Boosting insights in insurance tariff plans with machine learning methods

Katrien Antonio
LRisk - KU Leuven and ASE - University of Amsterdam

December 3, 2020

My personal website: https://katrienantonio.github.io

Acknowledgement

This talk is based on joint work with

Marie-Pier Côté (Laval, Canada), Roel Henckaerts, and Roel Verbelen,

who work/have worked with me at KU Leuven in the framework of the Ageas research chair on insurance analytics.

This keynote's mission is threefold

To discuss:

(1) specific considerations to keep in mind when using machine learning methods with frequency/severity data

(2) interpretation and comparison tools for machine learning methods, with a particular focus on pricing with frequency/severity data, including different types of risk factors

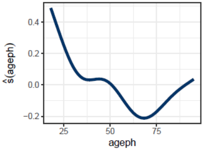
(3) maidrr, our strategy to construct a Model Agnostic Interpretable Data-driven suRRogate.

Road to Explainable AI (XAI)

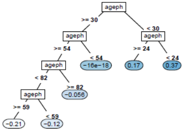
- An explainable AI (XAI) algorithm enables human users to understand, trust and manage its decisions.
- Matters in highly regulated industries, such as insurance and banking.
- Two roads or pathways to XAI:
  - after the event: use interpretation tools to (better) understand decision process in black box model
  - by design: develop and use transparent white box model.

