AKUR8

SAV Mitgliederversammlung

August 26th 2022

Jan Küthe

Aktuar (DAV) / Actuarial Data Scientist

Biography

Jan is an Actuary (DAV) from Germany and works at Akur8 as an Actuarial Data Scientist to help insurance companies unlock the potential of twenty-first century pricing methods.

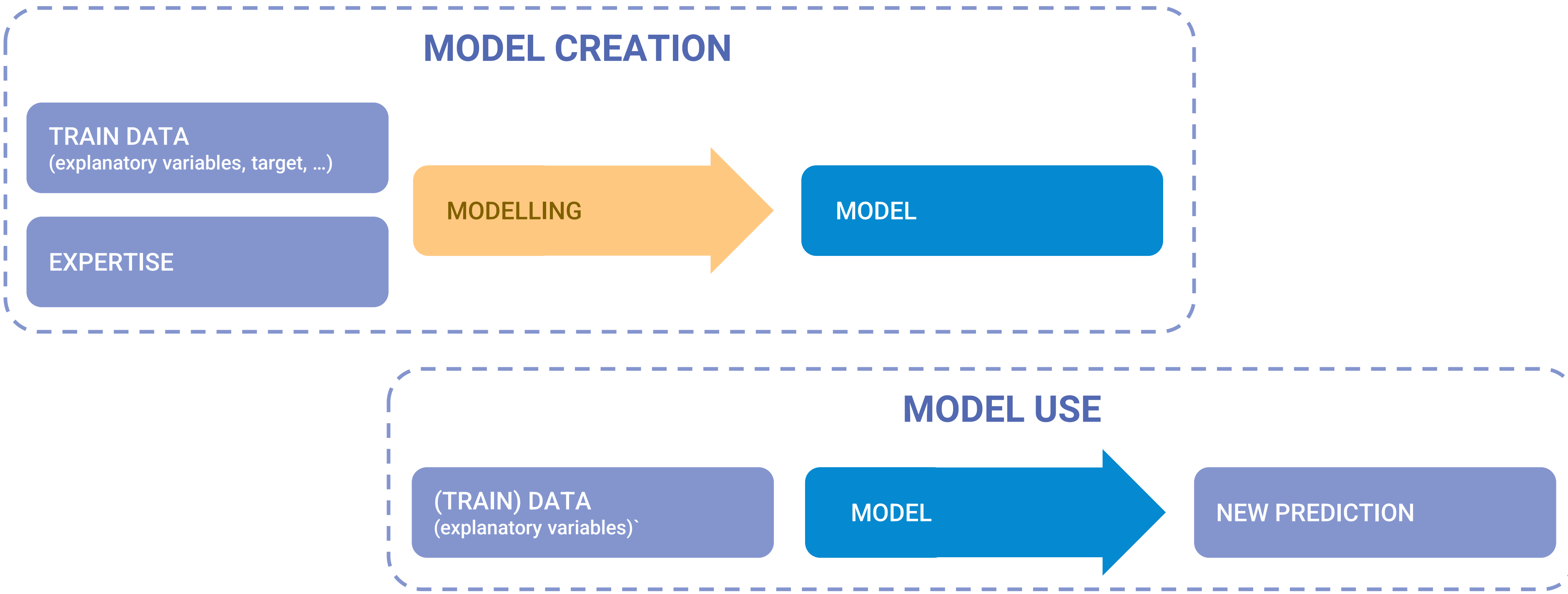
Before that he has been working in a global Actuarial Consultancy for three years. He holds a Master's degree in Mathematics from the University of Bonn and lives in Cologne.

AKUR8

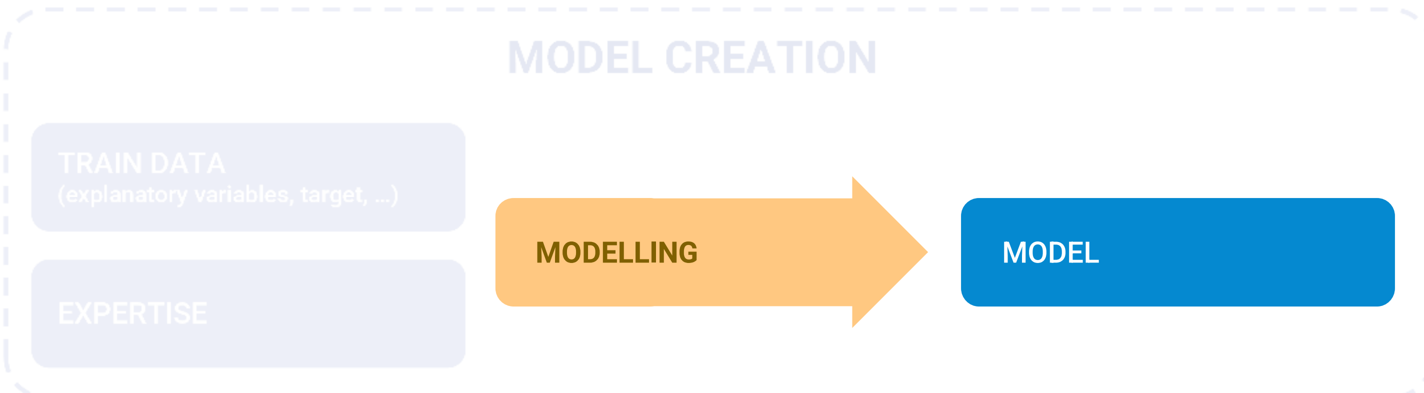
Modelling and Models

The Modelling Approach

Creating & using models



The Modelling Approach
Creating & using models



- What is a **transparent model**?
- What is **machine-learning**?
- Why are they **opposed**?
- What happens when the two concepts are **combined**?

The choice between black-box ML and traditional GLMs (presentation by Swiss Re)

Model approach comparison

GLM vs. other ML-methods

	XGBoost	Random Forest	GLM
Automatic Feature selection	✓	✓	X
Model Runtime	Longer	Medium	Short
Performance (AUC)	High	Medium	Medium
Interpretable results	X	X	✓

- Different modelling techniques display different performance along key measurement criteria
- Setting clear expectations a priori helps to select the preferred one

Presented by Swiss Re at the Swiss Association of Actuaries on the 06/10/2021: https://www.actuaries.ch/de/fach-arbeitsgruppen/fachgruppe-data-science/Ds%20AMs

Model creation & structure

	Creation Process	Result
GBM		**Black Box** (Trees Ensemble)
Random Forest		**Black Box** (Trees Ensemble)
Neural Network		**Black Box** (Neural Network)
Data-Prep + GLM		**Transparent** (Data-Prep + LM.)
GAM (manual)		**Transparent** (GAM)

Classic Actuarial approach

Transparency: Direct Models Visualization

While **model interpretability techniques** can be applied to any model, a **direct model understanding** is restricted to the specific class of models

To be understood, models must be:

- **Reducible:** the effects of the model can be **isolated** and **visualized** piece-by-piece
- **Parsimonious:** the model must incorporate a **limited number of effects** to be analyzable

This class of models restrict human-understandable models to:

→ **Simple rules**
→ **Shallow tree**
→ **Generalized Additive Models** (including GLMs), with **parsimonious interactions**

A GAM gives Direct Models Visualization

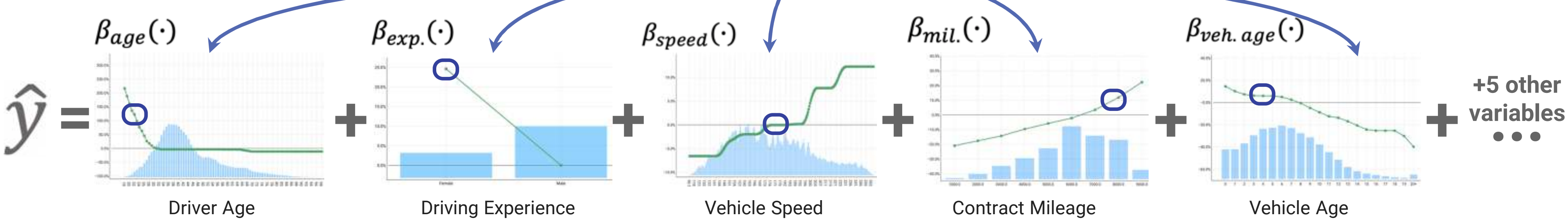
If a model can be decomposed, it can be visualized

Actuaries have been focusing during the past 20 years on the GAM modeling, because it allows the modeler to decompose the model's effects $\beta_j(X_j)$ and:

- Validate the effects
- "Force" the effects if no exposure is available

The GAM models are defined by their shape:

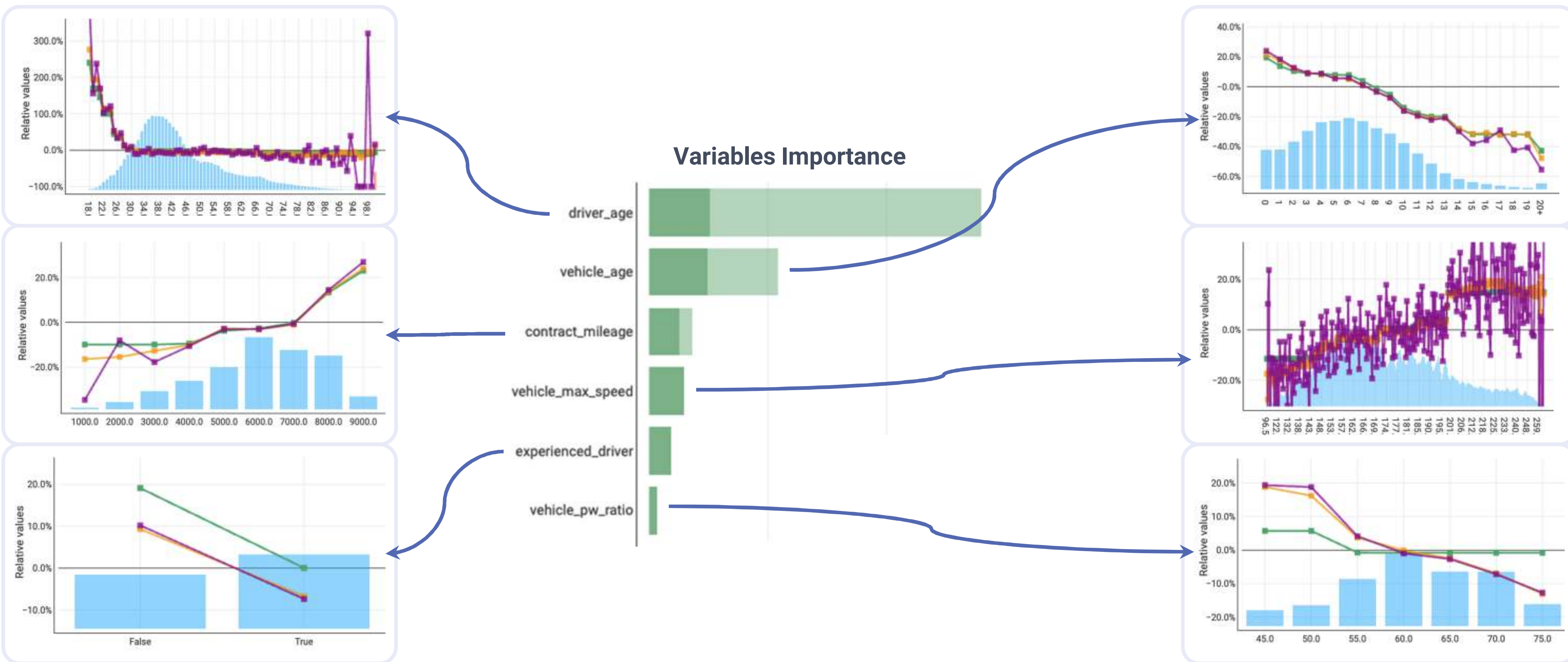
$$\hat{y}(X) = g^{-1}\left(\sum_d \beta_d(X_d)\right)$$



Here the model itself is visualized and fully understood by a human.

