Necessity of explaining ML models and a choice of XAI - approaches for supervised learning

Dr. Benjamin Müller, Wiebke Hansen

House of Insurance

November 9th, 2023

Why explain ML models?

ML models:

Random forest

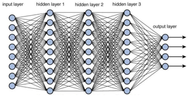
https://de.cleanpng.com/png-0tu3ea/

Why explain ML models?

ML models:

Random forest

https://de.cleanpng.com/png-0tu3ea/



Deep neural network

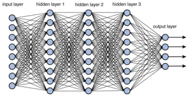
https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964

Why explain ML models?

ML models:

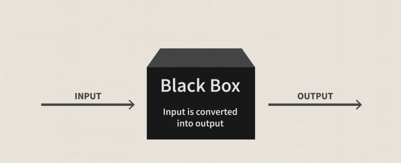
Random forest

https://de.cleanpng.com/png-0tu3ea/



Deep neural network

https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964



Blackbox model

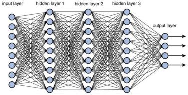
https://www.investopedia.com/terms/b/blackbox.asp

Why explain ML models?

ML models:

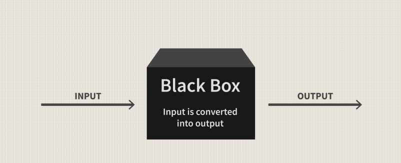
Random forest

https://de.cleanpng.com/png-0tu3ea/



Deep neural network

https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964



Blackbox model

https://www.investopedia.com/terms/b/blackbox.asp

Frequent criticism of ML models:

- "ML models are complex"
- "outcome of models is not understandable"

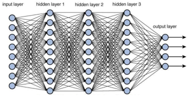
$\Rightarrow$ **intrinsic** motivation of explaining models

Why explain ML models?

ML models:

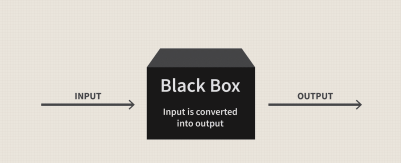
Random forest

https://de.cleanpng.com/png-0tu3ea/



Deep neural network

https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964



Blackbox model

https://www.investopedia.com/terms/b/blackbox.asp

Frequent criticism of ML models:

- "ML models are complex"
- "outcome of models is not understandable"

⇒ **intrinsic** motivation of explaining models

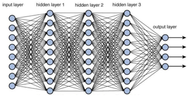
Articles/reports in literature:

Why explain ML models?

ML models:

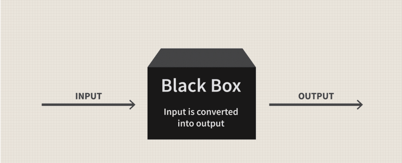
Random forest

https://de.cleanpng.com/png-0tu3ea/



Deep neural network

https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964



Blackbox model

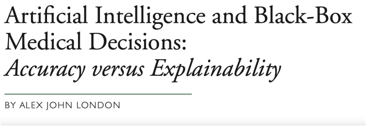
https://www.investopedia.com/terms/b/blackbox.asp

Frequent criticism of ML models:

- "ML models are complex"
- "outcome of models is not understandable"

⇒ **intrinsic** motivation of explaining models

Articles/reports in literature:


Artificial Intelligence and Black-Box Medical Decisions: *Accuracy versus Explainability*

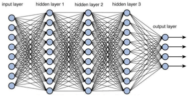
BY ALEX JOHN LONDON

Why explain ML models?

ML models:

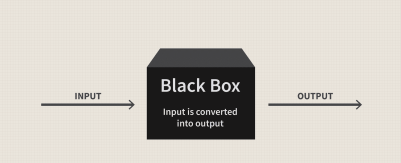
Random forest

https://de.cleanpng.com/png-0tu3ea/



Deep neural network

https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964



Blackbox model

https://www.investopedia.com/terms/b/blackbox.asp

Frequent criticism of ML models:

- "ML models are complex"
- "outcome of models is not understandable"

⇒ **intrinsic** motivation of explaining models

Articles/reports in literature:

Artificial Intelligence and Black-Box Medical Decisions:

Accuracy versus Explain

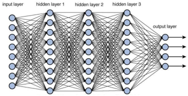
BY ALEX JOHN LONDON

Why explain ML models?

