Making machine learning techniques interpretable for predictive modelling

Xavier Maréchal, Reacfin

About the speaker

- **Xavier Maréchal** – *CEO Reacfin*
  - MSc. in Engineering Science (Applied Mathematics), MSc. in Actuarial Sciences and MSc. in Business Management
  - Co-author of "Actuarial Modelling of Claim Counts: Risk classification, Credibility and Bonus-Malus Systems"
  - Consultant for 16 years in Non-Life (Pricing, DFA models, Solvency 2)

- **Reacfin s.a.** is a **consulting firm**, spin-off of the University of Louvain (Louvain-la-Neuve - Belgium)

We develop, in partnership with our clients, **actuarial** & quantitative **financial** solutions, building on strong **data analytics**, while securing full transparency and integral knowledge transfer.

Reacfin Consulting

We offer consulting services in actuarial science & quantitative finance, including a.o. capital, portfolio, product, risk and liquidity management. We build our expertise on broad data analytics capacities.

Reacfin Know-How to Risk

Reacfin Tooling

We develop solutions in partnership with our clients, i.e. we integrate our solutions in our client's systems and processes and we secure full knowledge transfer (e.g. open source code).

Reacfin Academy

We share our knowledge with our clients. We offer a comprehensive learning platform, including on-site trainings, e-learning modules, e-classrooms and webinars.

Goals of this presentation

The problem

- Whereas advanced Machine learning techniques (e.g. random forest or neural networks) usually have a better predictive power than statistical techniques (e.g. GLM), their main drawback is that they are black-box and **their results are difficult to understand/interpret.**

Two different strategies to use ML for practical applications

- There are basically 2 strategies to use ML techniques in predictive modelling
  1. **Replacing** traditional models (e.g. GLM) by ML models
  2. **Combining** the pros of traditional and ML models to improve predictive modelling
- The goals of this presentation are therefore to
  - Briefly **remind some useful machine learning techniques** and explain why it is difficult to interpret their results
  - Present several techniques that have been developed in order to **better understand the results** of machine learning techniques
  - Explain how these **interpretation techniques** can be used to implement the 2 strategies presented above and improve predictive modelling

Agenda

- 1. A non-exhaustive reminder to some useful ML techniques
- 2. Adding complexity means increasing need for interpretability
- 3. An introduction to ML interpretation tools
- 4. Conclusions: how to make the most of ML techniques

What is machine learning?

Objectives of Machine Learning ("ML")

ML algorithms aim at finding by themselves the method that best predicts the outcome of the studied phenomenon.

Supervised vs. Unsupervised learning

- **Supervised learning:**
  - Inputs and examples of their desired outputs are provided
  - The goal is to learn a **general rule that maps inputs to outputs**.

➔ *Given a set of training examples (x_1, x_2,…, x_n, y), where y is the variable to be predicted , what is the most efficient algorithm to best approximate the realizations of y*

  - 2 main techniques
    - **Classification** : inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one (or multi-label classification) or more of these classes.
    - **Regression**: the outputs are continuous rather than discrete.

- **Unsupervised learning:**
  - No labels are given to the learning algorithm
  - The goal is to **find structure in its input** (discovering hidden patterns in data).
  - Main technique
    - **Clustering**: a set of inputs is to be divided into groups. Unlike in classification, the groups may not be known beforehand.

For a more complete presentation of some supervised models, check Reacfin webinar on "Machine learning applications to non-life pricing" https://www.reacfin.com/index.php/webinars/