Analysing a GAM

Only a limited number of variables play a role; each variables' impact is fully known



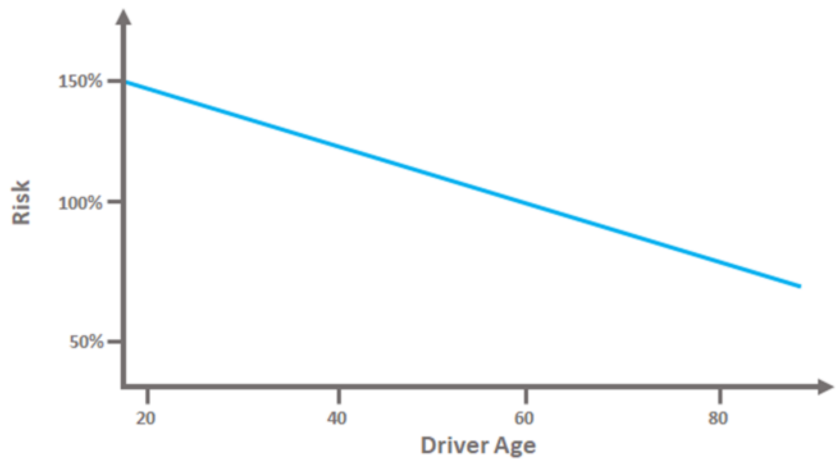
GLMs or GAMs

Linear or Additive

Linear models, GLMs and GAMs

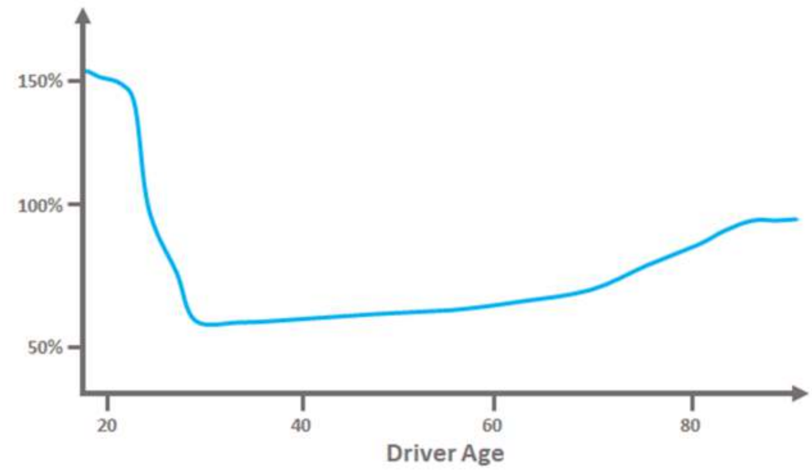
Linear Model

- Simple and well-known technique
- First regression created & learned
- Captures the linear relations in the data
- Simultaneously selects the variables and fit the trends

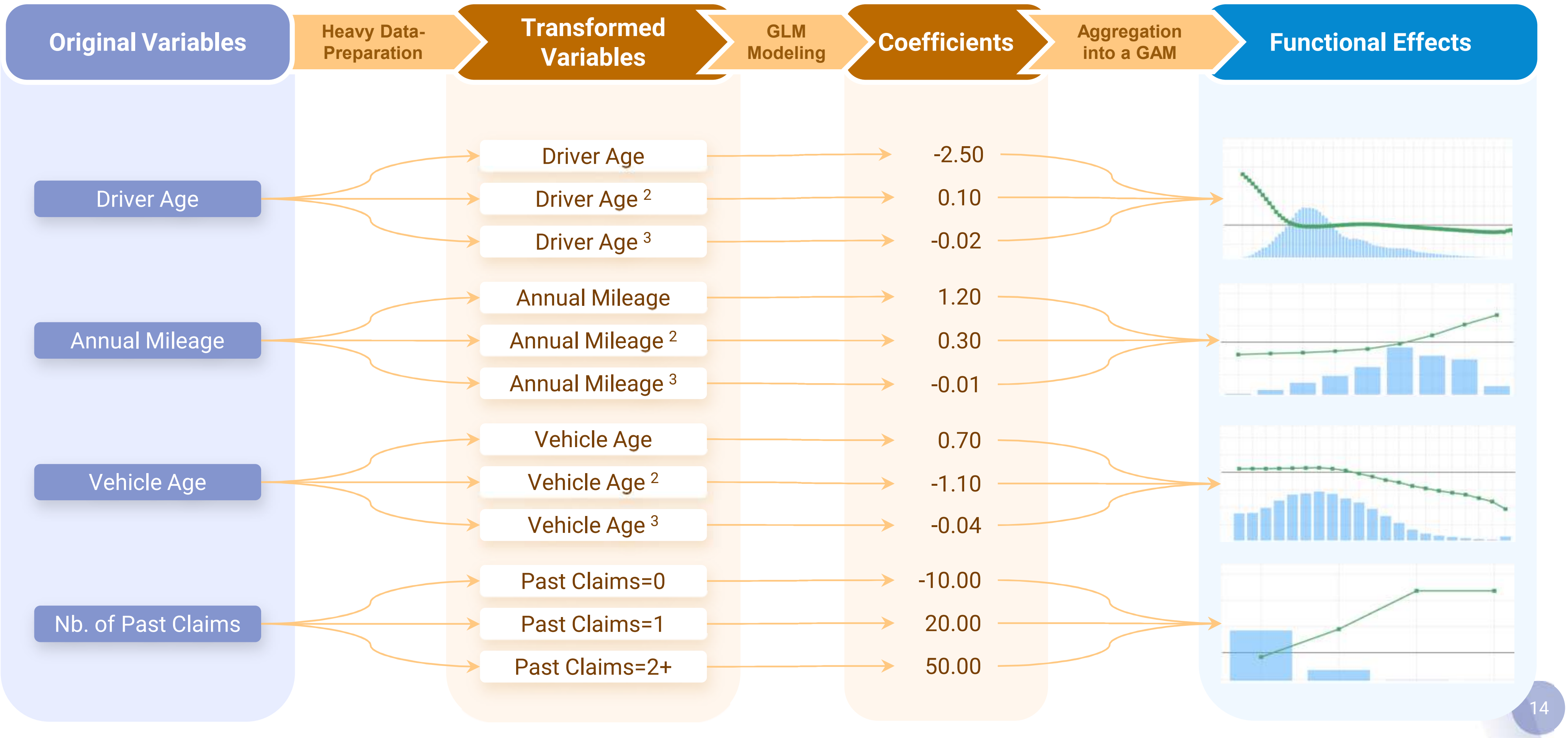


Additive Model

- Much more powerful models
- Captures non-linear effects
- Incorrectly called “GLMs”
- Requires both variables selection and fitting



Creating a GAM with variables transformations



Building GAMs manually

Only a limited number of variables play a role; each variable's impact is fully known

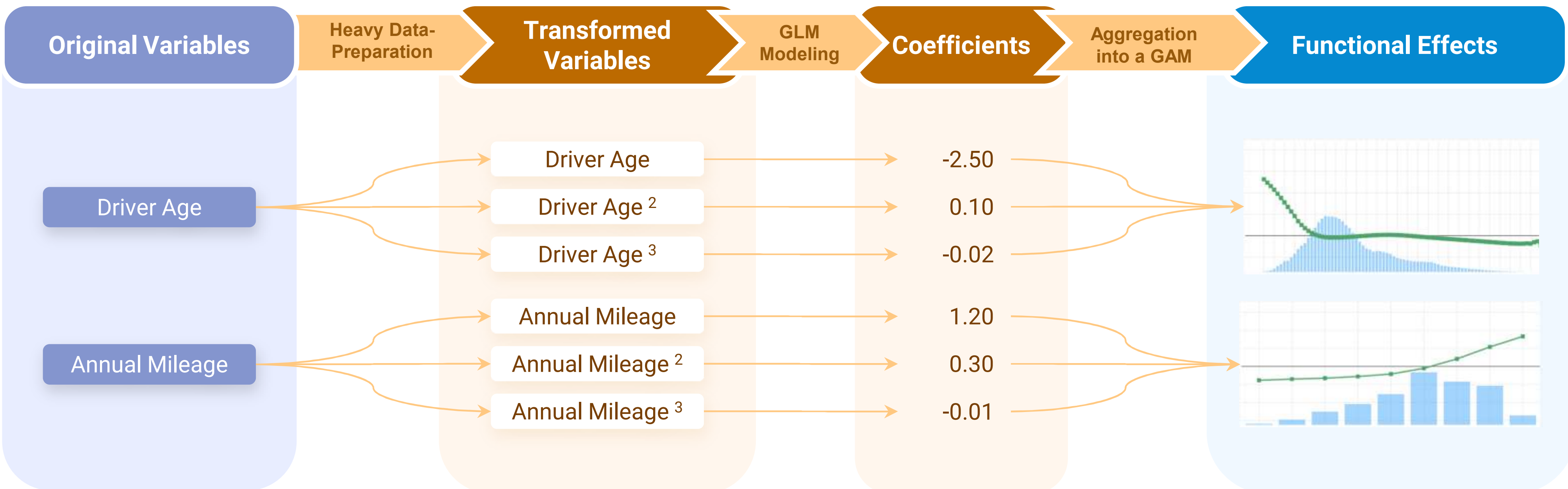
Generalized Additive Models are **transparent** by structure.