ML models:

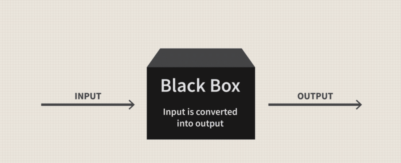
Random forest

https://de.cleanpng.com/png-0tu3ea/



Deep neural network

https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964



Blackbox model

https://www.investopedia.com/terms/b/blackbox.asp

Frequent criticism of ML models:

- "ML models are complex"
- "outcome of models is not understandable"

⇒ **intrinsic** motivation of explaining models

Articles/reports in literature:

Many considerations about explainable AI:

- prevent discrimination (cp. GDPR)
- regulation: **Artificial intelligence act**

⇒ **extrinsic** motivation of explaining models

What are explainable AI methods?

Purpose: Explaining AI models (e.g. Random Forest) and their output

"Why does the model predict what it predicts?"

What are explainable AI methods?

Purpose: Explaining AI models (e.g. Random Forest) and their output

"Why does the model predict what it predicts?"

Classification:

- **local vs. global**: explain the output of one **single** dataset or the output over **all** considered datasets

What are explainable AI methods?

Purpose: Explaining AI models (e.g. Random Forest) and their output

"Why does the model predict what it predicts?"

Classification:

- **local vs. global**: explain the output of one **single** dataset or the output over **all** considered datasets
- **model - agnostic vs. model - specific**: The explainability method is applicable to **all** ML methods respectively valid to a **single** type of model or a **group** of models.

What are explainable AI methods?

Purpose: Explaining AI models (e.g. Random Forest) and their output

"Why does the model predict what it predicts?"

Classification:

- **local vs. global**: explain the output of one **single** dataset or the output over **all** considered datasets
- **model - agnostic vs. model - specific**: The explainability method is applicable to **all** ML methods respectively valid to a **single** type of model or a **group** of models.

Selection of popular methods:

	model - agnostic	**model - specific**
global	Partial Dependence Plot (short: PDP)	Feature Importance for DecisionTreeRegressor (scikit - learn)
local	SHAP	...

Toy problem

Description of the problem:

- business: insurance
- Type of problem: supervised regression
- Underlying data set derived by `SwedishMotorInsurance`[a], 1.797 rows, 5 columns
- Features (all categorical):

Feature	**# distinct values**
Kilometres	5
Zone	7
Bonus	7
Make	9

[a] https://www.kaggle.com/code/ashwin8699/swedish-motor-insurance-simple-linear-regression/input

Toy problem

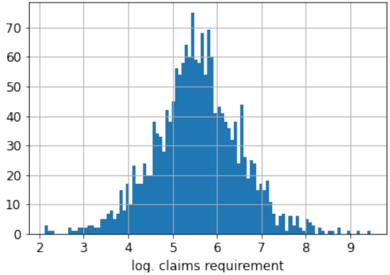
Description of the problem:

- business: insurance
- Type of problem: supervised regression
- Underlying data set derived by `SwedishMotorInsurance`[a], 1.797 rows, 5 columns
- Features (all categorical):

Feature	# distinct values
Kilometres	5
Zone	7
Bonus	7
Make	9

Target: "claims requirement"

$$\text{Claim requirement} = \frac{\text{Claim costs}}{\text{exposure}}$$



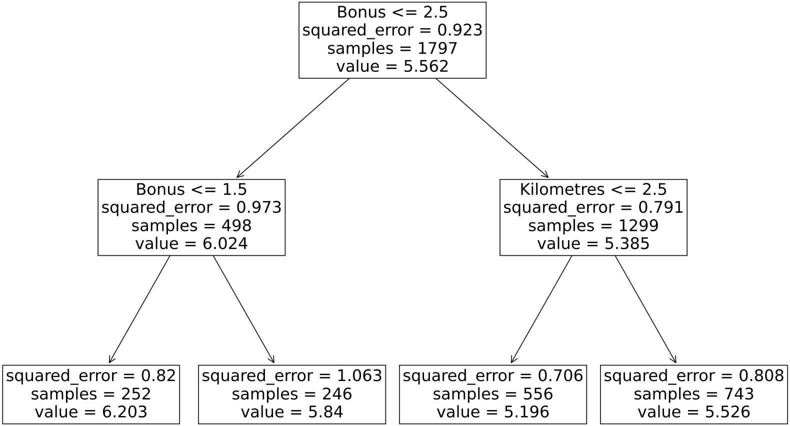
[a] https://www.kaggle.com/code/ashwin8699/swedish-motor-insurance-simple-linear-regression/input

