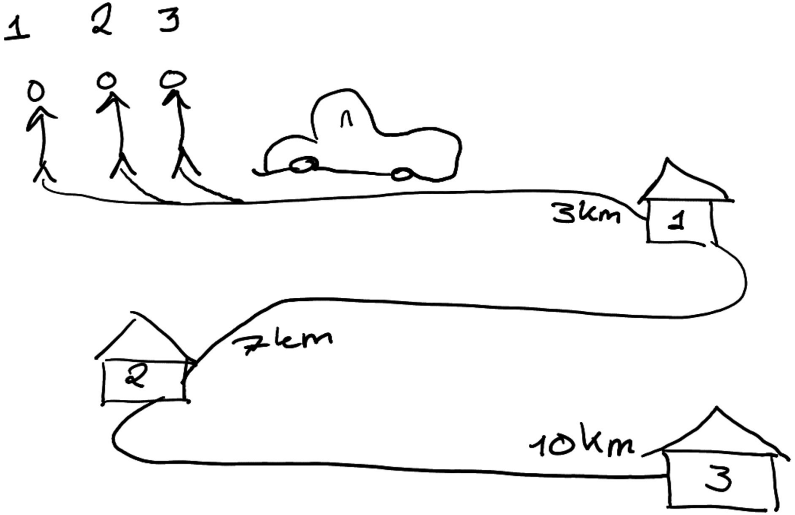


$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} (v(S \cup \{i\}) - v(S))$$

Example 1: Cab sharing



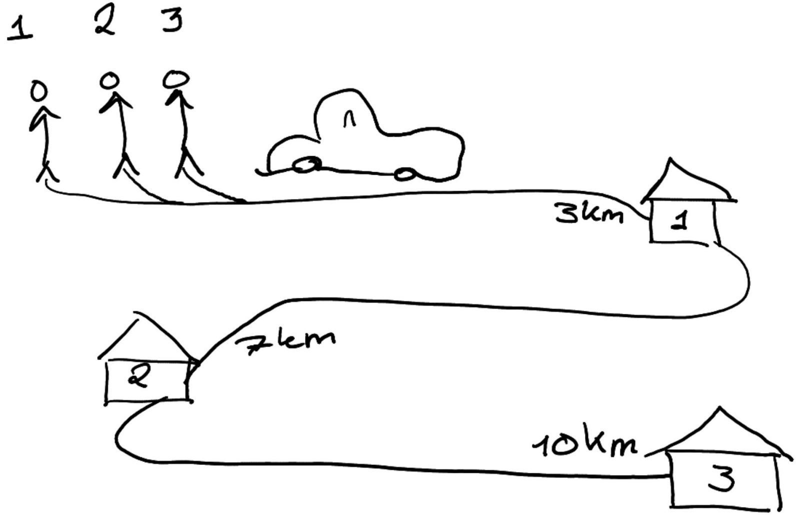
Example 1: Cab sharing



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

Characteristic function values

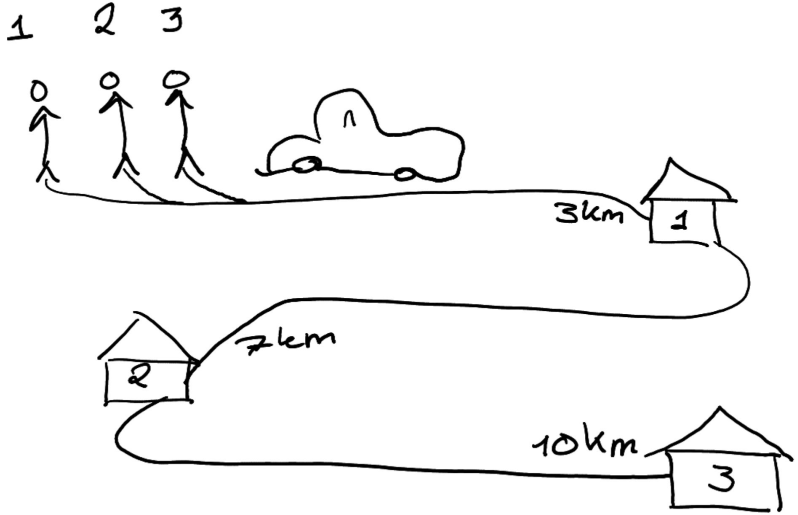
Example 1: Cab sharing



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

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

Example 1: Cab sharing



$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$

$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$

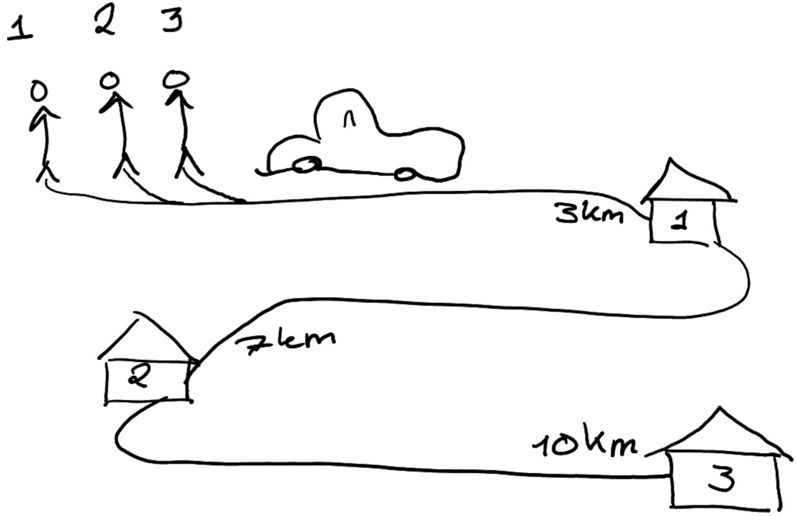
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$