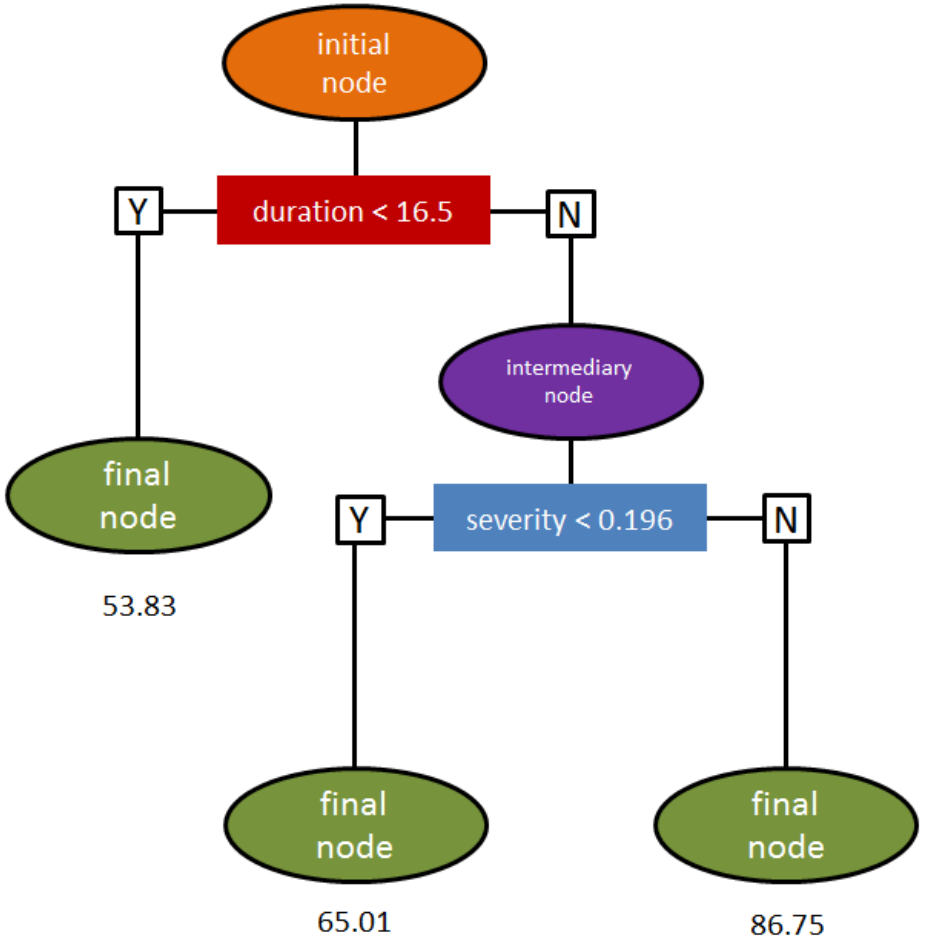
Examples of use

- Typically used to model **pricing or underwriting** related **target variables** in function of available **features**
  - Regression: frequency (#claims) or severity (claims cost)
  - Classification: lapse rates, conversion rates

- Typically used for **features engineering** (i.e. creating new variables)
  - E.g. vehicle classification, zoning,…

Focus on supervised models

A first simple ML model: Classification and regression trees (CART)



Purpose

- Tree enables to **segment the predictor space** into a number of simple homogenous regions defined according to the covariates
- **Splitting rules** can be summarized in a tree view
- For each region the prediction is set as the **region average**

Definitions

- The *root node* in orange:
  - at the top of the tree
  - contains the whole population
- The splitting rules set aim at segmenting the predictor space into a number of **simple regions** that are as **homogeneous** as possible with respect to the response variable
- The *leaves nodes* in green at the bottom of the tree: that is a node that is not further split.

An example of a more complex ML model: Bootstrap aggregation (Bagging)