So, as modelers can **understand** and **interact** with them, it is possible to create them manually (unlike ensemble of trees, which have to be created by machines).

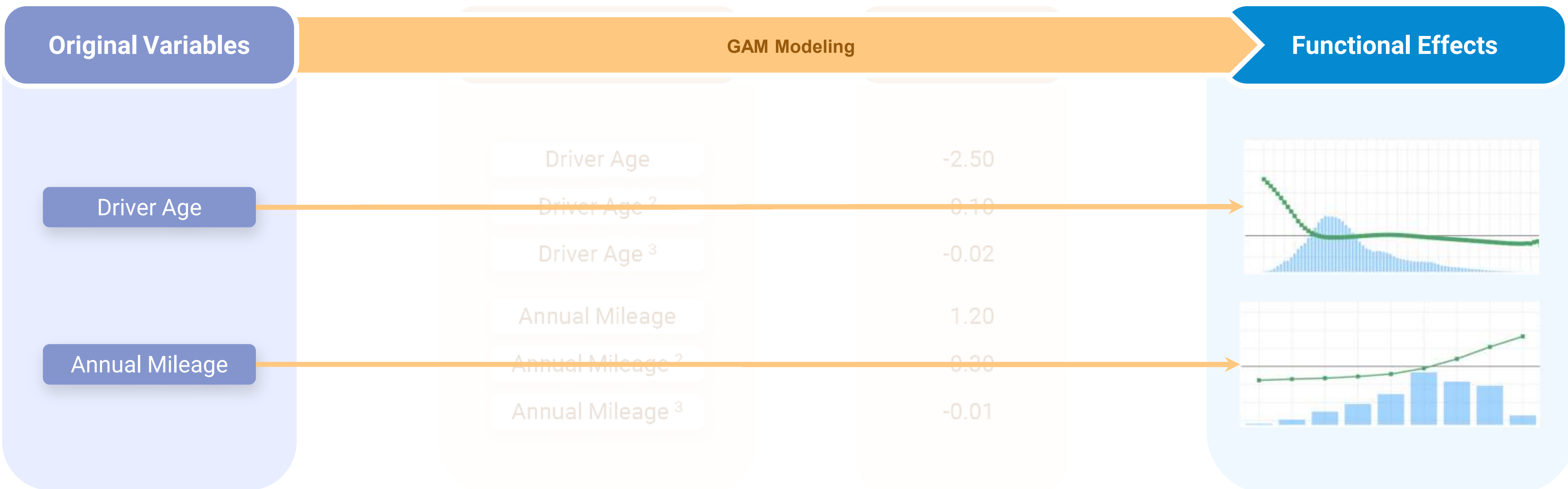
However, building GAMs through variables transformations and linear modeling leads to **severe limitations**!

- ✕ The right **set of variables** need to be **selected manually** by the modeler
- ✕ The right **transformations** need to be **manually created** by the modeler
- ✕ They are **limited to linear combinations** of the basis of variables created.
- ✕ Complexity is limited as creating too many transformations leads to **overfitting**.

Creating a GAM model through variable transformations...



… or creating a GAM with Machine Learning ?



Classic ML approach

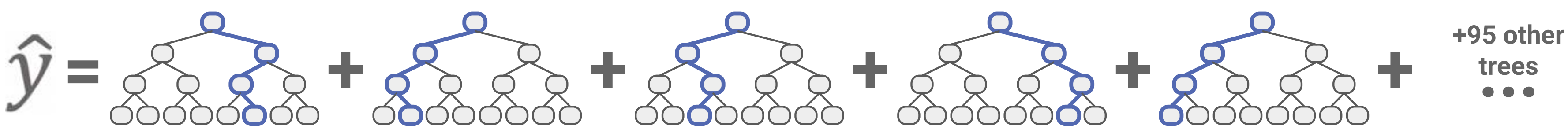
Black-Box models

But even Black-box models can be analyzed!

Most ML models are black-boxes: they **can't be directly understood, but can be analyzed.**

For instance, a Gradient Boosting generates predictions from an ensemble of decision trees: $\hat{y}(X) = g^{-1}\left(\sum_t \mathrm{T}_t(X)\right)$

Each tree T_t leverages all the dimensions of the data, generating interactions between the variables.



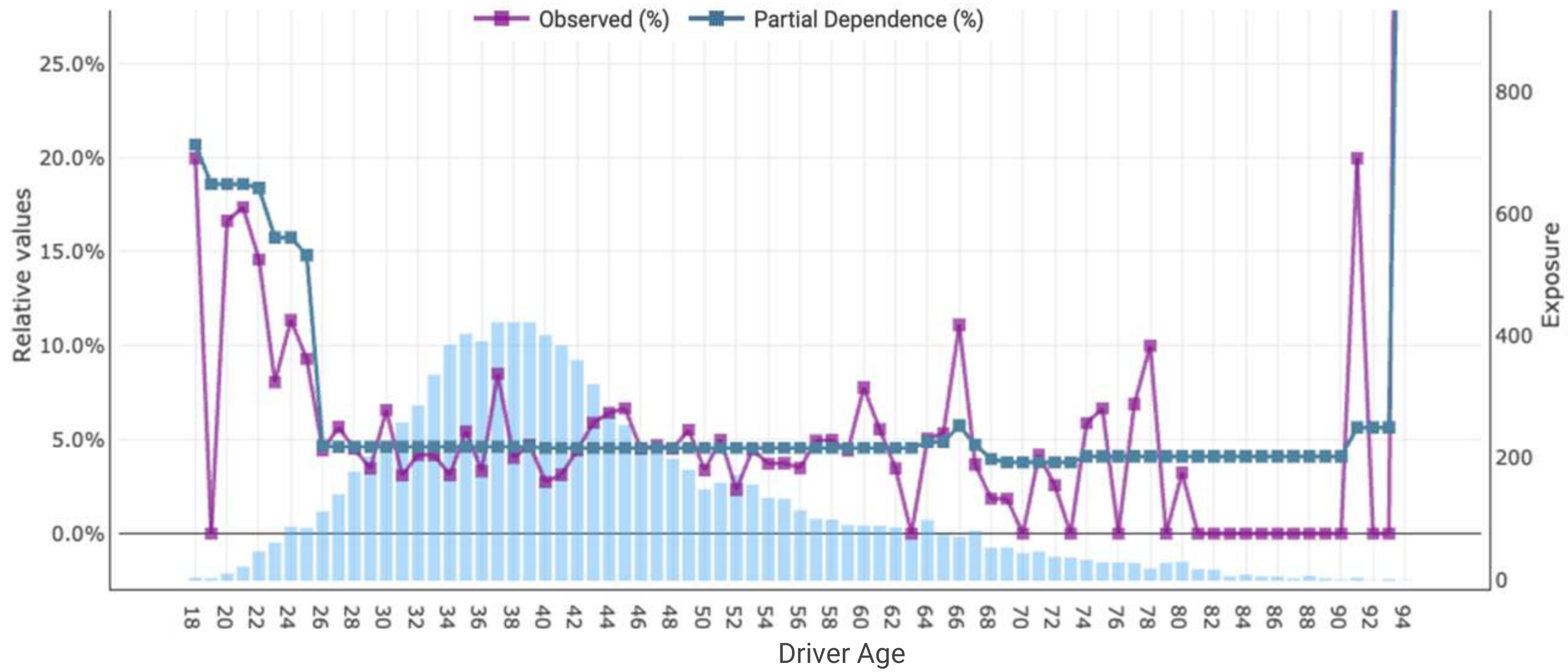
GBMs are really great because **they just work**:
it is straightforward to produce automatically good models.

As a GBM typically involves hundreds of trees of depth 2 to 6 (generating 2 to 6-way interactions),
this model is **not directly understandable** by a human.

For this reason, powerful model-analysis tools have been developed.

Example of black-box analysis

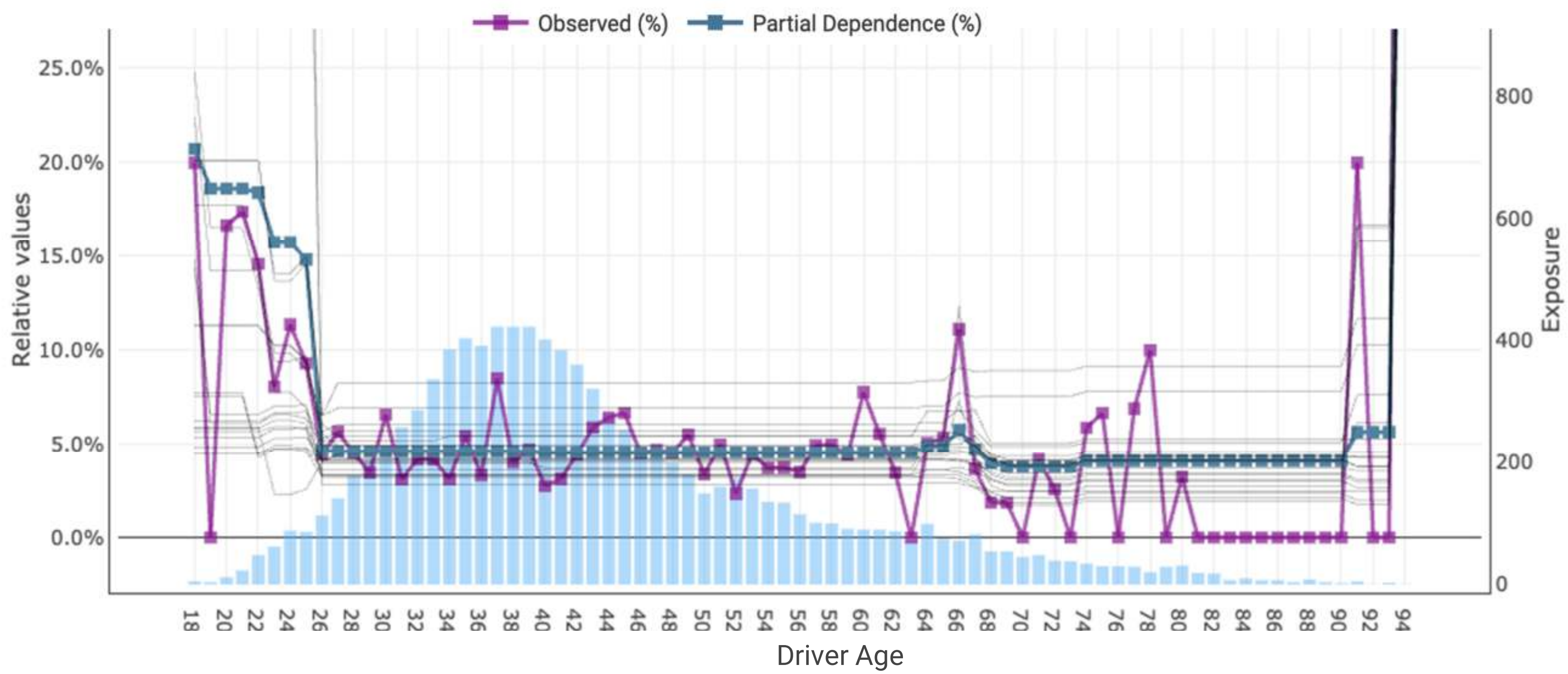
PDP: understand the global impact



For example: a Partial Dependence Plot (PDP)) and Individual Conditional Expectation (ICE) showing the impact of a driver's age.