GLMs and GAMs for insurance pricing

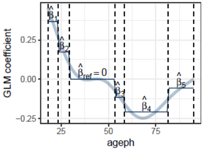
Starting point



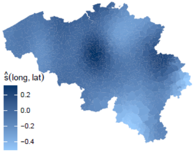
(1a) Smooth continuous effect



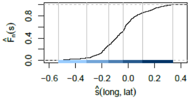
(1b) Supervised decision tree



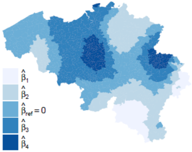
(1c) Binned continuous effect



(2a) Smooth spatial effect

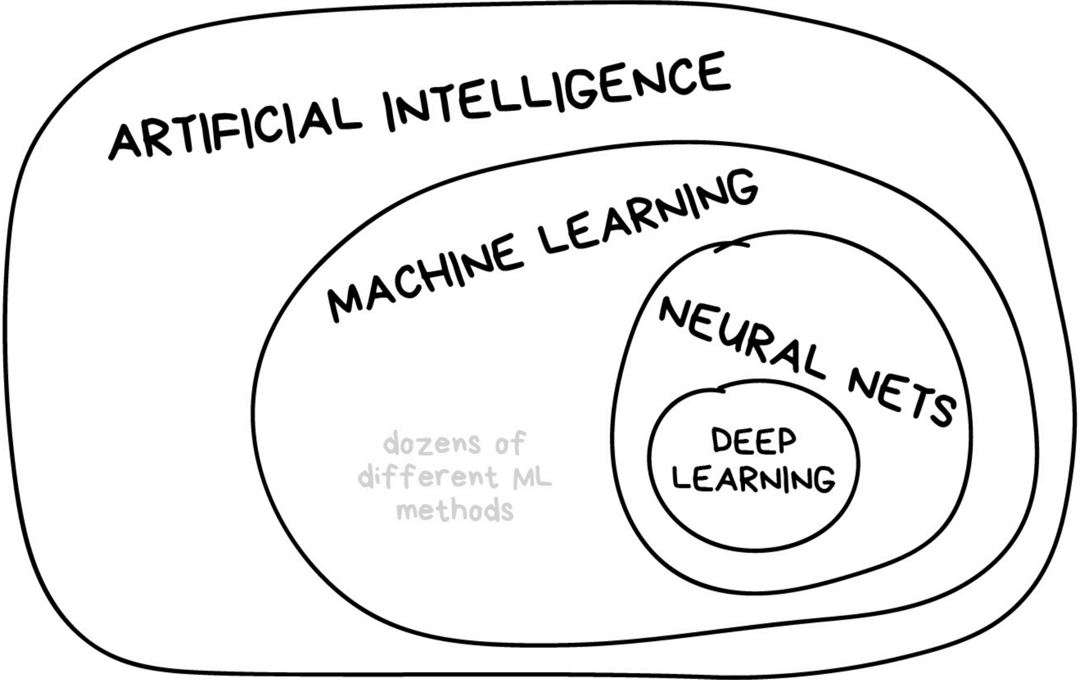


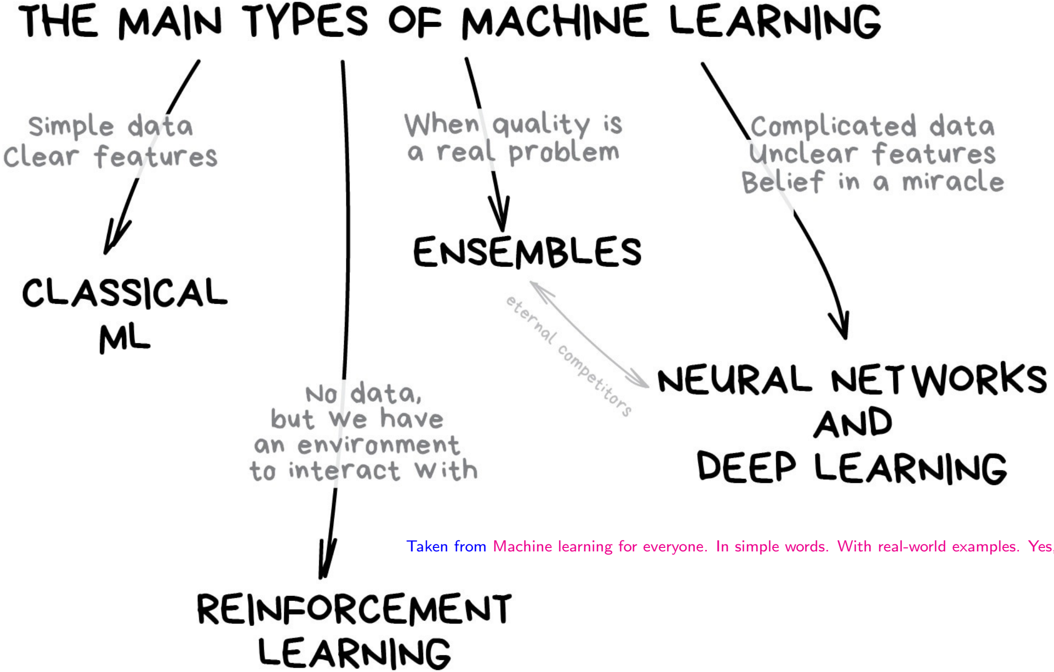
(2b) Unsupervised clustering



(2c) Binned spatial effect

A data driven binning strategy for the construction of insurance tariff classes by Henckaerts, Antonio, Clijsters and Verbelen (2018, Scandinavian Actuarial Journal), with GitHub repo.


ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
NEURAL NETS
DEEP LEARNING
dozens of different ML methods



Taken from Machine learning for everyone. In simple words. With real-world examples. Yes, again.

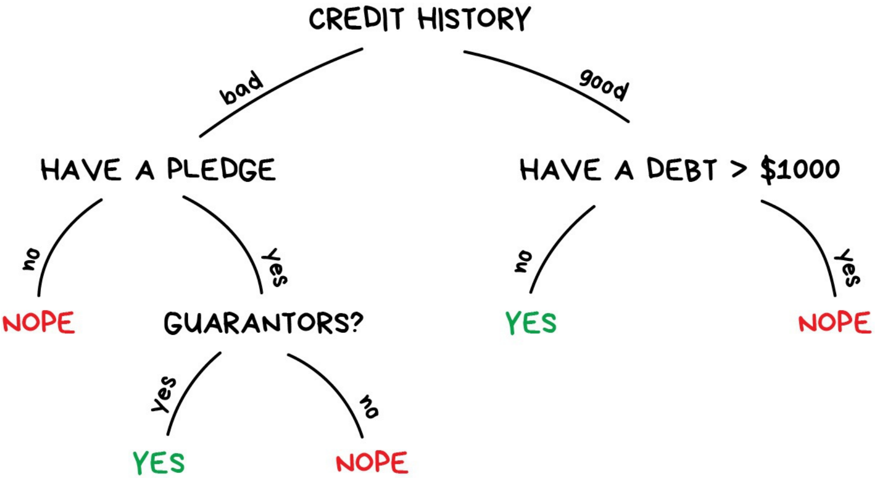
Let's dive into:

Boosting insights in insurance tariff plans with tree-based machine learning methods, by Roel Henckaerts, Marie-Pier Côté, Katrien Antonio and Roel Verbelen (2020, North American Actuarial Journal),

with reproducible examples in notebooks on GitHub:

- tree-based ML
- severity modeling.