$$v(\{1,2,3\}) = 10$$

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

N = total passengers

Example 1: Cab sharing



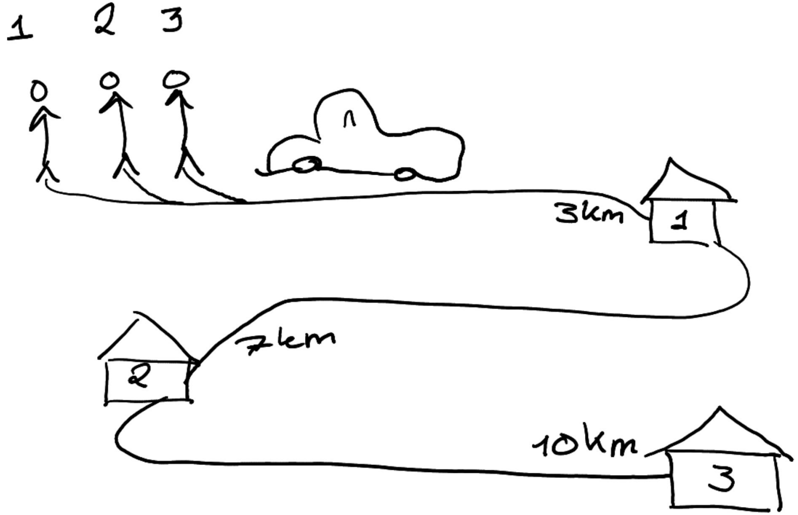
$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

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

N = total passengers

S = subsets of N

Example 1: Cab sharing



$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$

$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$

$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$

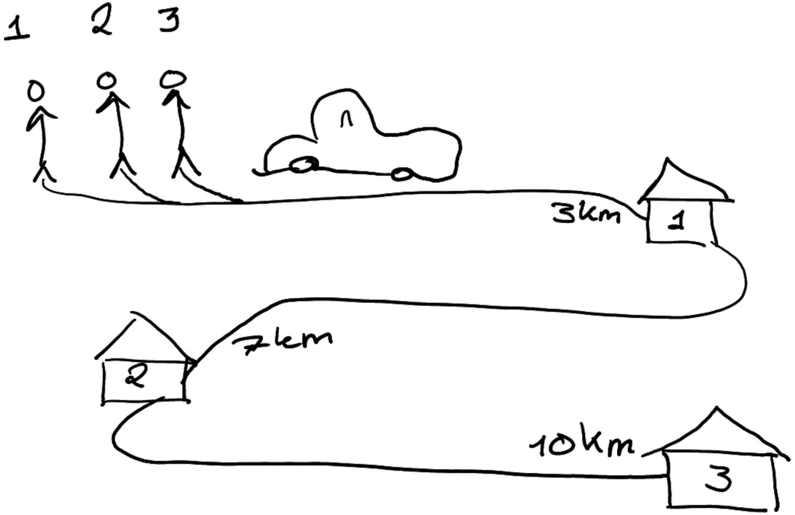
$$v(\{1,2,3\}) = 10$$

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

subsets S of N
excluding passenger i

N = total passengers
S = subsets of N

Example 1: Cab sharing

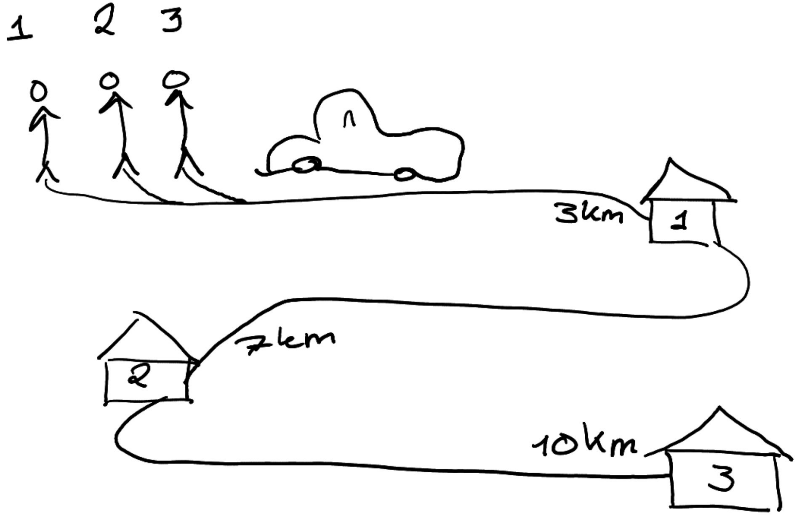


$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

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

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

Example 1: Cab sharing



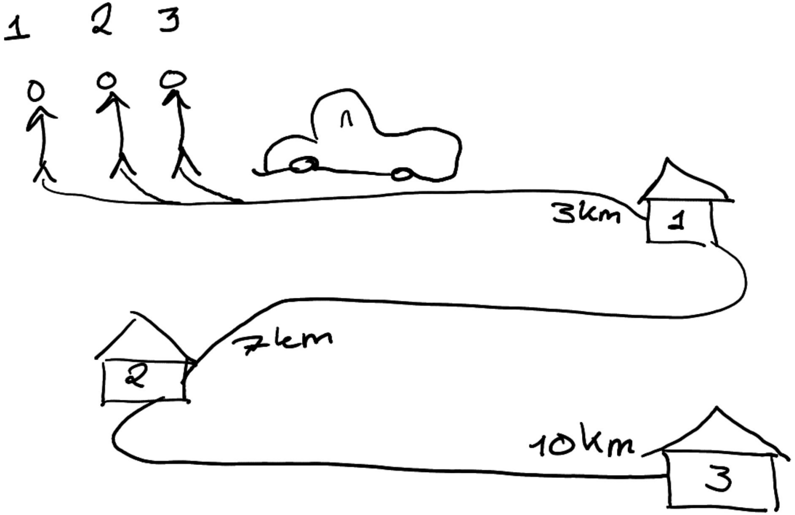
$$v(\{\}) = 0 \quad \text{(no passengers costs nothing)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 7, \quad v(\{3\}) = 10$$
$$v(\{1,2\}) = 7, \quad v(\{1,3\}) = 10, \quad v(\{2,3\}) = 10$$
$$v(\{1,2,3\}) = 10$$

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

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

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

Example 1: Cab sharing



True story: The fair and unique way to distribute the cost of the journey (at a price of 1NOK per km), is when

passenger 1 pays 1,

passenger 2 pays 3 and

passenger 3 pays 6.