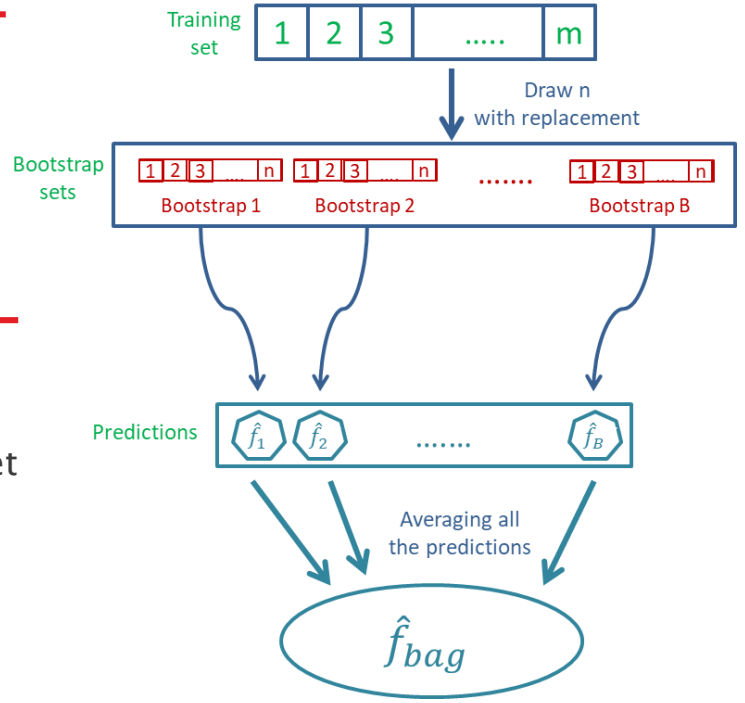
Main idea

- **B**ootstrap **agg**regation, or **Bagg**ing, is a general-purpose procedure for reducing the variance of a statistical learning method
- Recall that given a set of n independent observations $Z_1, Z_2, \ldots, Z_n$ each with variance σ^2, the variance of the mean $\bar{Z}$ of the observations is given by $\frac{\sigma^2}{n}$.
- **Averaging a set of observations reduces variance**.

Algorithm

1. Bootstrap, by taking **repeated samples** from the training data set.
2. Generate B different training data sets.
3. **Train our method** (e.g. regression tree) on the bth bootstrapped training set in order to get $\hat{f}_b(x)$ the prediction at point x.
4. We then **average all the predictions** to obtain:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x)$$



➡ The final prediction is **difficult to understand** as it is an **average** of a large number of "intermediate" predictions

➡ Similar difficulty in interpreting results is also an issue for other widely used ML models (**Random Forests, Gradient Boosting Models, Artificial Neural Networks**,….)

Agenda

- 1. A non-exhaustive reminder to some useful ML techniques
- 2. Adding complexity means increasing need for interpretability
- 3. An introduction to ML interpretation tools
- 4. Conclusions: how to make the most of ML techniques

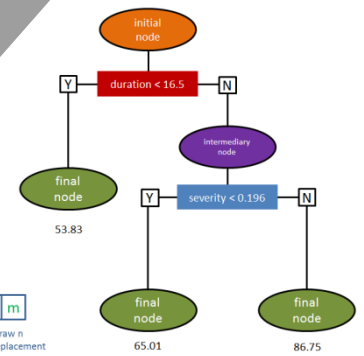
Some machine learning techniques are black boxes and interpretation of the results can be quite difficult

Increasing complexity to boost predictive power often means decreasing the interpretability of the results

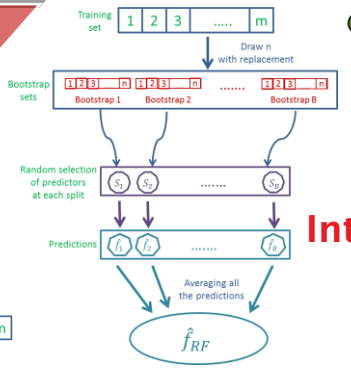
Complexity

Interpretability

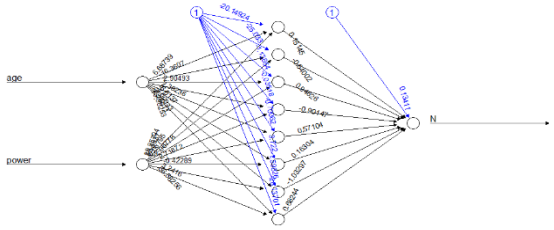
In the case of regression trees, understanding how the model predicts the target variable for new data points is not a problem, as it is very intuitive.

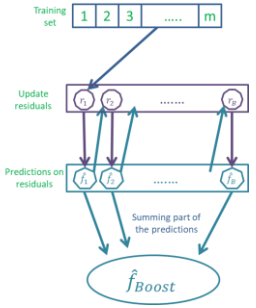


In the case of more complex methods such as Bagging and Random forests, even understanding how the model predicts values for new data points is rather difficult.