Simple model for toy problem

DecisionTreeRegressor from scikit - learn (deepness: 2)

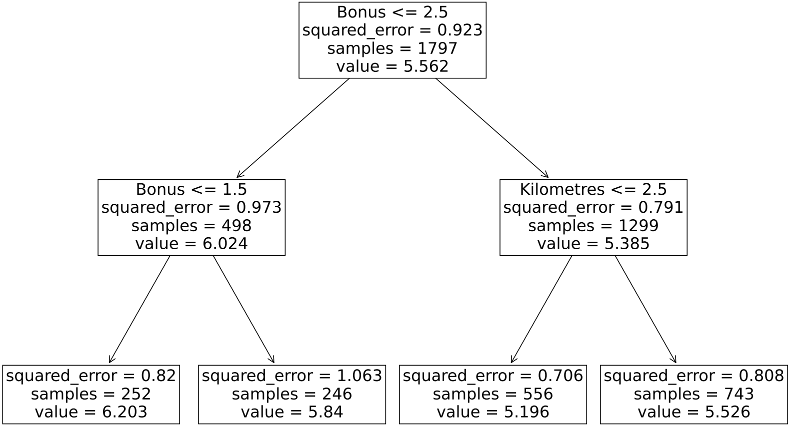
Simple model for toy problem

DecisionTreeRegressor from scikit - learn (deepness: 2)



Simple model for toy problem

DecisionTreeRegressor from scikit - learn (deepness: 2)



Result:

The decision tree depends only on the features “Bonus” and “Kilometres”.

Partial Dependence Plot

Implementation in scikit - learn:

`PartialDependenceDisplay` from `sklearn.inspection`

Building a pdp for a given model:

1. Select the feature for that you want to plot a PDP and determine the different values (= levels).
2. Iterate over the different levels:
   a) Change the dataset in the selected feature column to the fixed level.
   b) Predict the outcome for this dataset.
   c) The average of the predictions is the pdp value for the fixed level.

Partial Dependence Plot

Implementation in scikit - learn:

`PartialDependenceDisplay` from `sklearn.inspection`

Building a pdp for a given model:

1. Select the feature for that you want to plot a PDP and determine the different values (= levels).
2. Iterate over the different levels:
   a) Change the dataset in the selected feature column to the fixed level.
   b) Predict the outcome for this dataset.
   c) The average of the predictions is the pdp value for the fixed level.

Initial data set:

$$X = \begin{bmatrix} \text{Kilom.} & \text{Zone} & \text{Bonus} & \text{Make} \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 3 & 1 & 3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 5 & 7 & 7 & 9 \end{bmatrix} \Rightarrow \hat{y} = \begin{pmatrix} \text{pred.} \\ 6.20296 \\ \vdots \\ 5.52571 \\ \vdots \\ 5.52571 \end{pmatrix} \begin{matrix} \text{index} \\ 0 \\ \vdots \\ 797 \\ \vdots \\ 1796 \end{matrix}$$

$$\Rightarrow mean(\hat{y}) = 5.56175$$

Partial Dependence Plot

Implementation in scikit - learn:

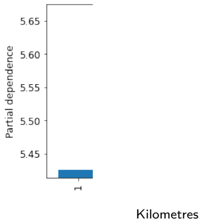
`PartialDependenceDisplay` from `sklearn.inspection`

Building a pdp for a given model:

1. Select the feature for that you want to plot a PDP and determine the different values (= levels).
2. Iterate over the different levels:
   a) Change the dataset in the selected feature column to the fixed level.
   b) Predict the outcome for this dataset.
   c) The average of the predictions is the pdp value for the fixed level.