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

The words and symbols

Marginal contribution: *Calculate the result with and without the player*

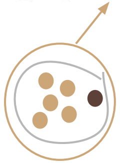
Average marginal contributions: *Sum and calculate prefactor over all coalitions (subsets)*

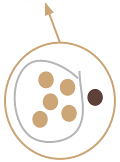
Shapley value for player i

Average over all coalitions

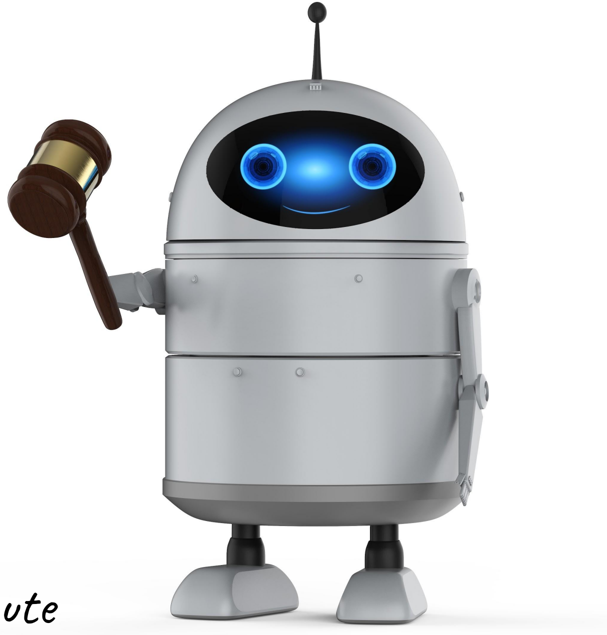
Marginal contribution in each coalition

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





Distributive justice



Symmetry: *Equally productive players receive the same pay*

Efficiency: *The entire output is shared among the players*

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

Additivity: *Train a random forest (prediction is an average of many decision trees). Can calculate the Shapley value for a feature in each tree individually, average them, and get the Shapley value for the feature value for the random forest.*

The axioms

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 in a single one:

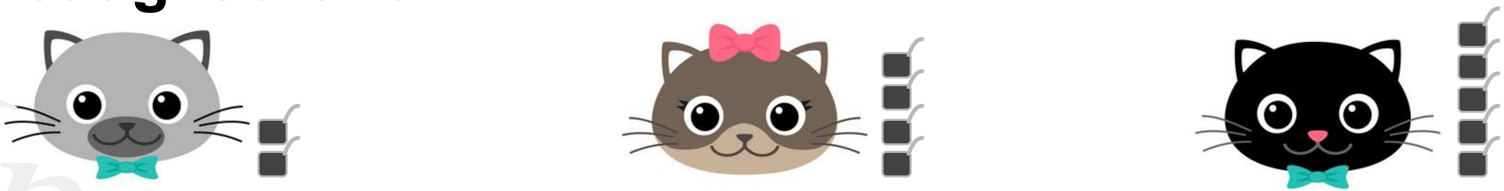
balanced contributions by (Myerson 1977/1980):

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

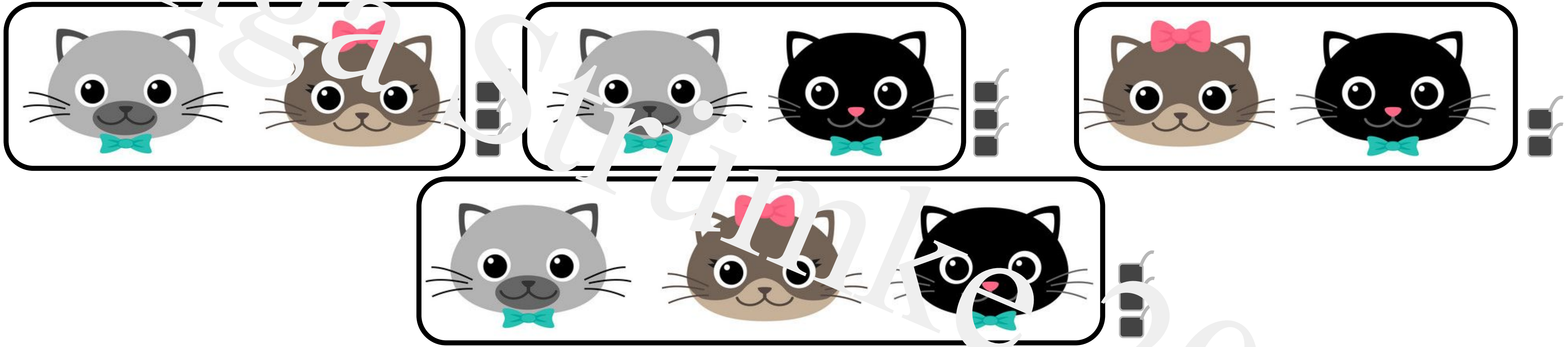
Just do it

Pencils out!

Mice caught alone:

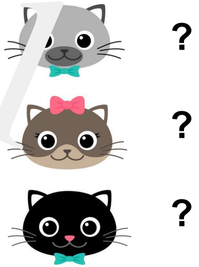


Mice caught in coalitions:

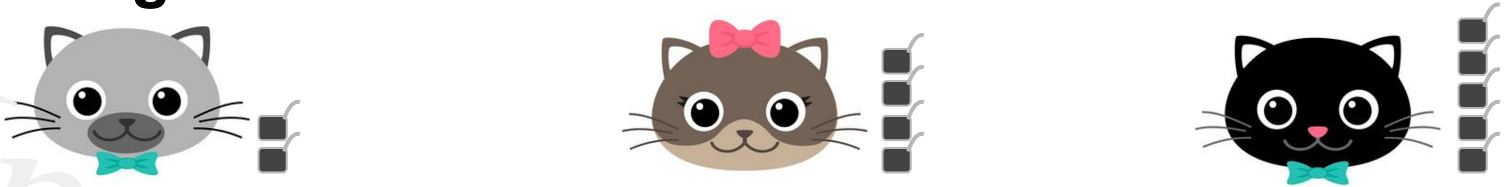


Shapley value per cat:

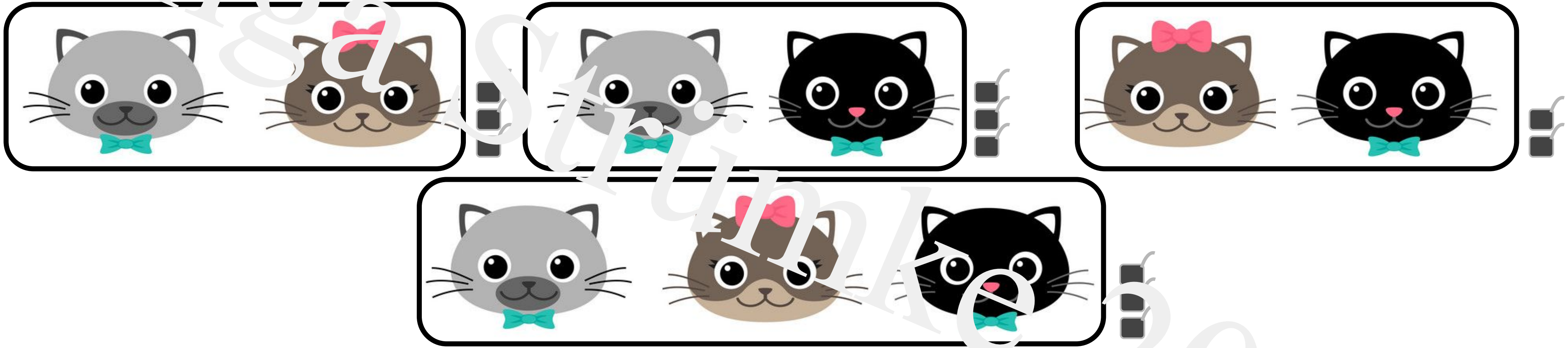
$$\phi_i(v) = \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!(N-|S|-1)!}{N!}\left(v(S\cup\{i\}) - v(S)\right), \quad i = 1,\ldots,N$$



Mice caught alone:

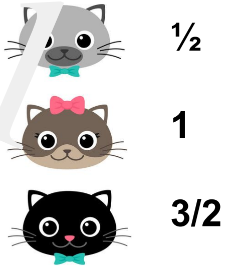


Mice caught in coalitions:



Shapley value per cat:

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



CAT 1:

$S \subseteq N\setminus\{1\}$: Subsets of N excluding cat 1:

$\{\}, \{2\}, \{3\}, \{2,3\}$

$$\overbrace{\underbrace{\frac{1(3-0-1)}{3!}}_{1/3}\left(v(\{1\})-v(\{\})\right)}^{\{\}} = \frac{1}{3}(2-0) = \frac{2}{3}$$

$$\overbrace{\underbrace{\frac{1(3-1-1)}{3!}}_{1/6}\left(v(\{1,2\}-v(\{2\})\right)}^{\{2\}} = \frac{1}{6}(3-4) = -\frac{1}{6}$$

$$\overbrace{\underbrace{\frac{1(3-1-1)}{3!}}_{1/6}\left(v(\{1,3\}-v(\{3\})\right)}^{\{3\}} = \frac{1}{6}(3-5) = -\frac{2}{6}$$

$$\overbrace{\underbrace{\frac{2(3-2-1)}{3!}}_{1/3}\left(v(\{1,2,3\})-v(\{2,3\})\right)}^{\{2,3\}} = \frac{1}{3}(3-2) = \frac{1}{3}$$

$$\varphi_1 = \frac{2}{3}-\frac{1}{6}-\frac{2}{6}+\frac{1}{3} = \frac{4-1-2+2}{6} = \frac{1}{2}$$

CAT 2:

$S \subseteq N\setminus\{2\}$: Subsets of N excluding cat 2:

$\{\}, \{1\}, \{3\}, \{1,3\}$

$$\overbrace{\underbrace{\frac{1(3-0-1)}{3!}}_{1/3}\left(v(\{2\})-v(\{\})\right)}^{\{\}} = \frac{1}{3}(4-0) = \frac{4}{3}$$

$$\overbrace{\underbrace{\frac{1(3-1-1)}{3!}}_{1/6}\left(v(\{1,2\}-v(\{1\})\right)}^{\{1\}} = \frac{1}{6}(3-2) = \frac{1}{6}$$

$$\overbrace{\underbrace{\frac{1(3-1-1)}{3!}}_{1/6}\left(v(\{2,3\}-v(\{3\})\right)}^{\{3\}} = \frac{1}{6}(2-5) = -\frac{3}{6}$$

$$\overbrace{\underbrace{\frac{2(3-2-1)}{3!}}_{1/3}\left(v(\{1,2,3\})-v(\{1,3\})\right)}^{\{1,3\}} = \frac{1}{3}(3-3) = 0$$

$$\varphi_2 = \frac{4}{3}+\frac{1}{6}-\frac{3}{6}+0 = \frac{8+1-3}{6} = 1$$

CAT 3:

$S \subseteq N\setminus\{3\}$: Subsets of N excluding cat 3:

$\{\}, \{1\}, \{2\}, \{1,2\}$

$$\overbrace{\underbrace{\frac{1(3-0-1)}{3!}}_{1/3}\left(v(\{3\})-v(\{\})\right)}^{\{\}} = \frac{1}{3}(5-0) = \frac{5}{3}$$

$$\overbrace{\underbrace{\frac{1(3-1-1)}{3!}}_{1/6}\left(v(\{1,3\}-v(\{1\})\right)}^{\{1\}} = \frac{1}{6}(3-2) = \frac{1}{6}$$

$$\overbrace{\underbrace{\frac{1(3-1-1)}{3!}}_{1/6}\left(v(\{2,3\}-v(\{2\})\right)}^{\{2\}} = \frac{1}{6}(2-4) = -\frac{2}{6}$$

$$\overbrace{\underbrace{\frac{2(3-2-1)}{3!}}_{1/3}\left(v(\{1,2,3\})-v(\{1,2\})\right)}^{\{1,3\}} = \frac{1}{3}(3-3) = 0$$

$$\varphi_3 = \frac{5}{3}+\frac{1}{6}-\frac{2}{6}+0 = \frac{10+1-2}{6} = \frac{3}{2}$$

Just do it

Computers out!

Coding time

Regarding that prefactor...

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

|S| depends on coalition ***sizes*** *(not members). N is constant.*

Coding time

Regarding that prefactor...

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

|S| denotes coalition **size** (independent of specific members).