Example of black-box analysis

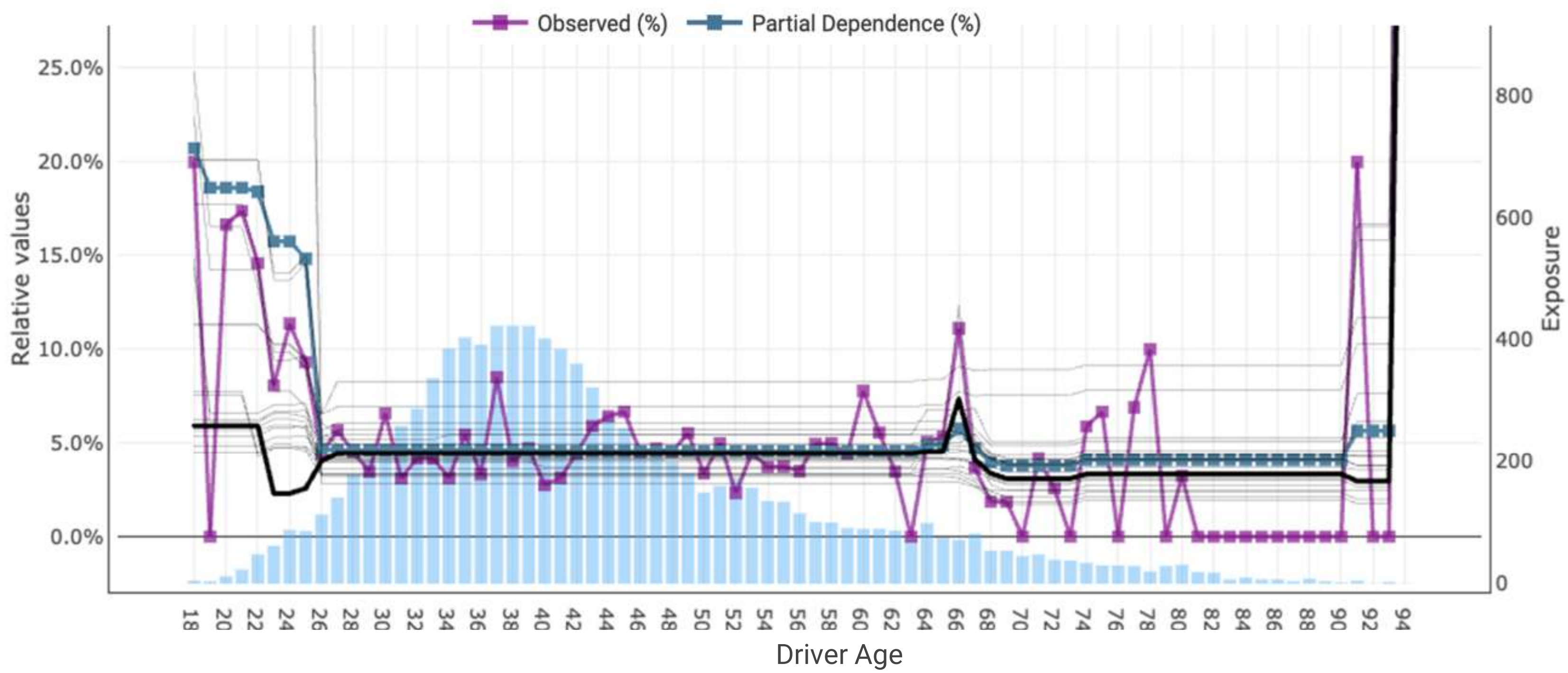
ICE: visualize the conditional impacts



For example: a Partial Dependence Plot (PDP)) and Individual Conditional Expectation (ICE) showing the impact of a driver's age.

Example of black-box analysis

ICE: visualize the conditional impacts



For example: a Partial Dependence Plot (PDP)) and Individual Conditional Expectation (ICE) showing the impact of a driver's age.

Global Parameters and Model Parameters

Models creation is automated:

- The user defines **global parameters** and **data**.
- The algorithm **fits** on the data and produces the **model**.

GLOBAL PARAMETERS

- Number of trees
- Trees depth
- Learning Rate

DATA

- Explanatory Variables
- Target Variable

FIT

MODEL PARAMETERS

- Ensemble of trees
  (split points, split variables, leaves estimates)

The model itself is often less looked-at than the global parameters.

For instance, when building a GBM, by maximizing the back-test results (through a k-fold) a user will find the global parameters, not the best model.

Finding the best Global Parameters

The Grid-Search approach

The grid-search approach seeks to find the best **Global Parameters**.

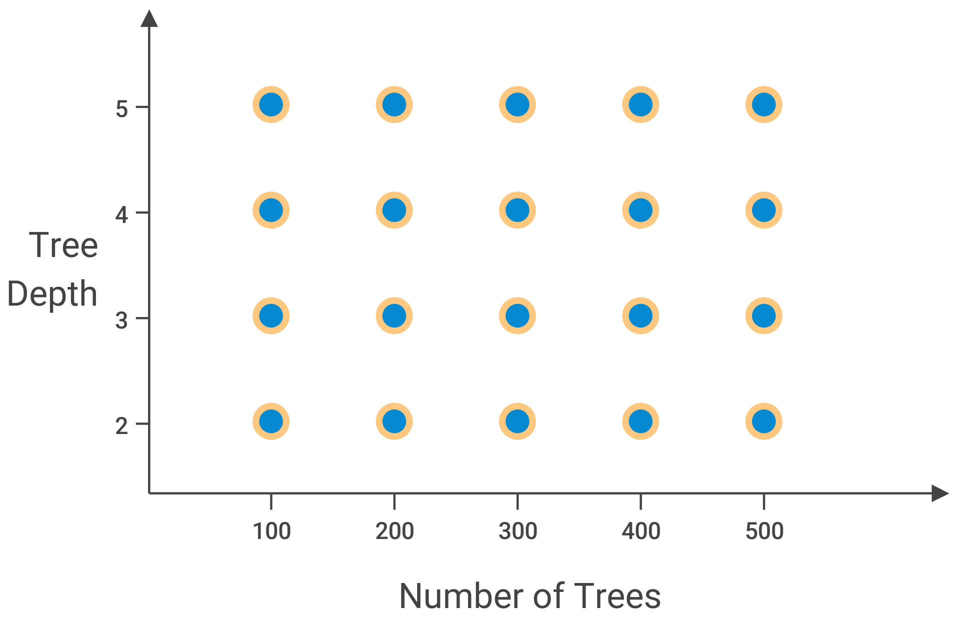
Based on a modeling data set, models are fitted with different global parameters, and their performance is measured.

The models themselves are not looked at: only the out-of-sample performance is considered.

The set of global parameters leading to the best performance is considered the best one.

They are used to fit a model on the entire data set: this model will be the one used in production.

It is possible to follow this whole process without ever looking at the selected model.



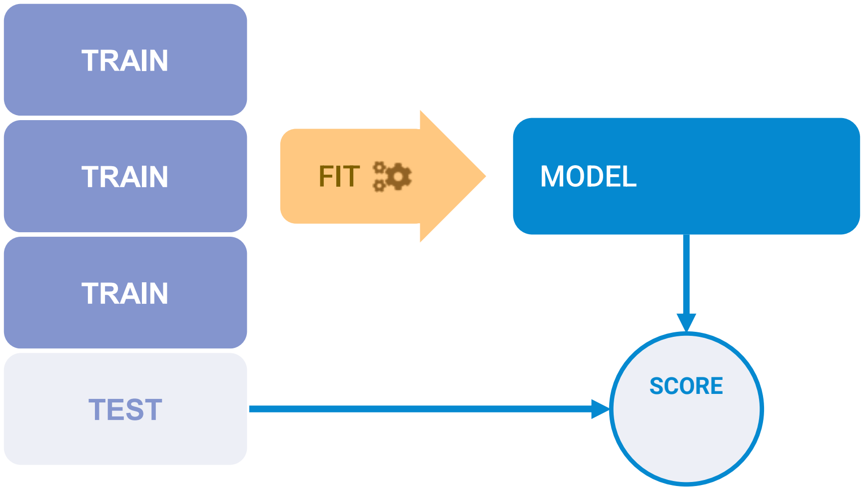
Testing a models performance

The k-fold approach

As the grid-search approach of modeling is “blind”, it relies a lot on performances measures.

To make sure the performance measure is as precise as possible, a k-fold approach is used: the data is split in K subsets (typically 4) and K models are created on all the subsets but one. The performances of these models are tested on the last subset.

This approach is very efficient, and works perfectly well independently of the model. However, it requires a completely automated model creation process.