Things may be even worse for GBM and NN.





Understanding the results of ML models is nevertheless key for sound business decision-making as many stakeholders use the results of the models

Quant (Actuaries, data scientist,…)

- Are able to understand the technical details
- Trust its outputs based on cross-validation, error measures and assesment plots

⟷

Other stakeholders

- Not necessarily « quantitative people »
- Should nevertheless understand and trust results to take decisions

Machine learning techniques usually improve predictive power but at the expense of a certain loss of interpretability ➔ Find trade-off between

- Predictive power
- Capacity to understand the results
- Ability to take sound decisions based on the results

High-end questions

Who will use the results? For what purpose? With which impact?

Agenda

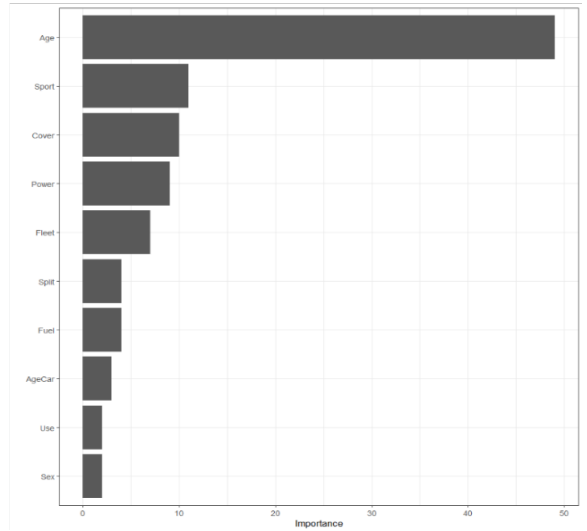
- 1. A non-exhaustive reminder to some useful ML techniques
- 2. Adding complexity means increasing need for interpretability
- 3. An introduction to ML interpretation tools
- 4. Conclusions: how to make the most of ML techniques

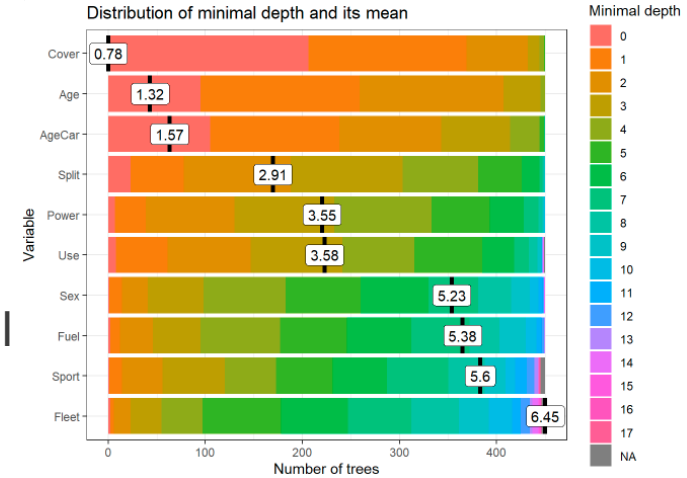
Global vs local Interpretability of ML techniques

- **Global Model Interpretability**
  - How does the trained model make predictions?
    - Which features are **important** and what kind of **interactions** between them take place?
    - Global model interpretability helps to understand the **distribution of the target outcome based on the features**.
    - Global model interpretability is very difficult to achieve in practice → Any model that exceeds a handful of parameters or weights is difficult to understand
    - Some models are interpretable at a parameter level :
      - For linear models, the interpretable parts are the weights,
      - For trees interpretable parts are the splits (selected features plus cut-off points) and leaf node predictions.
  - Global Interpretation tools
    - Interpretable Models by nature (eg. Linear models, Regression Tree)
    - Feature Importance
    - Partial Dependant Plot (PDP), Individual Conditional Expectation (ICE) and Accumulated Local Effects (ALE)
    - Interaction Measures (H-statistic)