N is constant.

$$\phi_i(v) = \frac{1}{N} \sum_{k=0}^{N-1} \frac{1}{|\mathcal{S}_k|} \sum_{S \in S_k} [v(S \cup \{i\}) - v(S)]$$

all sets of **size** k $|\mathcal{S}_k| = \binom{N-1}{k}$

Coding time

Task 1: What are the Shapley values for the three cats if the characteristic function values are as follows?

$$
\begin{aligned}
&v(\{\}) = 0 \quad \text{(no cats, no mice)} \\
&v(\{1\}) = 3, \quad v(\{2\}) = 3, \quad v(\{3\}) = 4 \\
&v(\{1,2\}) = 4, \quad v(\{1,3\}) = 4, \quad v(\{2,3\}) = 4 \\
&v(\{1,2,3\}) = 8
\end{aligned}
$$

Coding time

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

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

(for example)

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

Coding time

Task 1: What are the Shapley values for the three cats if the characteristic function values are as follows?

$$v(\{\}) = 0 \quad \text{(no cats, no mice)}$$
$$v(\{1\}) = 3, \quad v(\{2\}) = 3, \quad v(\{3\}) = 4$$
$$v(\{1,2\}) = 4, \quad v(\{1,3\}) = 4, \quad v(\{2,3\}) = 4$$
$$v(\{1,2,3\}) = 8$$

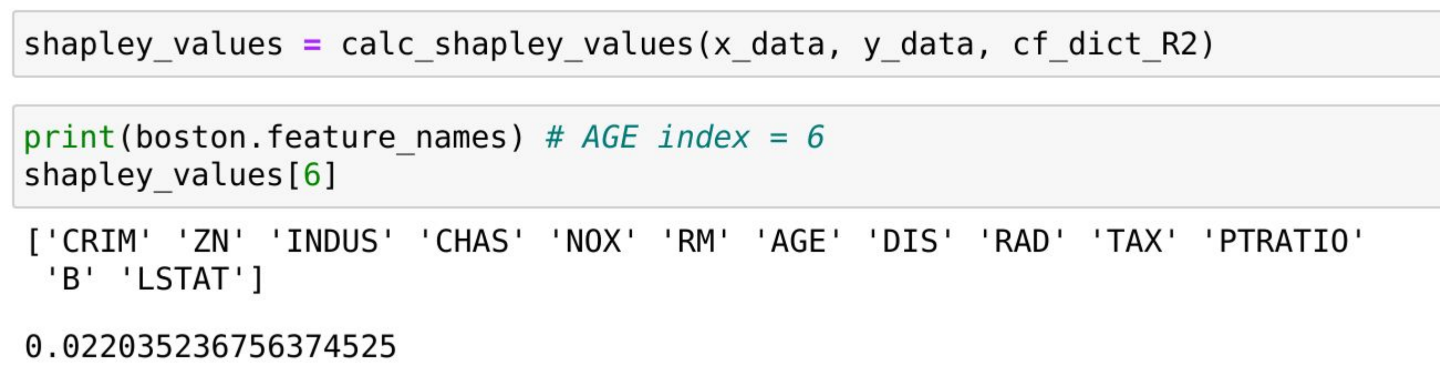
Task 2: Calculate the Shapley decomposition of Boston housing data set, using the coefficient of determination R^2 as characteristic function

Back to business!

- *Coffee / tea*
- *Stretch*
- *What is the Shapley value for **AGE**, on the Boston Housing data set?*

Back to business!

- *Coffee / tea*
- *Stretch*
- *What is the Shapley value for AGE?*



```
shapley_values = calc_shapley_values(x_data, y_data, cf_dict_R2)
```

```
print(boston.feature_names) # AGE index = 6
shapley_values[6]
```

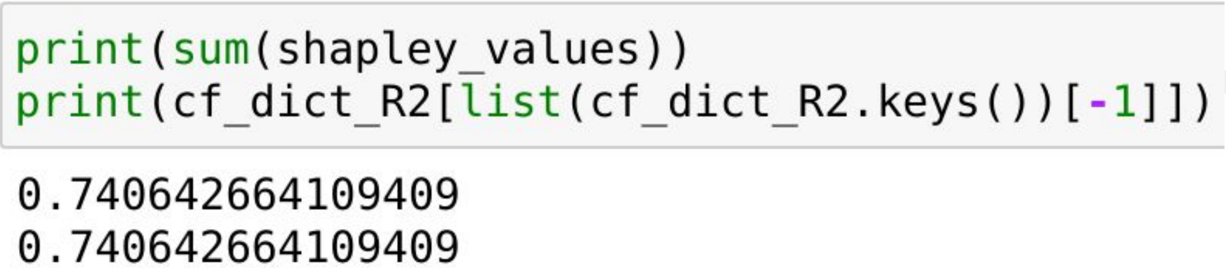
```
['CRIM' 'ZN' 'INDUS' 'CHAS' 'NOX' 'RM' 'AGE' 'DIS' 'RAD' 'TAX' 'PTRATIO'
 'B' 'LSTAT']

0.022035236756374525
```

The characteristic function

Shapley value or ***decomposition:*** *Distribution of the* ***value****, represented by the characteristic function*

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



```
print(sum(shapley_values))
print(cf_dict_R2[list(cf_dict_R2.keys())[-1]])
```

```
0.740642664109409
0.740642664109409
```

The characteristic function

Shapley value or **decomposition:** Distribution of the **value**, represented by the characteristic function

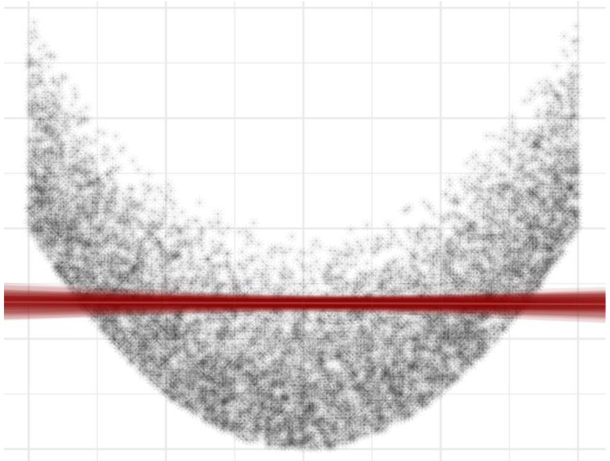
$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!}(v(S \cup \{i\}) - v(S))$$

aka "where the action is"

The characteristic function

R^2: Nice, quick to calculate, but linear

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



$Y=aX^2$

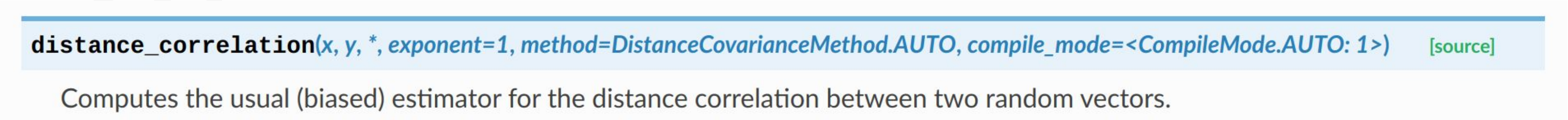
Estimate of $R^2=0.0043$ with 95% bootstrap CI (0.001, 0.013) over 100 fits.