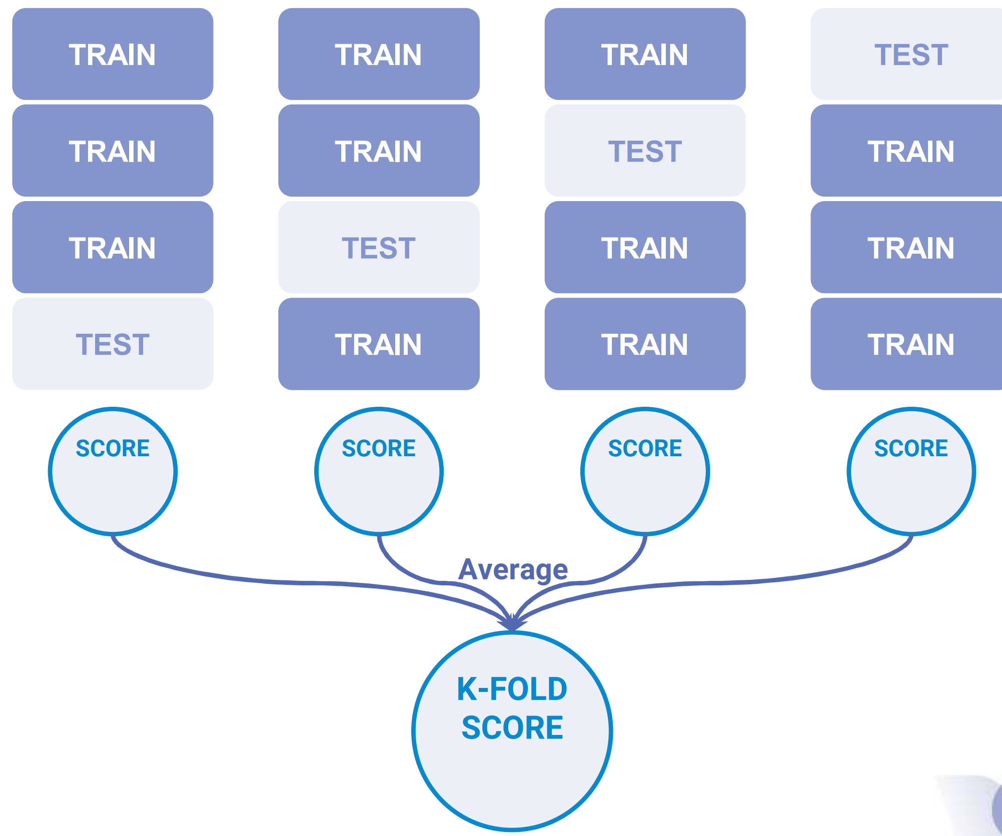
Testing a models performance

The K-Fold approach

As the grid-search approach of modeling is “blind”, it relies a lot on performances measures.

To make sure the performance measure is as precise as possible, a k-fold approach is used: the data is split in K subsets (typically 4) and K models are created on all the subsets but one. The performances of these models are tested on the last subset.

This approach is very efficient, and works perfectly well independently of the model. However, it requires a completely automated model creation process.



The Dilemma

Trees Ensembles and GAMs

Strengths and Limits

Strengths associated with **Tree ensembles** models are related to their **creation process**.

Strengths associated with **GAMs** are related to their **model structure**.

Tree Ensembles	GAM
Model structure • Sum of small effects of all the variables • (Deep) Trees	**Model structure** • Sum of effects of single variables
Model Understanding • Via reverse-engineering or local analysis	**Model Understanding** • Direct visualization
Model Creation • Machine learning	**Model Creation** • Human-creation • Machine-learning (?)

Models creation & structure

	Creation Process	Result
GBM		Black Box (Trees Ensemble)
Random Forest		Black Box (Trees Ensemble)
Neural Network		Black Box (Neural Network)
Data prep + LM		Transparent (Data prep + LM)
GAM (manual)		Transparent (GAM)
GAM (automated)		Transparent (GAM)

Mixing ML & Actuarial approaches

AKUR8

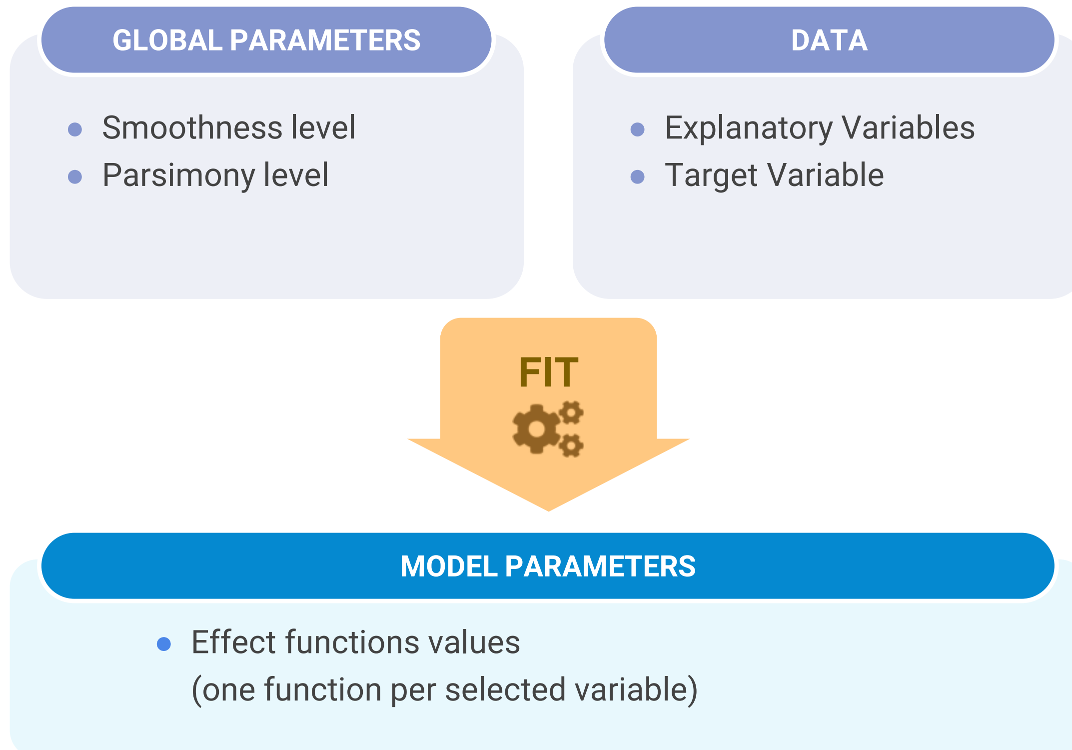
Global Parameters and Model Parameters

Applying ML to GAMs

It is possible to design an algorithm fitting GAMs, based on **2 global parameters**:

- **Level of smoothness**: how significant should the selected effects be?
- **Level of parsimony**: how many variables should be included in the model?

We developed this algorithm: Models can be **generated automatically** for many values of the global parameters (machine-learning Grid-Search approach), **tested on independant back-tests** and **fully analyzed**.



The Fitting Process

Optimizing the Likelihood with Constraints

Maximum of Likelihood

Maximize the Likelihood of the observations.

This is the standard approach used in GLM modeling, where the probability of observing the target given the predictors and a loss function (the likelihood) is optimized.

Smoothness Constraint

Similar to a **credibility approach**: all effects are supposed to be null.
This hypothesis is tested for every level and, **if the effect is significant** enough, it is included in the model.

More or less sensitive models are obtained by modulating the significance threshold: models selecting only significant effects will be very smooth and **robust**, models with more permissive threshold will be more **sensitive**.

Parsimony Constraint

In order to **improve the readability** of the models created, all the **least significant variables are not included** in the model.

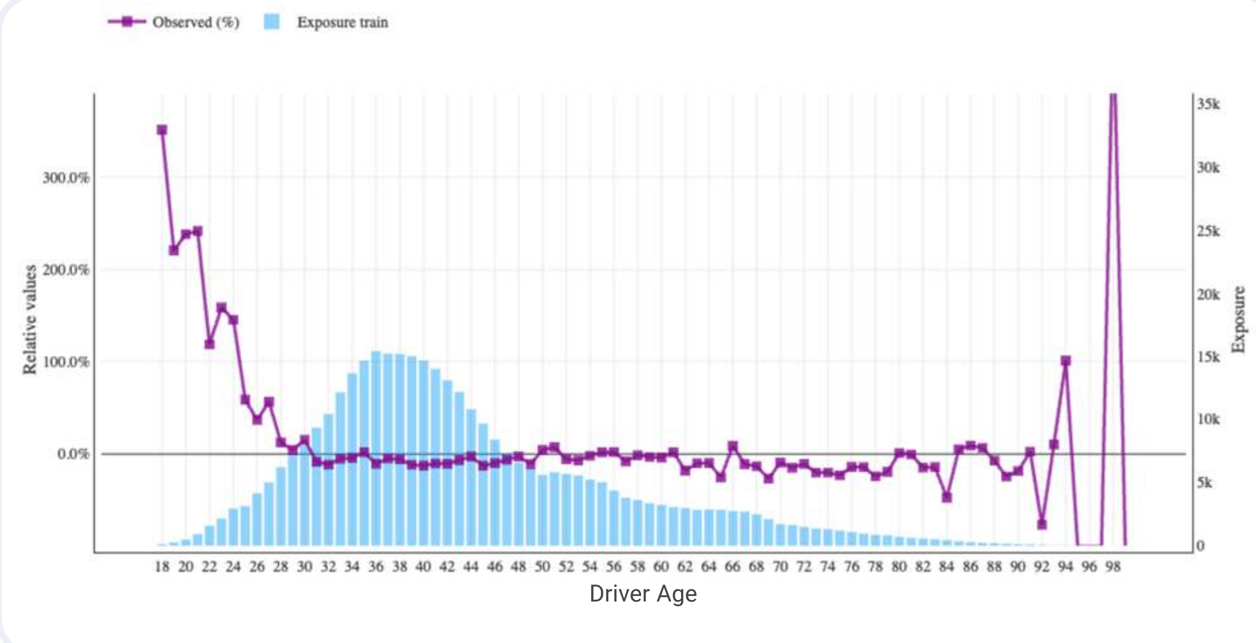
These are the variables that would provide the lowest gains in likelihood if included in the model.

This approach provides an optimal subset of variables to be included in the model.