Data set with *Kilometres* = 1:

$$X = \begin{array}{c} \begin{array}{cccc} \text{Kilom.} & \text{Zone} & \text{Bonus} & \text{Make} \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 7 & 7 & 9 \end{bmatrix} \end{array} \Rightarrow \hat{y} = \begin{array}{c} \text{pred.} \\ \begin{pmatrix} 6.20296 \\ \vdots \\ 5.19614 \\ \vdots \\ 5.19614 \end{pmatrix} \end{array} \begin{array}{c} \text{index} \\ 0 \\ \vdots \\ 797 \\ \vdots \\ 1796 \end{array}$$



$$\Rightarrow mean(\hat{y}) = 5.42549$$

Partial Dependence Plot

Implementation in scikit - learn:

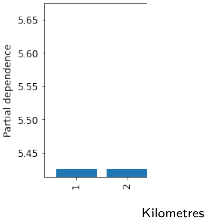
`PartialDependenceDisplay` from `sklearn.inspection`

Building a pdp for a given model:

1. Select the feature for that you want to plot a PDP and determine the different values (= levels).
2. Iterate over the different levels:
   a) Change the dataset in the selected feature column to the fixed level.
   b) Predict the outcome for this dataset.
   c) The average of the predictions is the pdp value for the fixed level.

Data set with *Kilometres* = 2:

$$X = \begin{bmatrix} \text{Kilom.} & \text{Zone} & \text{Bonus} & \text{Make} \\ 2 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2 & 1 & 3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2 & 7 & 7 & 9 \end{bmatrix} \Rightarrow \hat{y} = \begin{pmatrix} \text{pred.} \\ 6.20296 \\ \vdots \\ 5.19614 \\ \vdots \\ 5.19614 \end{pmatrix} \begin{matrix} \text{index} \\ 0 \\ \vdots \\ 797 \\ \vdots \\ 1796 \end{matrix}$$



$\Rightarrow mean(\hat{y}) = 5.42549$

Partial Dependence Plot

Implementation in scikit-learn:

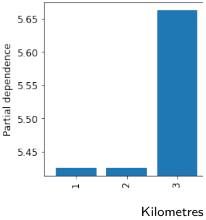
`PartialDependenceDisplay` from `sklearn.inspection`

Building a pdp for a given model:

1. Select the feature for that you want to plot a PDP and determine the different values (= levels).
2. Iterate over the different levels:
   a) Change the dataset in the selected feature column to the fixed level.
   b) Predict the outcome for this dataset.
   c) The average of the predictions is the pdp value for the fixed level.

Data set with *Kilometres* = 3:

$$X = \begin{array}{c} \text{Kilom.} \quad \text{Zone} \quad \text{Bonus} \quad \text{Make} \\ \begin{bmatrix} 3 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 3 & 1 & 3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 3 & 7 & 7 & 9 \end{bmatrix} \end{array} \Rightarrow \hat{y} = \begin{array}{cc} \text{pred.} & \text{index} \\ \begin{pmatrix} 6.20296 \\ \vdots \\ 5.52571 \\ \vdots \\ 5.52571 \end{pmatrix} & \begin{array}{c} 0 \\ \vdots \\ 797 \\ \vdots \\ 1796 \end{array} \end{array}$$



$$\Rightarrow mean(\hat{y}) = 5.66372$$

Partial Dependence Plot

<u>Implementation in scikit-learn:</u>

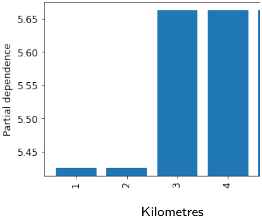
`PartialDependenceDisplay` from `sklearn.inspection`

<u>Building a pdp for a given model</u>:

1. Select the feature for that you want to plot a PDP and determine the different values (= levels).
2. Iterate over the different levels:
   a) Change the dataset in the selected feature column to the fixed level.
   b) Predict the outcome for this dataset.
   c) The average of the predictions is the pdp value for the fixed level.

<u>Data set with *Kilometres* = 4:</u>

$$X = \begin{bmatrix} \text{Kilom.} & \text{Zone} & \text{Bonus} & \text{Make} \\ \color{blue}{4} & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \color{blue}{4} & 1 & 3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \color{blue}{4} & 7 & 7 & 9 \end{bmatrix} \Rightarrow \hat{y} = \begin{pmatrix} \text{pred.} \\ 6.20296 \\ \vdots \\ 5.52571 \\ \vdots \\ 5.52571 \end{pmatrix} \begin{matrix} \text{index} \\ 0 \\ \vdots \\ 797 \\ \vdots \\ 1796 \end{matrix}$$



$$\Rightarrow mean(\hat{y}) = 5.66372$$

Partial Dependence Plot

Implementation in scikit - learn:

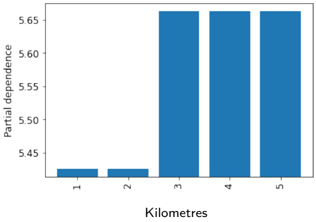
`PartialDependenceDisplay` from `sklearn.inspection`

Building a pdp for a given model:

1. Select the feature for that you want to plot a PDP and determine the different values (= levels).
2. Iterate over the different levels:
   a) Change the dataset in the selected feature column to the fixed level.
   b) Predict the outcome for this dataset.
   c) The average of the predictions is the pdp value for the fixed level.