Global Model Interpretation
Features Importance

- **Features Importance**
  - In a tree-based method : Go through all the splits for which the feature was used and measure how much it has reduced the Loss Function (e.g. MSE, Poisson Deviance,…) compared to the parent node.
  - The sum of all importance measures is scaled to 100.
  - This means that each variable importance can be interpreted as share of the overall model importance
- One can get additional measures such as:
  - Minimal depth and its mean :
    - Which variables were the most often on the top of the tree
    - Mean depth of first split
- Features Importance can be used as a **features' selection tool**
  - Goal: Identify the **most relevant variables**
  - Pay attention: when some variables are correlated, their **global impact can be spread** between them, therefore reducing individual importance of each variable





Global Model Interpretation
Partial dependence plot

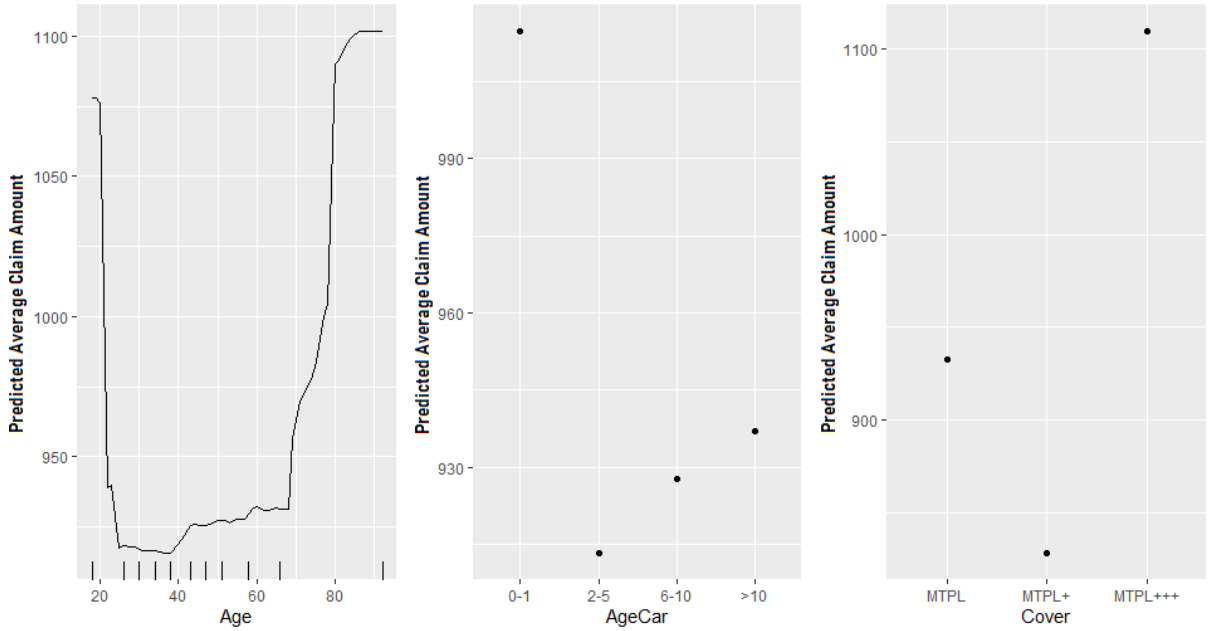
- **Partial Dependence Function/Plot**
  - Partial dependence plot (short PDP or PD plot) shows the **marginal effect one or two features** have on the predicted outcome of a machine learning model
  - It can be computed as

$$PD_{age}(age) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}\left(age, agecar^{i}, cover^{i}, \dots\right)$$

  - In this formula, $agecar^{i}$,… are actual features' values from the dataset for the features in which we are not interested, $\hat{f}$ is the trained model and n is the number of instances in the dataset.
  - So we marginalize model outputs over the distribution of the features we are not interested in (e.g. agecar, cover, …)
    - → the function shows the relationship between the feature age we are interested in and the predicted outcome.
    - → By marginalizing over the other features, we get a function that depends only on features age, interactions with other features included.
  - **Attention point**: With correlated features, computation of a PDP involves averaging predictions of artificial data instances that can be **unlikely in reality**.
    - → Alternatives exist: Individual Conditional Expectation (**ICE**) or Accumulated local effect plot (**ALE**)