1. Controlling the smoothness: Signal and Noise

Raw data contains both **signal** and **noise.**

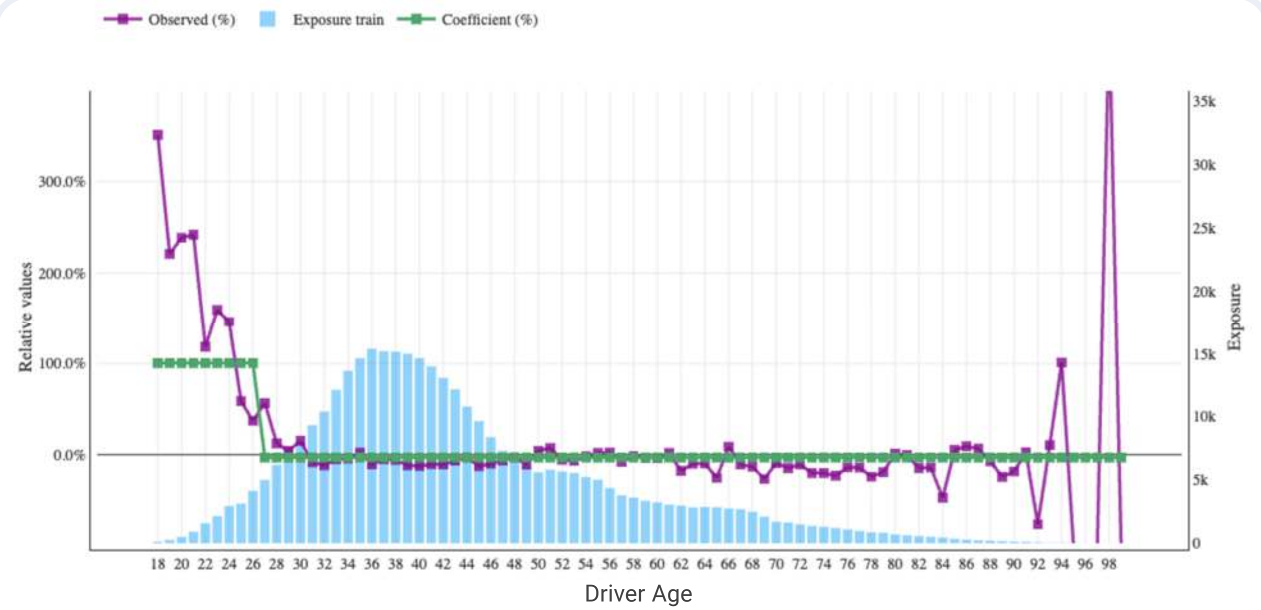
A trade-off needs to be found between robustness and sensitivity.