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

Just do it

Characteristic functions out!

Non-linear correlation in the characteristic function



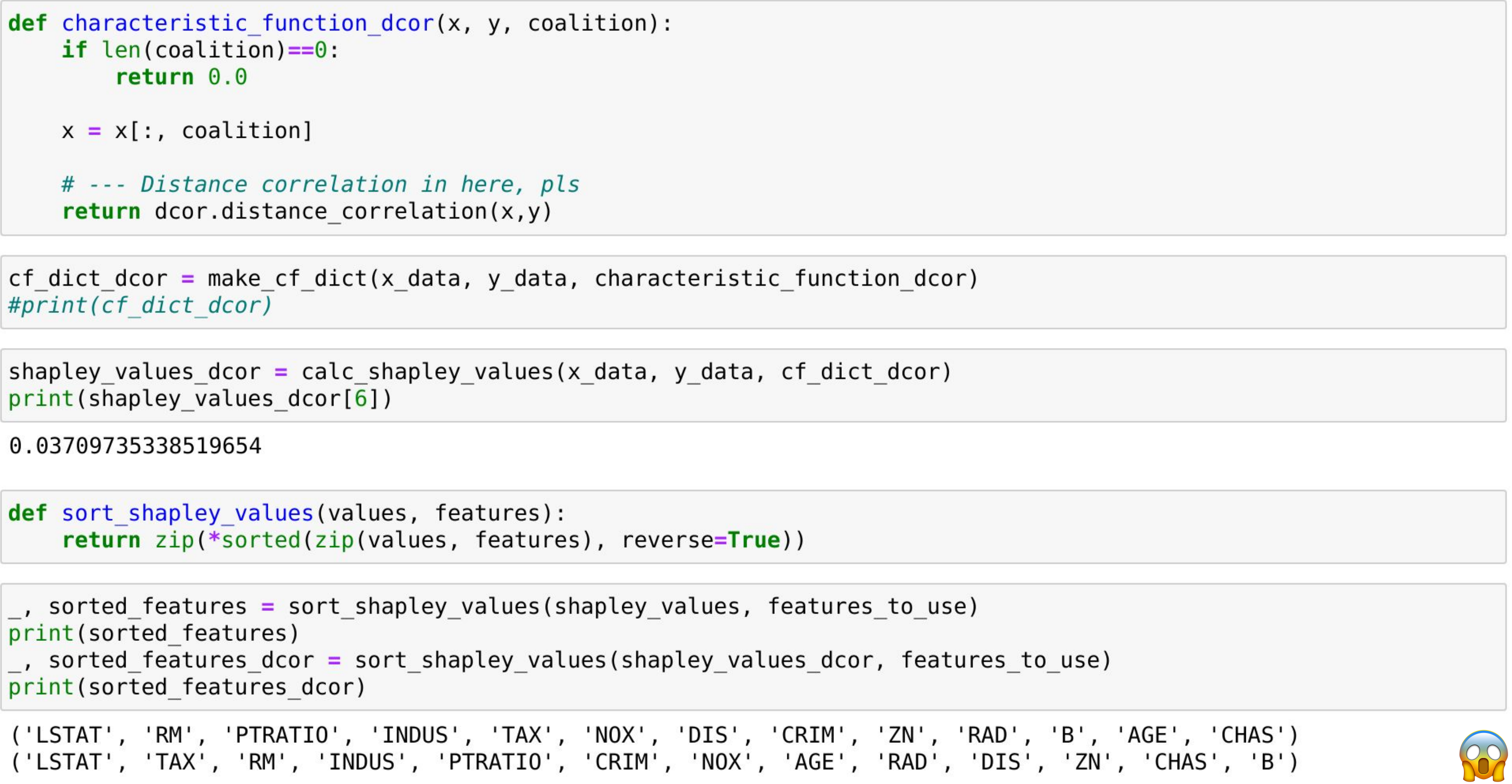
distance_correlation(*x, y, *, exponent=1, method=DistanceCovarianceMethod.AUTO, compile_mode=<CompileMode.AUTO: 1>*) [source]

Computes the usual (biased) estimator for the distance correlation between two random vectors.

Non-linear correlation in the characteristic function

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

Non-linear correlation in the characteristic function



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

    x = x[:, coalition]

    # --- Distance correlation in here, pls
    return dcor.distance_correlation(x,y)
```

```
cf_dict_dcor = make_cf_dict(x_data, y_data, characteristic_function_dcor)
#print(cf_dict_dcor)
```