Data set with *Kilometres* = 5:

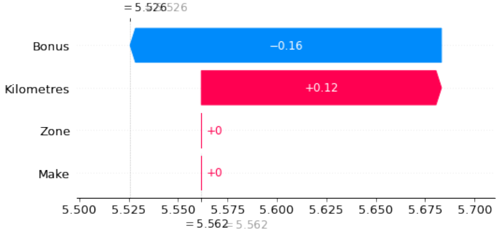
$$X = \begin{bmatrix} \text{Kilom.} & \text{Zone} & \text{Bonus} & \text{Make} \\ 5 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 5 & 1 & 3 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 5 & 7 & 7 & 9 \end{bmatrix} \Rightarrow \hat{y} = \begin{pmatrix} \text{pred.} & \text{index} \\ 6.20296 & 0 \\ \vdots & \vdots \\ 5.52571 & 797 \\ \vdots & \vdots \\ 5.52571 & 1796 \end{pmatrix}$$



$\Rightarrow mean(\hat{y}) = 5.66372$

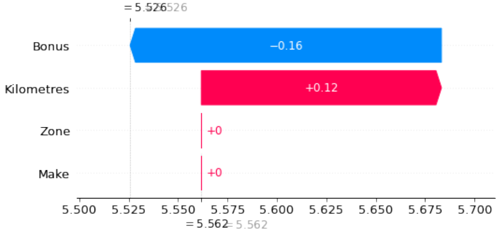
Shapley values

How did each feature contribute to an individual result?



Shapley values

How did each feature contribute to an individual result?



The individual result (5.526) deviates from the observed mean (5.562).

$\Longrightarrow$ As expected, *Zone* and *Make* do not have any impact on the result.

$\Longrightarrow$ *Bonus* reduces the result by ≈ 0.16.

$\Longrightarrow$ *Kilometres* causes a positive shift of ≈ 0.12.

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

E.g.: Instance 797 *has Bonus* = 3.

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

$$E.g.: Instance\ 797\ has\ Bonus = 3.$$

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S) \right)$$

2. Compute the weighted average

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

$$E.g.: Instance\ 797\ has\ Bonus = 3.$$

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S) \right)$$

2. Compute the weighted average

$$\phi_j(x) = \sum_{S \subseteq \{F_1, \dots, F_p\} \setminus \{F_j\}} \frac{|S|!(p - |S| - 1)!}{p!} \cdot mc(x, F_j, S)$$

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

E.g.: Instance 797 *has Bonus* = 3.

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S)\right)$$

2. Compute the weighted average

$$\phi_j(x) = \sum_{S \subseteq \{F_1,\dots,F_p\} \setminus \{F_j\}} \frac{|S|!(p - |S| - 1)!}{p!} \cdot mc(x, F_j, S)$$

Calculation of marginal contribution:

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

1. Choose random instance z of the data set
2. Create x_{-} with values x on set S and values from z for the other features
3. Create x_{+} with values x on set $S \cup \{F_j\}$ and values from z for the other features
4. Calculate $mc^m(x, F_j, S) := \hat{f}(x_{+}) - \hat{f}(x_{-})$

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

$$E.g.: Instance\ 797\ has\ Bonus = 3.$$

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S) \right)$$

2. Compute the weighted average

$$\phi_j(x) = \sum_{S \subseteq \{F_1, \ldots, F_p\} \setminus \{F_j\}} \frac{|S|!(p - |S| - 1)!}{p!} \cdot mc(x, F_j, S)$$

Calculation of marginal contribution:

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

1. Choose random instance z of the data set
2. Create x_- with values x on set S and values from z for the other features
3. Create x_+ with values x on set $S \cup \{F_j\}$ and values from z for the other features
4. Calculate $mc^m(x, F_j, S) := \hat{f}(x_+) - \hat{f}(x_-)$

Set **marginal contribution** as $mc(x, F_j, S) \approx \frac{1}{M} \sum_{m=1}^{M} mc^m(x, F_j, S)$.