Global Model Interpretation
Partial dependence plot

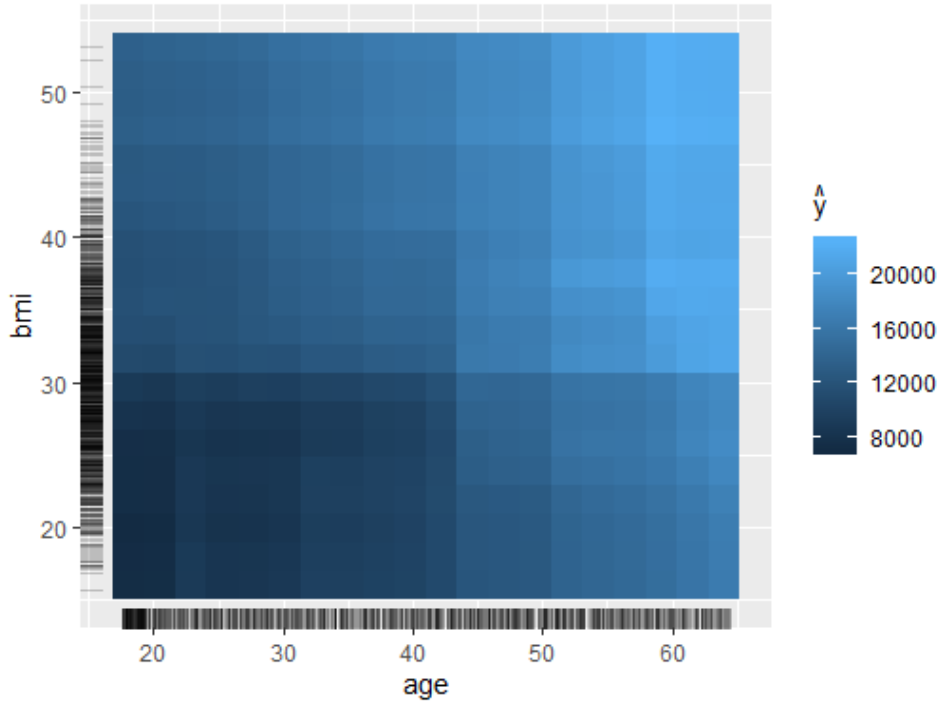
- Example of Partial Dependence Plot (1D) on Average Claim Amount :



- Partial dependence plot can be used as a **features' impact explanation tool**
  - It allows to better understand the marginal impact of a variable on the prediction
  - It is very similar to the interpretation of the multiplicative factors we obtain in a GLM or GAM model

Global Model Interpretation
Partial dependence plot

- Example of Partial Dependence Plot (2D) :
  - PD can be generalized to more than one feature
  - PDP -2D can be very useful to highlight interactions



Global Model Interpretation
Detection of interaction between variables with H-Statistics

- Interaction Measures (H-Statistics)
  - In case of interaction, prediction cannot be expressed as the sum of the feature effects, because the effect of one feature depends on the value of the other feature
  - **How to measure the level of interaction between two features?**

→ Have a look at H-Statistic. The main idea is:

- If two features do not interact, we can decompose the partial dependence function

$$PD_{age,power}(age, power) = PD_{age}(age) + PD_{power}(power)$$

- Measure the difference between the observed partial dependence function and the decomposed one without interactions.

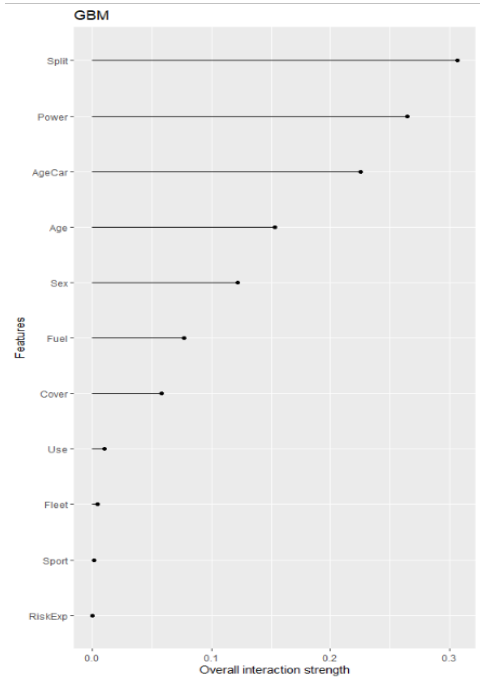
$$H^2 = \sum_{i=1}^{n}\left[PD_{age,power}(age^i, power^i) - PD_{age}(age^i) - PD_{power}(power^i)\right]^2 \Big/ \sum_{i=1}^{n} PD_{age,power}{}^2(age^i, power^i)$$