GIVE A LOAN?



DECISION TREE

Regression trees

- The process of building a regression tree with CART (Breiman et al., 1984):

  1. divide the predictor space into J distinct, non-overlapping regions $R_1, R_2, \ldots, R_J$

     top-down, greedy strategy with recursive binary splitting

  2. for every observation in region R_j we make the same prediction:

     the mean of the response values for the training observations in R_j.

- The prediction obtained with a regression tree:

$$f_{\text{tree}}(X_1, \ldots, X_p) = \bar{y}_1 I_{\{\boldsymbol{X} \in R_1\}} + \ldots + \bar{y}_J I_{\{\boldsymbol{X} \in R_J\}},$$

where $\bar{y}_j = \text{ave}\left(y_i | \boldsymbol{X}_i \in R_j\right)$.

Regression trees

Tree pruning

- Prune a large tree by minimizing:

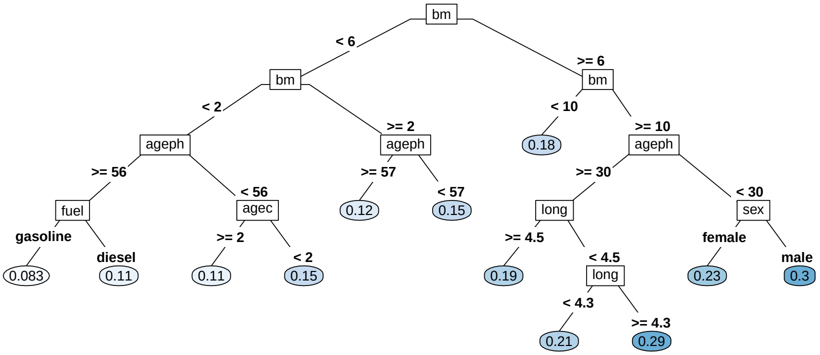
$$\sum_{j=1}^{J} \sum_{i:\boldsymbol{x}_i \in R_j} L(y_i, \hat{y}_{R_j}) \ + \ J \cdot cp \cdot \sum_{i:\boldsymbol{x}_i \in R} L(y_i, \hat{y}_R)$$

  - $cp = 0$ gives biggest possible tree
  - $cp = 1$ gives root tree without splits.

- We employ a tuning strategy and search grid to find the optimal value for cp, e.g. via cross-validation.

Regression trees

Example of a frequency tree



MTPL data set analyzed in Henckaerts et al. (2020, NAAJ).