1. Controlling the smoothness: Signal and Noise

Robust model

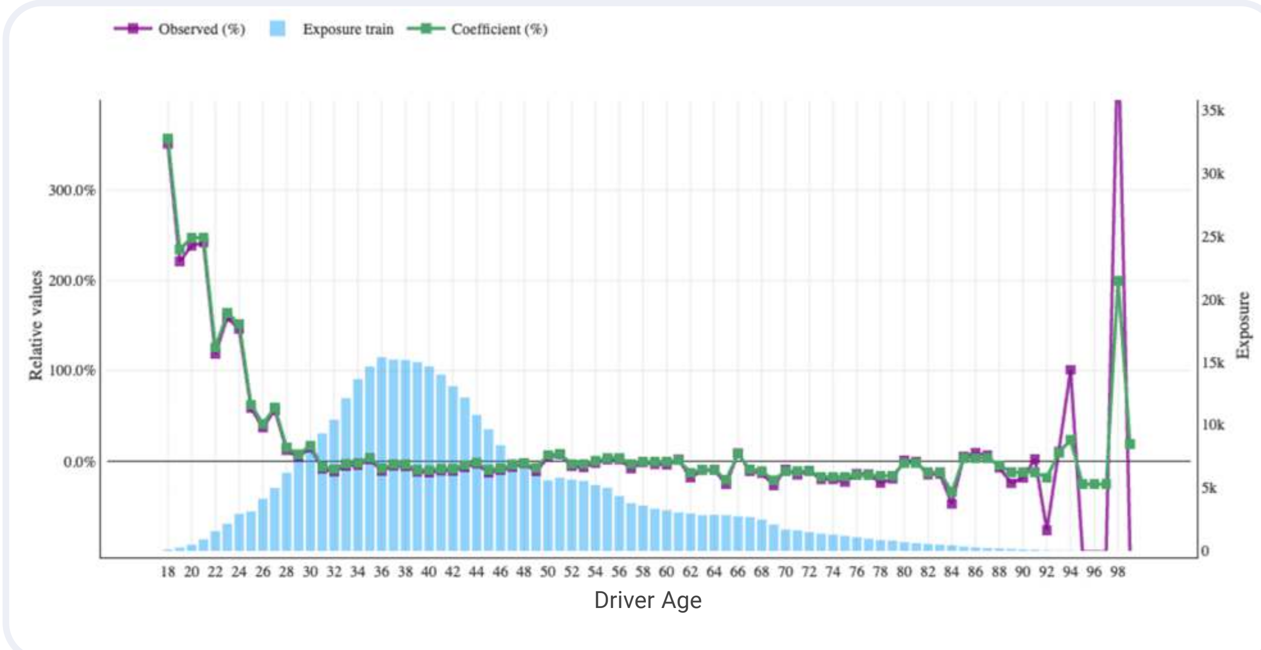
Missing part of the predictive signal



1. Controlling the smoothness: Signal and Noise

Over-fitted model

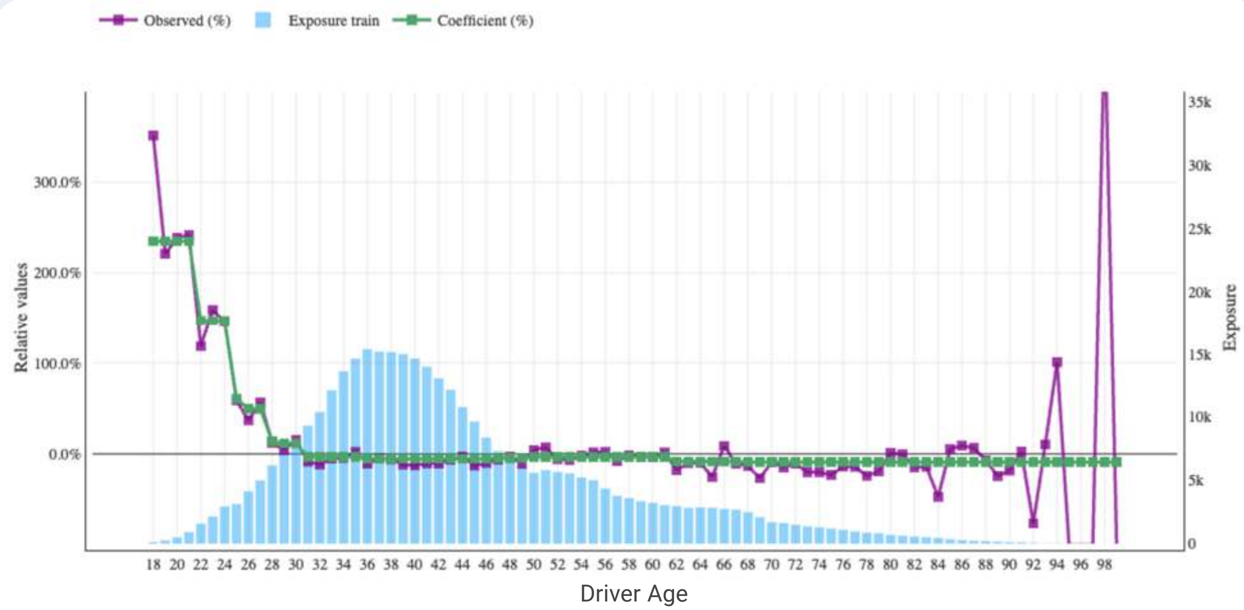
Capturing noise



1. Controlling the smoothness: Signal and Noise

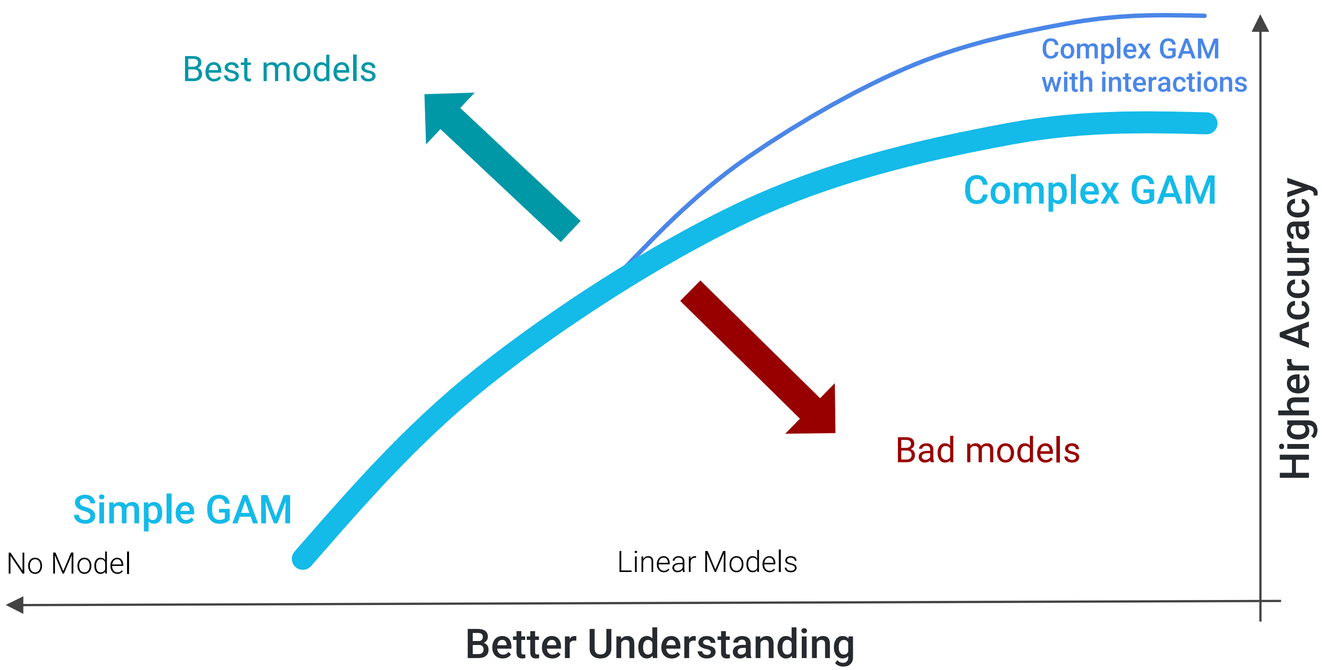
Efficient model

Good trade-off, capturing signal and rejecting noise



2. Parsimony has a cost (but it is worth it)

Understanding / Accuracy trade-off

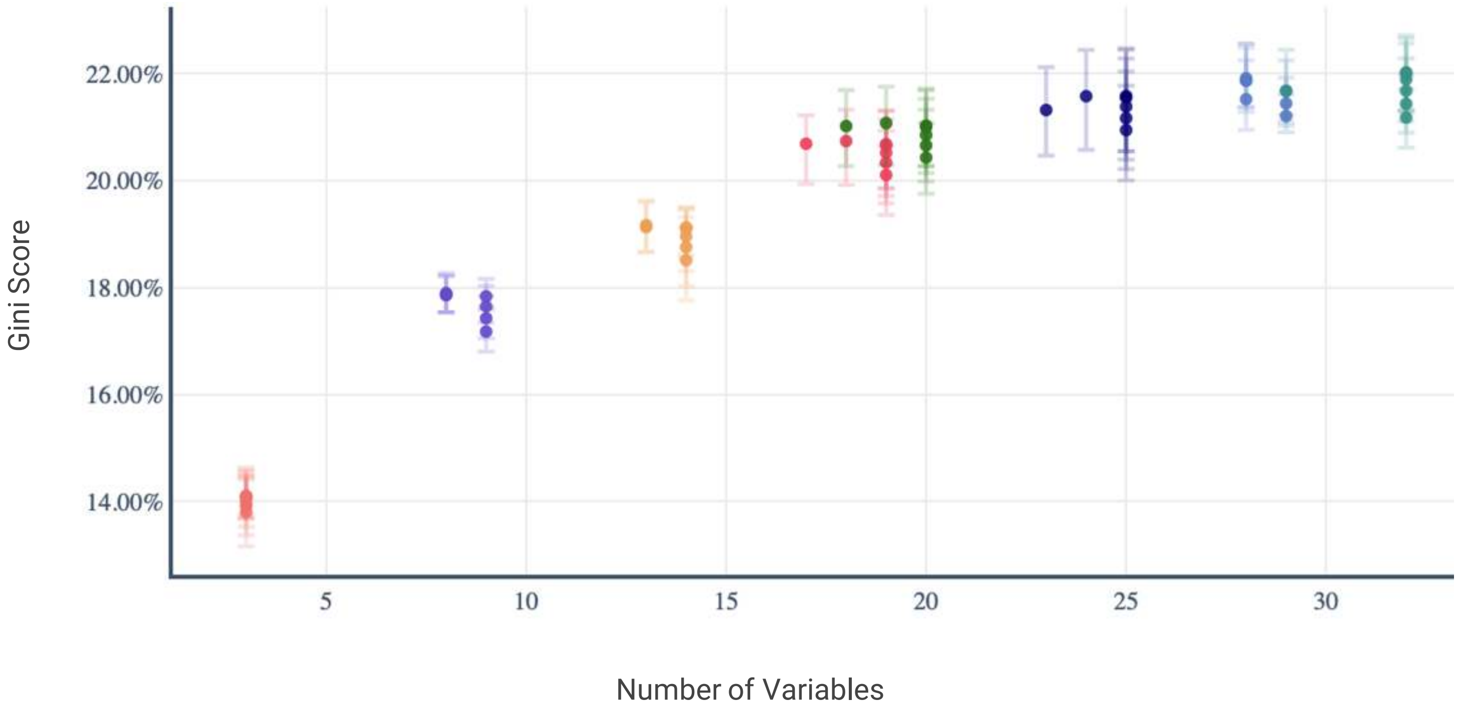


Black-box models
(GBMs, RF, NN...)

The accuracy is measured on a back-test; actual results when moving to productions will not be

2. **Parsimony has a cost** (but it is worth it)

Grid-search result

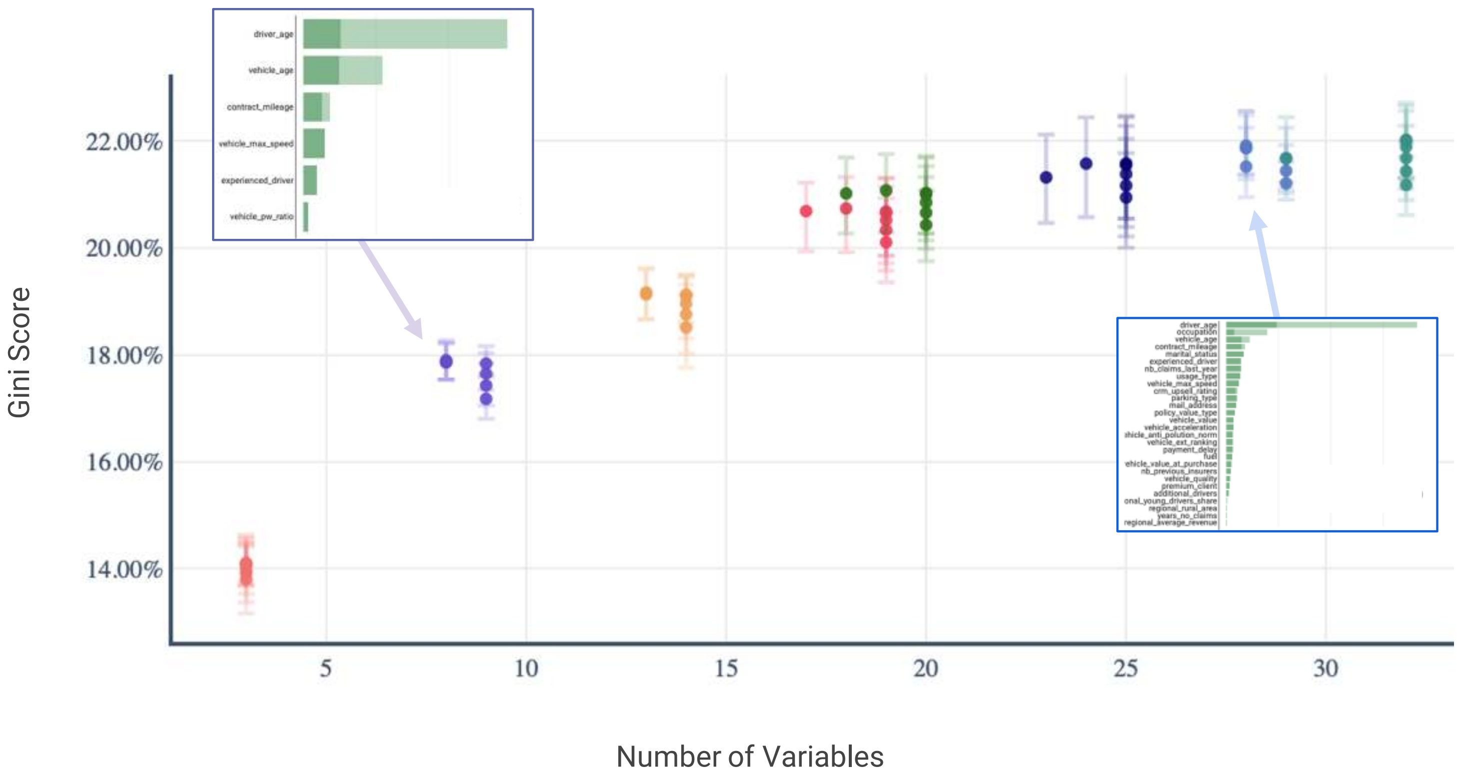


Grid-search results:
each **point** represents
one **model**.

The gain in models quality and the fading marginal improvement are clearly visible.

2. Parsimony has a cost (but it is worth it)

Grid-search result



Grid-search results:
each **point** represents
one **model**.

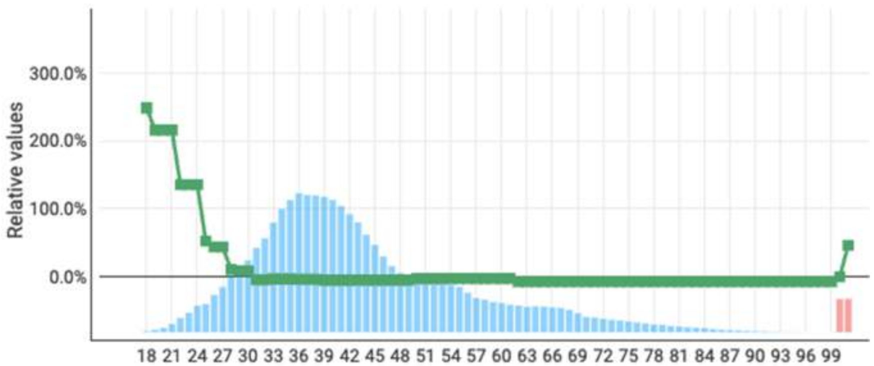
The gain in models quality and the fading marginal improvement are clearly visible.

3. Stay in control and interpret what you see

What is overfitting?

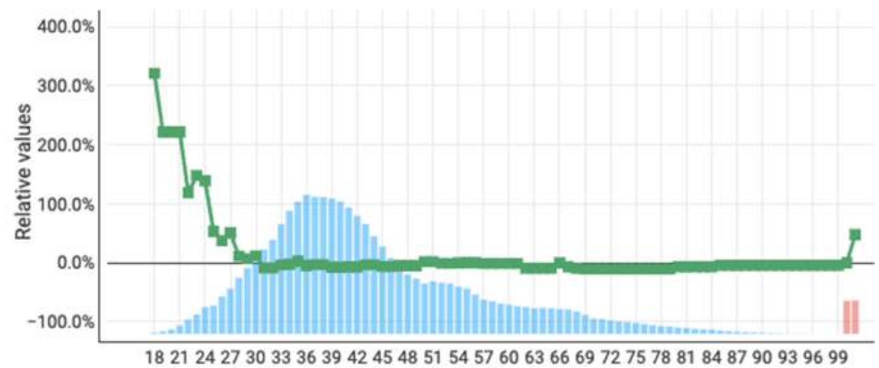
Which model should be selected?

Out-of-sample **Gini: 20.5%**
Robust model



Model on the left might lead to **better results** once **deployed in production**.