Example for $S = \{Kilometres\}$ and fixed m:

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

E.g.: Instance 797 *has Bonus* = 3.

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S)\right)$$

2. Compute the weighted average

$$\phi_j(x) = \sum_{S \subseteq \{F_1,\dots,F_p\} \setminus \{F_j\}} \frac{|S|!(p - |S| - 1)!}{p!} \cdot mc(x, F_j, S)$$

Calculation of marginal contribution:

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

1. Choose random instance z of the data set
2. Create x_- with values x on set S and values from z for the other features
3. Create x_+ with values x on set $S \cup \{F_j\}$ and values from z for the other features
4. Calculate $mc^m(x, F_j, S) := \hat{f}(x_+) - \hat{f}(x_-)$

Set **marginal contribution** as $mc(x, F_j, S) \approx \frac{1}{M} \sum_{m=1}^{M} mc^m(x, F_j, S)$.

Example for $S = \{Kilometres\}$ and fixed m:

```
S =  ['Kilometres']
Feature of interest: Bonus


Instance of interest (index=797):
    Kilometres Zone Bonus Make
797          3    1     3    1


Random instance (e.g. index=194):
    Kilometres Zone Bonus Make
194          1    4     2    2
```

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

E.g.: Instance 797 *has Bonus* = 3.

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S) \right)$$

2. Compute the weighted average

$$\phi_j(x) = \sum_{S \subseteq \{F_1, \ldots, F_p\} \setminus \{F_j\}} \frac{|S|!(p - |S| - 1)!}{p!} \cdot mc(x, F_j, S)$$

Calculation of marginal contribution:

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

1. Choose random instance z of the data set
2. Create x_- with values x on set S and values from z for the other features
3. Create x_+ with values x on set $S \cup \{F_j\}$ and values from z for the other features
4. Calculate $mc^m(x, F_j, S) := \hat{f}(x_+) - \hat{f}(x_-)$

Set **marginal contribution** as $mc(x, F_j, S) \approx \frac{1}{M} \sum_{m=1}^{M} mc^m(x, F_j, S)$.

Example for $S = \{Kilometres\}$ and fixed m:

```
S =  ['Kilometres']
Feature of interest: Bonus

Instance of interest (index=797):
     Kilometres Zone Bonus Make
797           3    1     3    1

Random instance (e.g. index=194):
     Kilometres Zone Bonus Make
194           1    4     2    2
```

```
x_minus:
     Kilometres Zone Bonus Make
194           3    4     2    2

x_plus:
     Kilometres Zone Bonus Make
194           3    4     3    2
```

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

E.g.: Instance 797 *has Bonus* = 3.

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S)\right)$$

2. Compute the weighted average

$$\phi_j(x) = \sum_{S \subseteq \{F_1,\dots,F_p\} \setminus \{F_j\}} \frac{|S|!(p - |S| - 1)!}{p!} \cdot mc(x, F_j, S)$$

Calculation of marginal contribution:

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

1. Choose random instance z of the data set
2. Create x_- with values x on set S and values from z for the other features
3. Create x_+ with values x on set $S \cup \{F_j\}$ and values from z for the other features
4. Calculate $mc^m(x, F_j, S) := \hat{f}(x_+) - \hat{f}(x_-)$

Set **marginal contribution** as $mc(x, F_j, S) \approx \frac{1}{M}\sum_{m=1}^{M} mc^m(x, F_j, S)$.

Example for $S = \{Kilometres\}$ and fixed m:

```
S =  ['Kilometres']
Feature of interest: Bonus

Instance of interest (index=797):
    Kilometres  Zone  Bonus  Make
797          3     1      3     1

Random instance (e.g. index=194):
    Kilometres  Zone  Bonus  Make
194          1     4      2     2
```

```
x_minus:
    Kilometres  Zone  Bonus  Make
194          3     4      2     2

x_plus:
    Kilometres  Zone  Bonus  Make
194          3     4      3     2
```

Shapley values: explanation

Aim:

Compute shapley value for fixed instance x (e.g. *sample 797*) and fixed feature F_j (e.g. *Bonus*), given model and data set with p features

General idea:

How does the prediction change if we add the information of the instance for F_j to different feature combinations?

E.g.: Instance 797 *has Bonus* = 3.