Loss functions inspired by GLMs

Frequency

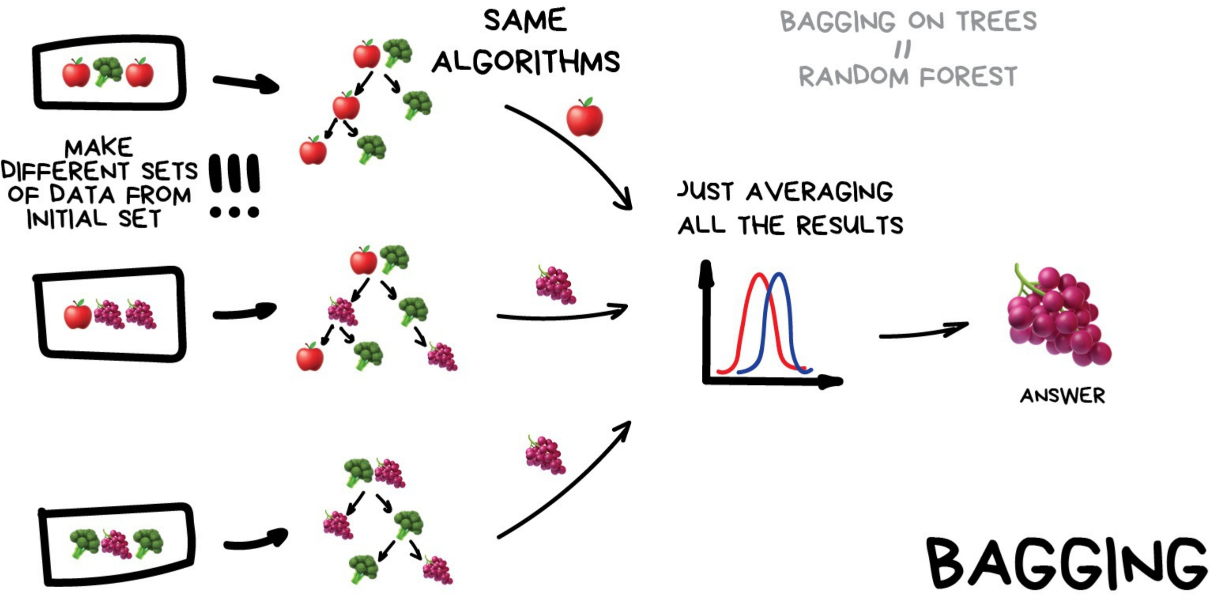
- classic: GLM with *count* distribution (e.g. Poisson or NegBin)
- ML: use *Poisson deviance* as loss function

$$D(\boldsymbol{y}, \hat{f}(\boldsymbol{x})) = 2 \sum_{i=1}^{n} \left(y_i \cdot \ln \frac{y_i}{\hat{f}(\boldsymbol{x}_i)} - (y_i - \hat{f}(\boldsymbol{x}_i)) \right)$$

Severity

- classic: GLM with *skewed* distribution (e.g. Gamma or LogNorm)
- ML: use *Gamma deviance* as loss function

$$D(\boldsymbol{y}, \hat{f}(\boldsymbol{x})) = 2 \sum_{i=1}^{n} w_i \cdot \left(\frac{y_i - \hat{f}(\boldsymbol{x}_i)}{\hat{f}(\boldsymbol{x}_i)} - \ln \frac{y_i}{\hat{f}(\boldsymbol{x}_i)} \right)$$

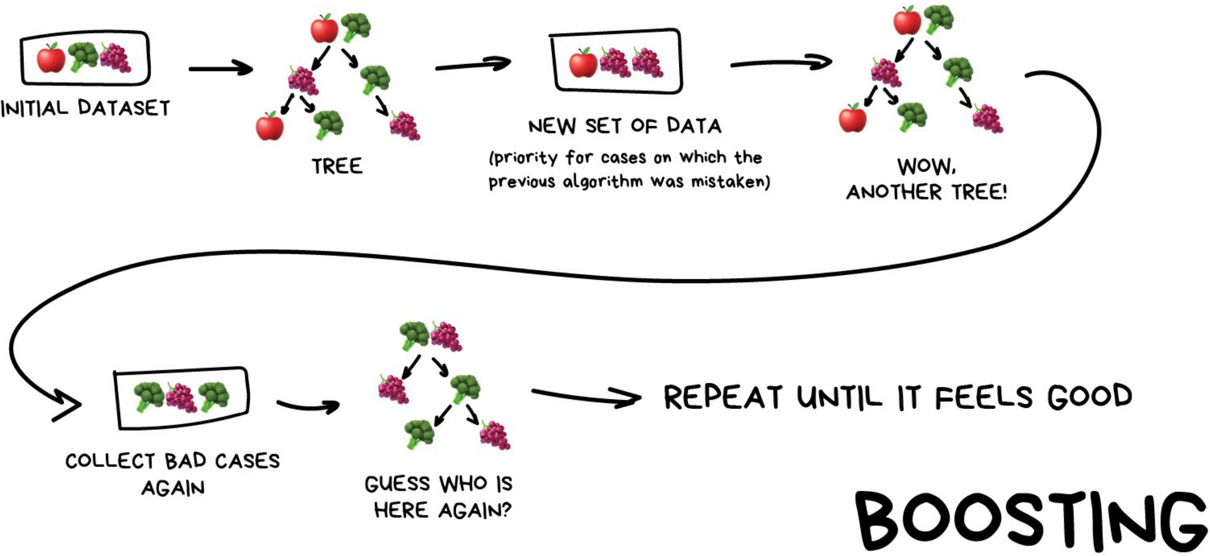

SAME ALGORITHMS
BAGGING ON TREES
=
RANDOM FOREST
MAKE DIFFERENT SETS OF DATA FROM INITIAL SET
!!!
...
JUST AVERAGING ALL THE RESULTS
ANSWER
BAGGING