```
shapley_values_dcor = calc_shapley_values(x_data, y_data, cf_dict_dcor)
print(shapley_values_dcor[6])
```

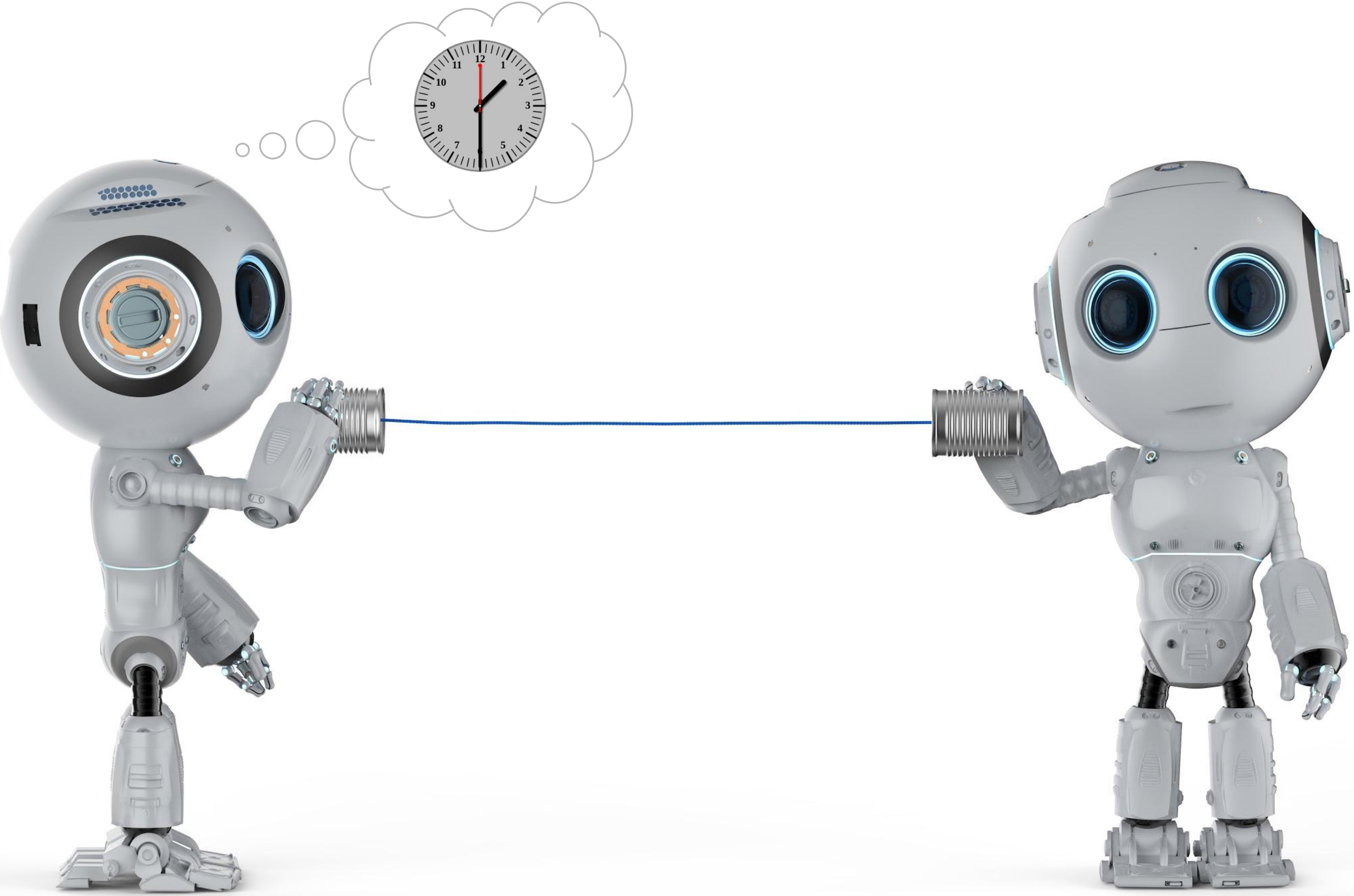
```
0.037097353385l9654
```

```
def sort_shapley_values(values, features):
    return zip(*sorted(zip(values, features), reverse=True))
```

```
_, sorted_features = sort_shapley_values(shapley_values, features_to_use)
print(sorted_features)
_, sorted_features_dcor = sort_shapley_values(shapley_values_dcor, features_to_use)
print(sorted_features_dcor)
```

```
('LSTAT', 'RM', 'PTRATIO', 'INDUS', 'TAX', 'NOX', 'DIS', 'CRIM', 'ZN', 'RAD', 'B', 'AGE', 'CHAS')
('LSTAT', 'TAX', 'RM', 'INDUS', 'PTRATIO', 'CRIM', 'NOX', 'AGE', 'RAD', 'DIS', 'ZN', 'CHAS', 'B')
```

😱


12
1
2
3
4
5
7
8
9
10
11

Just do it

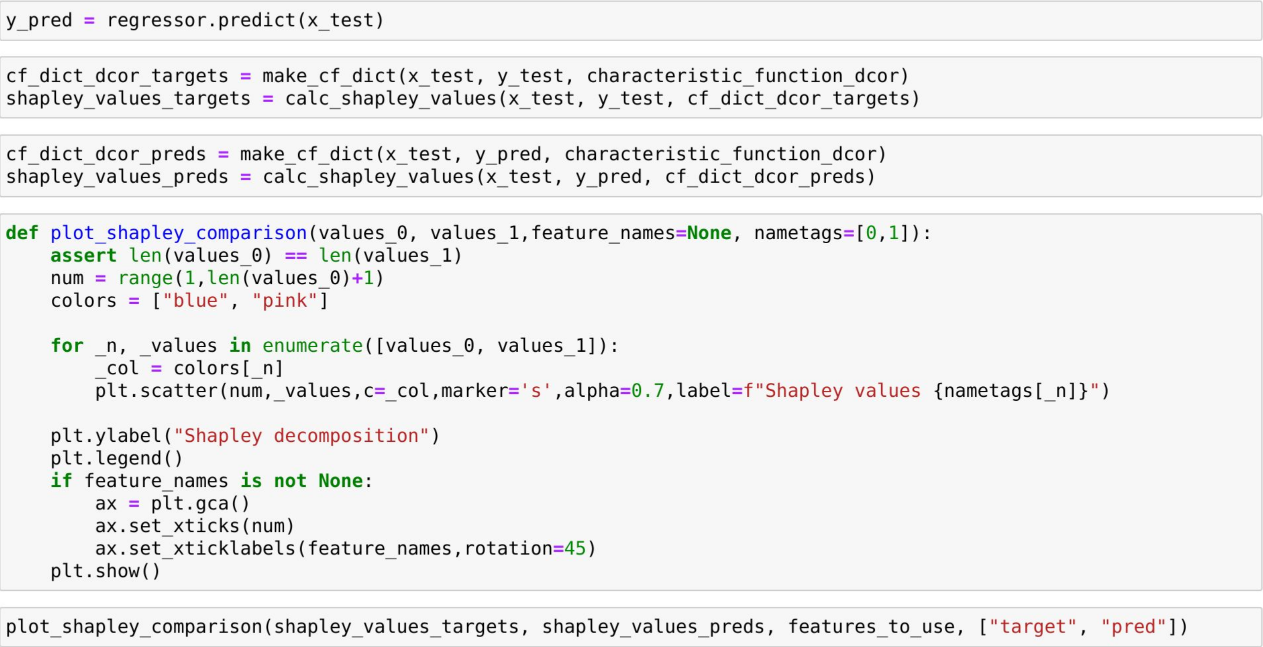
Models out!

Explaining a model, take 1

Machine learning model ✔️

Does it capture the **dependence structure in the data?**

Let's find out!



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

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

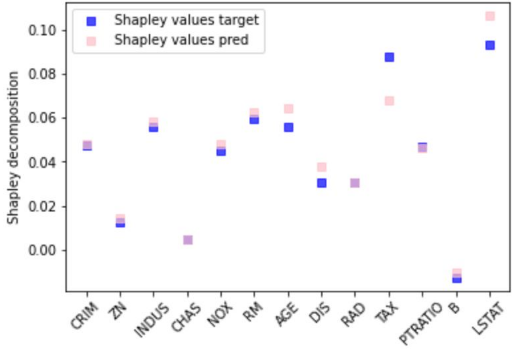
```
cf_dict_dcor_preds = make_cf_dict(x_test, y_pred, characteristic_function_dcor)
shapley_values_preds = calc_shapley_values(x_test, y_pred, cf_dict_dcor_preds)
```