- *H is 0 if there is no interaction at all*
- *A value H of 1 between two features means that each single PD function is constant and the effect on the prediction only comes through the interaction.*

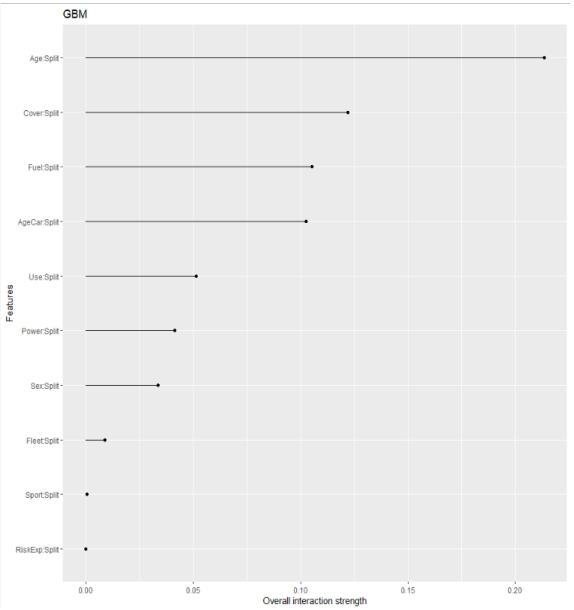
  - It is also possible to measure the **total interaction** of a feature which tells us **whether and to what extent a feature interacts** in the model **with all other features**

Global Model Interpretation
Detection of interaction between variables with H-Statistics

Total interaction for each feature with all other features



2-way interactions between the split feature and the other features



- H-Statistics can be used as a **features' interaction identification tool**
  - It allows to identify features strongly interacting with other features
  - It can then be used for **features engineering** (e.g. creating a new feature as an interaction between 2 features)

Global vs local Interpretability of ML techniques

- **Local Interpretability for a Single Prediction**
  - Why did the model make a certain prediction for an instance?
    - If you look at an individual prediction, the behavior of the otherwise complex model might behave more pleasantly.
    - You can zoom in on a single instance and examine what the model predicts for this input, and explain why.
      - Shapley Value
      - Breakdown