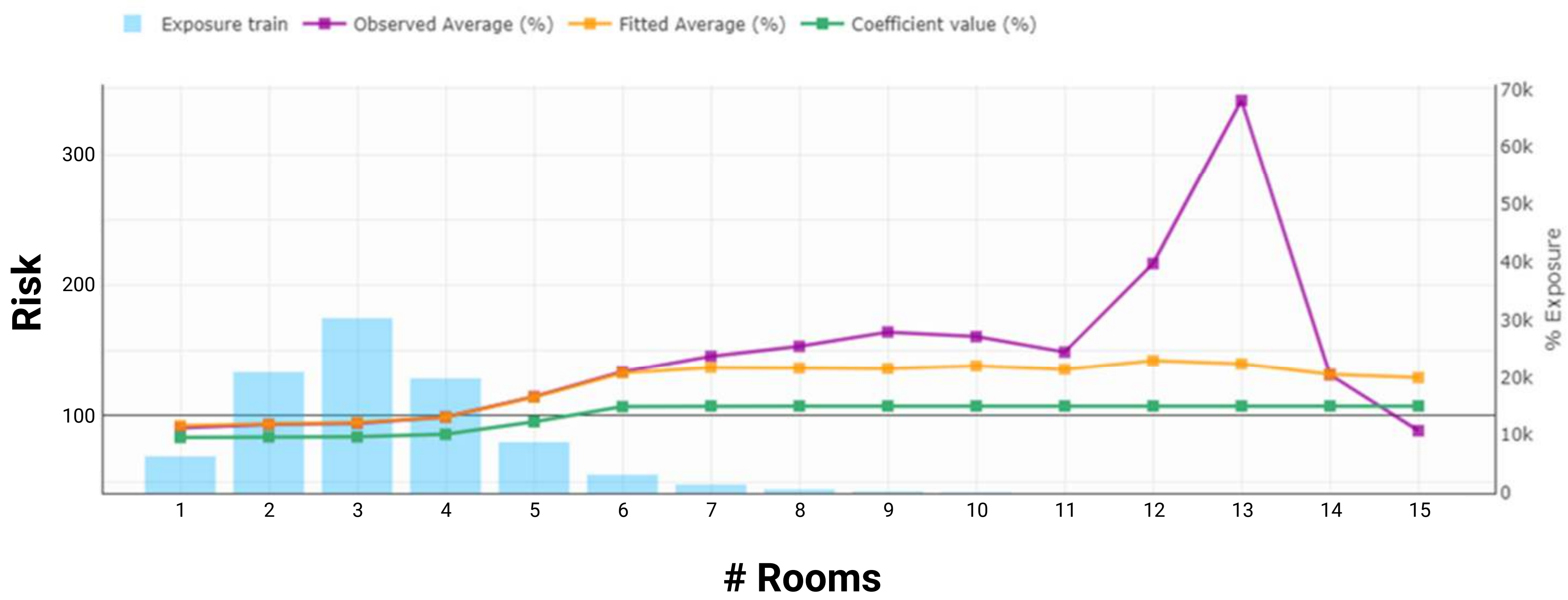
Out-of-sample **Gini: 21%**
Noisy Model



Model on the right has **stronger results on the back-test** but does **not inspire much trust**.

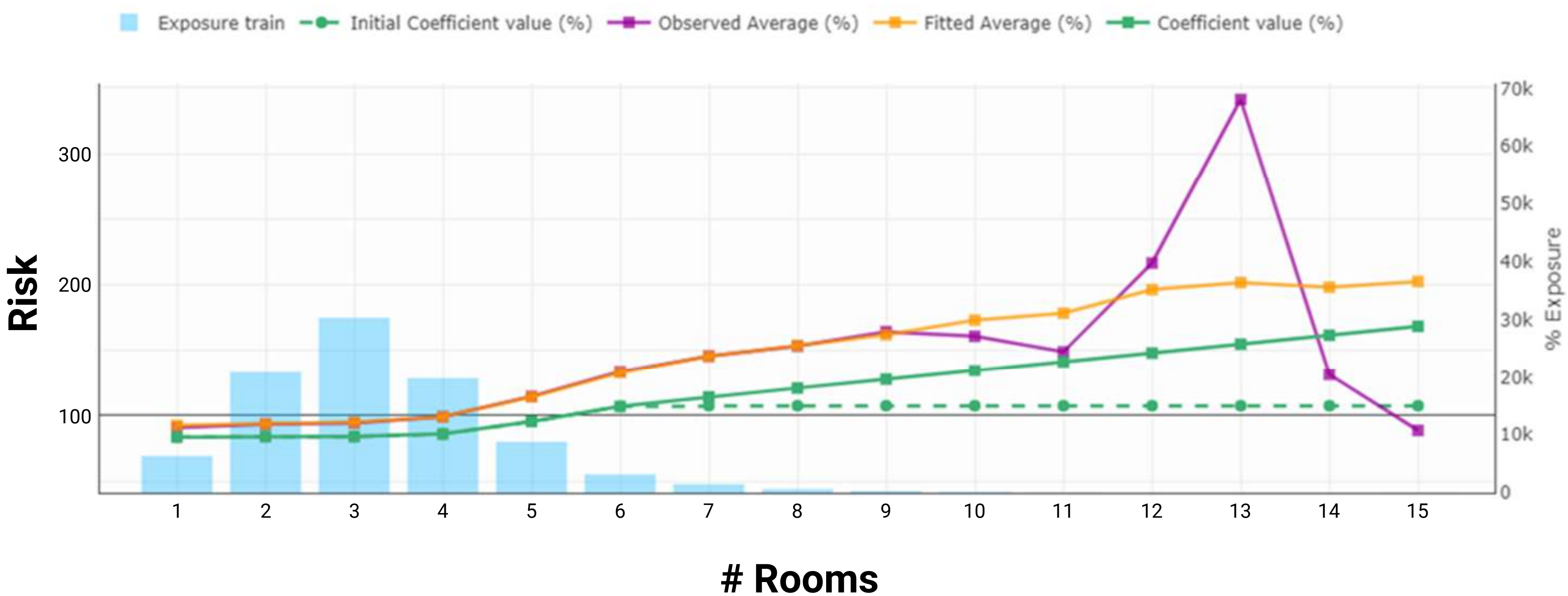
4. Interact with the models

Spotting the issues is nice..



4. Interact with the models

… solving the issues is better !



4. Interact with the models

A three-step process

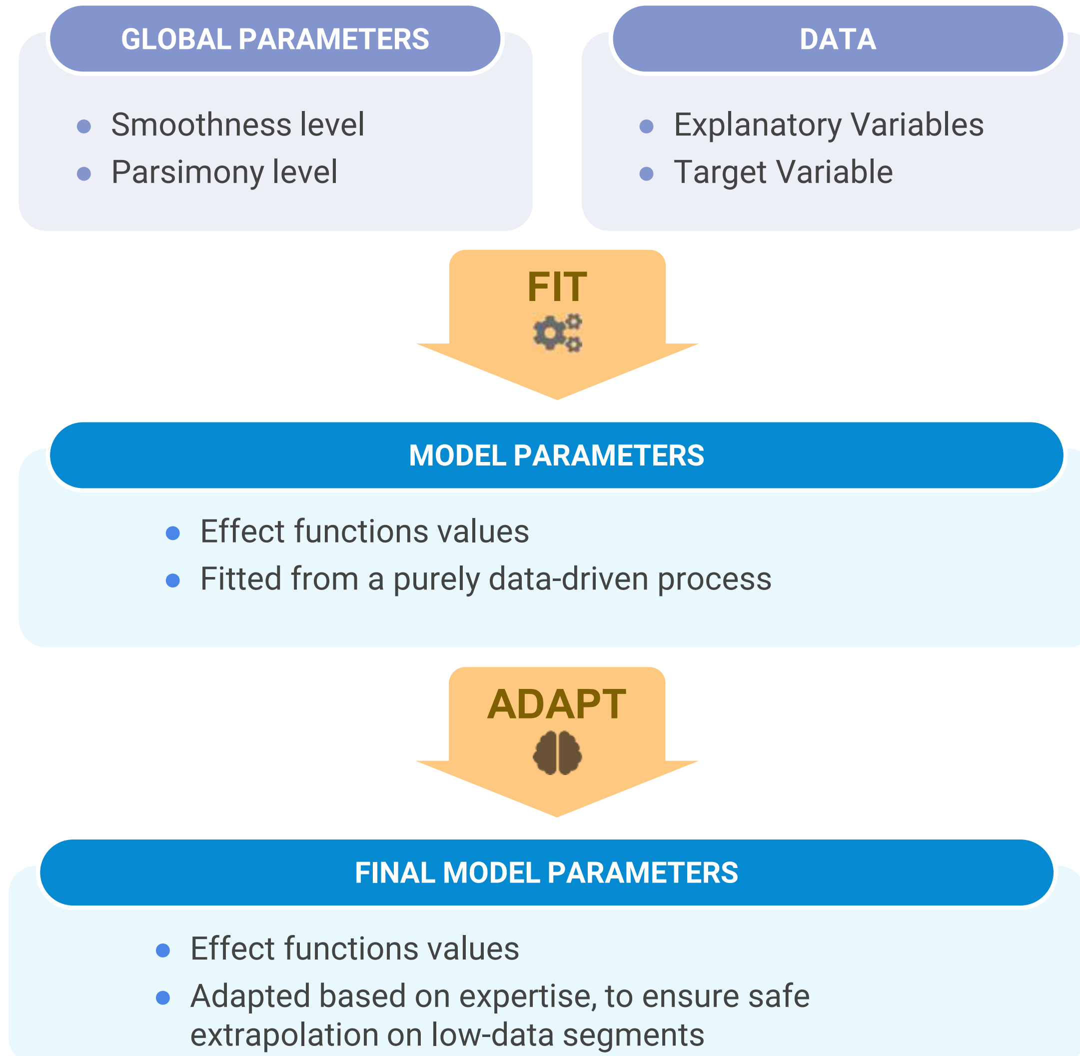
It is possible to directly leverage a model right out of the fit process.

This would be similar to a classic data-science approach.

However, handling transparent models opens the possibility of interacting with them, integrating expert knowledge in the modeling.

So the process is (on purpose) mixing elements of:

- Machine-Learning: **automated fit**, purely **data-driven** model creation, acting on **global parameters** to control overfitting.
- Direct interaction with the models: control of all the **effects** captured in the fitting model, analysis and potentially edition of the **effects** to ensure a good extrapolation of the model.



Conclusion

Mixing Data-Science automation and Actuarial Expertise

ML & Back-test performance

- Allows automated model creation
- Based on statistical criteria
- Easy to measure & reproduce
- Data-driven
- Pushes towards complexity over understanding

+

Actuarial expertise and transparency

- Minimizing the back-test error is not enough
- Performance can't be measured before deployments (and sometimes not even after)
- Direct interactions with the model itself is key to include all the operational constraints.

→

Understanding and capability to interact with a model is key; model's simplicity has value.

Models must allow the inclusion of expertise, safety and provide extrapolation capabilities.

Transparent modeling can and should be combined with machine-learning techniques.

Transparency is not "under-sophistication" or "primitiveness" but realism and efficiency.

Thank you!

Questions?

jan.kuethe@akur8.com

AKUR8