We can add the feature "Bonus" to the following feature combinations S:

- $S = \emptyset$
- $S = \{Kilometres\}$
- $S = \{Make\}$
- $S = \{Zone\}$
- $S = \{Kilometres, Make\}$
- $S = \{Kilometres, Zone\}$
- $S = \{Make, Zone\}$
- $S = \{Kilometres, Make, Zone\}$

1. Compare the performance of each S with and without F_j (**marginal contribution**)

$$mc(x, F_j, S) := \left(val_x(S \cup \{F_j\}) - val_x(S)\right)$$

2. Compute the weighted average

$$\phi_j(x) = \sum_{S \subseteq \{F_1,\dots,F_p\} \setminus \{F_j\}} \frac{|S|!(p - |S| - 1)!}{p!} \cdot mc(x, F_j, S)$$

Calculation of marginal contribution:

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

1. Choose random instance z of the data set
2. Create x_- with values x on set S and values from z for the other features
3. Create x_+ with values x on set $S \cup \{F_j\}$ and values from z for the other features
4. Calculate $mc^m(x, F_j, S) := \hat{f}(x_+) - \hat{f}(x_-)$

Set **marginal contribution** as $mc(x, F_j, S) \approx \frac{1}{M} \sum_{m=1}^{M} mc^m(x, F_j, S)$.

Example for $S = \{Kilometres\}$ and fixed m:

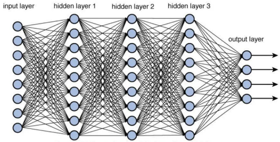
```
S =  ['Kilometres']
Feature of interest: Bonus

Instance of interest (index=797):
    Kilometres Zone Bonus Make
797          3    1     3    1

Random instance (e.g. index=194):
    Kilometres Zone Bonus Make
194          1    4     2    2
```

```
x_minus:
    Kilometres Zone Bonus Make
194          3    4     2    2

x_plus:
    Kilometres Zone Bonus Make
194          3    4     3    2

Prediction of x_plus:
[5.52570936]
Prediction of x_minus:
[5.84011094]
marginal contribution:
[-0.31440159]
```

Summary and research interests

Intrinsic/personal motivation

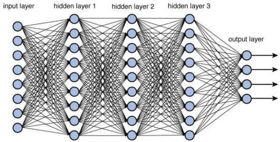
We want to understand
(complex) ML models



Summary and research interests

Intrinsic/personal motivation

We want to understand (complex) ML models



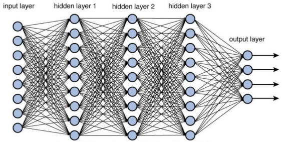
Extrinsic motivation

We **have** to explain ML models

Summary and research interests

<u>Intrinsic/personal motivation</u>

We want to understand (complex) ML models



<u>Extrinsic motivation</u>

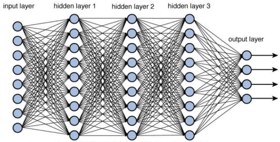
We **have** to explain ML models

$\Longrightarrow$ Increasing future relevance of XAI

Summary and research interests

<u>Intrinsic/personal motivation</u>

We want to understand (complex) ML models



<u>Extrinsic motivation</u>

We **have** to explain ML models

$\Longrightarrow$ Increasing future relevance of XAI

But also: Understand explanation methods.

Don't explain a black box with a black box.

Literature

- A. J. London: Artificial Intelligence and Black-Box Medical Decisions: Accuracy versus Explainability (https://www.cmu.edu/dietrich/philosophy/docs/london/hastings.pdf)
- Bias in Algorithms - Artificial Intelligence and Discrimination (https://fra.europa.eu/sites/default/files/fra_uploads/fra-2022-bias-in-algorithms_en.pdf)
- The Artificial Intelligence Act (https://artificialintelligenceact.eu)
- C. Molnar: Interpretable Machine Learning (https://christophm.github.io/interpretable-ml-book/)
- Python packages:
  - scikit - learn (https://scikit-learn.org/stable/index.html)
  - shap (https://shap.readthedocs.io/en/latest/index.html)

Talk on **November 21st, 2023** at DAV/DGVFM autumn meeting in Hanover:

14:45 - 15:30 Uhr	**Erklärbare Künstliche Intelligenz: Eine Diskussion für Aktuarinnen und Aktuare** *Prof. Dr. Anja Bettina Schmiedt (TH Rosenheim), Dr. Simon Hatzesberger (Allianz), Dr. Benjamin Müller (HDI)*

Thank you for your attention