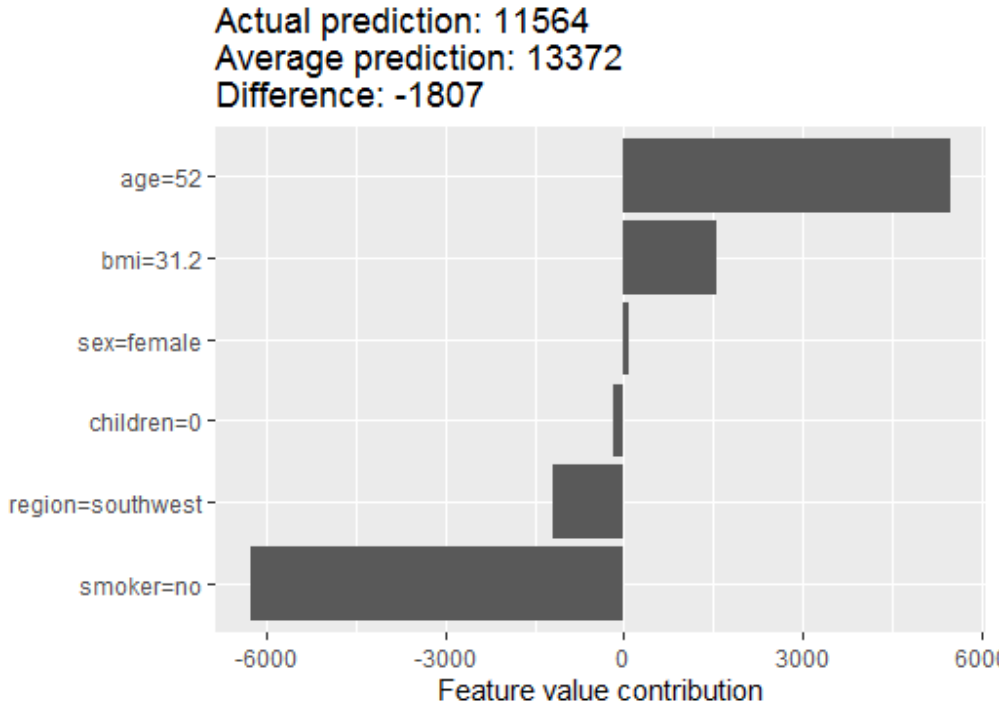
- **Local Interpretability for a Group of Predictions**
  - Why did the model make specific predictions for a group of instances?
    - Model predictions for multiple instances can be explained either with global model interpretation methods or with explanations of individual instances.
    - The global methods can be applied by taking the group of instances, treating them as if the group were the complete dataset, and using the global methods with this subset.
      - LIME (Local Interpretable Model-agnostic explanations)
      - LIVE
    - The individual explanation methods can be used on each instance and then listed or aggregated for the entire group.

Local Interpretability for a Single Prediction

- **Shapley Value :**
  - The shapley value measures for a single prediction how much each specific feature value will contribute to make the instance prediction different from the overall prediction.
  - The computation time increases exponentially with the number of features.

From Game Theory

- The Shapley value is the average marginal contribution of a feature value across all possible coalitions (= sets composed of different number of features).
- For each of these coalitions we compute the prediction with and without the feature value of interest and take the difference to get the marginal contribution.
- The Shapley value is the (weighted) average of marginal contributions across all the coalitions.



Agenda

- 1. A non-exhaustive reminder to some useful ML techniques
- 2. Adding complexity means increasing need for interpretability
- 3. An introduction to ML interpretation tools
- 4. Conclusions: how to make the most of ML techniques

How to make the most of ML techniques?

Two different strategies

1. Replacing traditional models (e.g. GLM) by ML models
2. Combining the pros of traditional and ML models to improve predictive modelling

Replacing traditional models by ML models

- The main drawback of this approach is the black-box effect of the ML results

- There is therefore a strong need in using interpretations tools
  - **Feature importance** to select the most relevant variables (e.g. if we have too many variables available and/or we want to limit the number of modelled variables)
  - **PDP, ICE, ALE and/or H-Statistics** to understand the impact of the selected variables on the prediction and identify the potential interactions
  - **Shapley value** to better understand the prediction on specific data points

How to make the most of ML techniques?

Combining traditional and ML models

- ML methods would then be used to perform features extraction, features selection and/or features engineering
  - **Feature extraction** = reducing the dimensionality of too voluminous datasets (in terms of # features)
  - **Feature selection** = selecting the most relevant variables to our problem
  - **Feature engineering** = identifying the best representation of the sample data to learn a solution to your problem (e.g. interactions)

- The selected/engineered variables could then be **introduced in our usual model (e.g. GLM) in order to obtain easily interpretable results combined with more insights**

For a more complete presentation of interpretability tools, check Reacfin webinar on "Explainable machine learning" https://www.reacfin.com/index.php/webinars/

Thank you very much for your attention!

Contact details

Xavier Maréchal

CEO of Reacfin

xavier.marechal@reacfin.com

M +32 497 48 98 48

About us

Reacfin is a consulting firm focused on setting up top quality, tailor-made risk management frameworks and offering state-of-art actuarial and financial techniques, methodologies and risk strategies.

Reacfin

Know-How to Risk

Reacfin Tooling

Reacfin Consulting

Reacfin Academy

Reacfin SA - Place de l'Université, 25 B-1348 Louvain-la-Neuve (Belgium) - 0032 0 10 84 07 50 - www.reacfin.com