```
def plot_shapley_comparison(values_0, values_1,feature_names=None, nametags=[0,1]):
    assert len(values_0) == len(values_1)
    num = range(1,len(values_0)+1)
    colors = ["blue", "pink"]

    for _n, _values in enumerate([values_0, values_1]):
        _col = colors[_n]
        plt.scatter(num,_values,c=_col,marker='s',alpha=0.7,label=f"Shapley values {nametags[_n]}")

    plt.ylabel("Shapley decomposition")
    plt.legend()
    if feature_names is not None:
        ax = plt.gca()
        ax.set_xticks(num)
        ax.set_xticklabels(feature_names,rotation=45)
    plt.show()
```

```
plot_shapley_comparison(shapley_values_targets, shapley_values_preds, features_to_use, ["target", "pred"])
```



Explaining the data or explaining a model? Shapley values that uncover non-linear dependencies

Daniel Vidali Fryer[*]
School of Mathematics and Physics,
The University of Queensland, St Lucia, Australia
Inga Strümke[†]
Simula Research Laboratory, Oslo, Norway
Hien Nguyen[‡]
Department of Mathematics and Statistics,
La Trobe University, Melbourne, Australia

Abstract

Shapley values have become increasingly popular in the machine learning literature thanks to their attractive axiomatisation, flexibility, and uniqueness in satisfying certain notions of 'fairness'. The flexibility arises from the myriad potential forms of the Shapley value *game formulation*. Amongst the consequences of this flexibility is that there are now many types of Shapley values being discussed, with such variety being a source of potential misunderstanding.

To the best of our knowledge, all existing game formulations in the machine learning and statistics literature fall into a category which we name the model-dependent category of game formulations. In this work, we consider an alternative and novel formulation which leads to the first instance of what we call model-independent Shapley values. These Shapley values use a (non-parametric) measure of non-linear dependence as the characteristic function. The strength of these Shapley values is in their ability to uncover and attribute non-linear dependencies amongst features.

We introduce and demonstrate the use of the energy distance correlations, affine-invariant distance correlation, and Hilbert-Shmidt independence criterion as Shapley value characteristic functions. In particular, we demonstrate their potential value for exploratory data analysis and model diagnostics. We conclude with an interesting expository application to a classical medical survey data set.

[*] daniel.fryer@uq.edu.au (Corresponding Author)

[†] inga@simula.no

[‡] h.nguyen5@latrobe.edu.au

Explaining the data or explaining a model? Shapley values that uncover non-linear dependencies

07/12/2020 · by Daniel Vidali Fryer, et al. · Simula Research Lab · La Trobe University · The University of Queensland · ♡ 52 · ➦ share

Shapley values have become increasingly popular in the machine learning literature thanks to their attractive axiomatisation, flexibility, and uniqueness in satisfying certain notions of `fairness'. The flexibility arises from the myriad potential forms of the Shapley value game formulation. Amongst the consequences of this flexibility is that there are now many types of Shapley values being discussed, with such variety being a source of potential misunderstanding. To the best of our knowledge, all existing game formulations in the machine learning and statistics literature fall into a category which we name the model-dependent category of game formulations. In this work, we consider an alternative and novel formulation which leads to the first instance of what we call model-independent Shapley values. These Shapley values use a (non-parametric) measure of non-linear dependence as the characteristic function. The strength of these Shapley values is in their ability to uncover and attribute non-linear dependencies amongst features. We introduce and demonstrate the use of the energy distance correlations, affine-invariant distance correlation, and Hilbert-Shmidt independence criterion as Shapley value characteristic functions. In particular, we demonstrate their potential value for exploratory data analysis and model diagnostics. We conclude with an interesting expository application to a classical medical survey data set.

Summary

Shapley values do dependence attribution. From game theory to ML

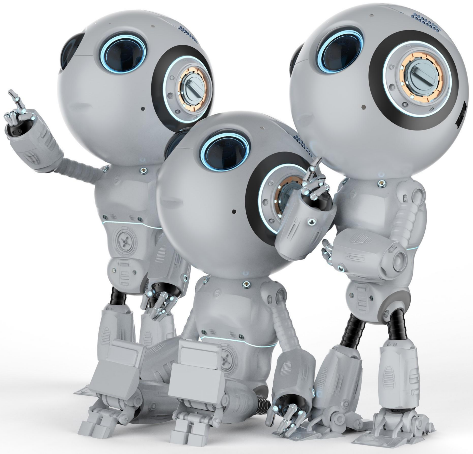
N all players --> features

i player --> feature

S coalition of players --> set of features

v game --> model

The Shapley value 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 $v(N)$



Explaining a model

How to calculate the marginal contributions, i.e. how do we remove features from our model?

Følg med i morgen...

Inga Strümke

inga@simula.no / inga@strumke.com