Bagging

- We generate T different bootstrapped data sets $\{D_t\}_{t=1,\dots,T}$ from the training data D.
- We train our method on the t-th bootstrapped training set and get $\hat{f}_{\text{tree}}(\boldsymbol{x}|D_t)$. Finally, we average all the predictions

$$\hat{f}_{\text{bagg}}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T}\hat{f}_{\text{tree}}(\boldsymbol{x}|D_t).$$

  This is called bootstrap aggregating (or bagging) and goes back to Breiman (1996).
- With random forests each time a split in a tree is considered, a at random m out of p predictors are chosen as split candidates (Breiman, 2001).


INITIAL DATASET
TREE
NEW SET OF DATA
(priority for cases on which the previous algorithm was mistaken)
WOW, ANOTHER TREE!
COLLECT BAD CASES AGAIN
GUESS WHO IS HERE AGAIN?
REPEAT UNTIL IT FEELS GOOD
BOOSTING

Boosting

Boosting trees

- Fitting trees in a forward stagewise procedure, solve:

$$\hat{\Theta}_t = \arg\min_{\Theta_t} \sum_{i=1}^{n} L(y_i, f_{t-1}(\boldsymbol{x}_i) + f_{\text{tree}}(\boldsymbol{x}; \Theta_t)),$$

  where $\Theta_t = \{R_{jt}, b_{jt}\}_1^{J_t}$, the regions and fitted values of the tree.

- With squared-error loss:

  fit a regression tree to the current residuals $y_i - f_{t-1}(\boldsymbol{x}_i)$.

- With other loss functions, this idea generalizes to pseudo-residuals.

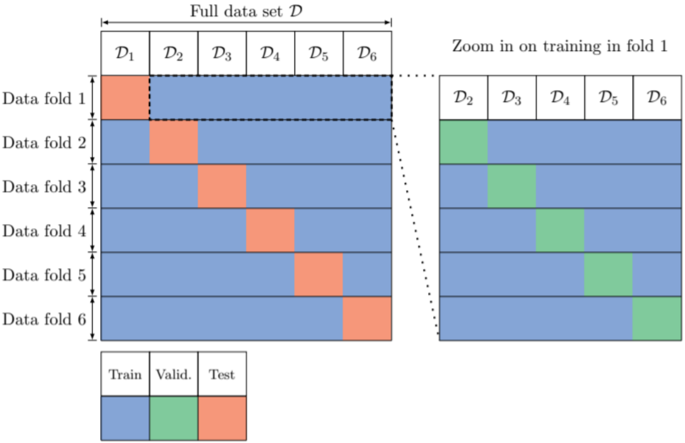
Boosting

The gradient boosting algorithm

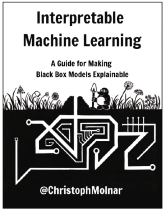
- The gradient tree-boosting algorithm (Friedman, 2001):
  - initializes to the optimal constant model, which is just a single terminal node tree
  - fits a small tree of depth d to the pseudo-residuals $\rho_{it} = -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}$ evaluated at current model fit f_{t-1} (more details in the paper)
  - a shrinkage parameter λ controls the learning speed by shrinking updates $f_{\text{new}}(\boldsymbol{x}) = f_{\text{old}}(\boldsymbol{x}) + \lambda \cdot \text{update}$.
- Stochastic gradient boosting injects randomness in the training process by subsampling the data at random without replacement in each iteration.

Tuning and comparison strategy

Stratified sampling on the MTPL data in Henckaerts et al. (2020)



Interpretation tools



- Classical statistical methods are highly interpretable:
  - coefficients in a GLM
  - smooth effects in a GAM.
- Not the case for machine learning methods:
  - ✓ regression trees
  - ✗ random forests
  - ✗ boosted trees.
- There is a need for interpretation tools: look under the hood!

Interpretation tools

PDPs

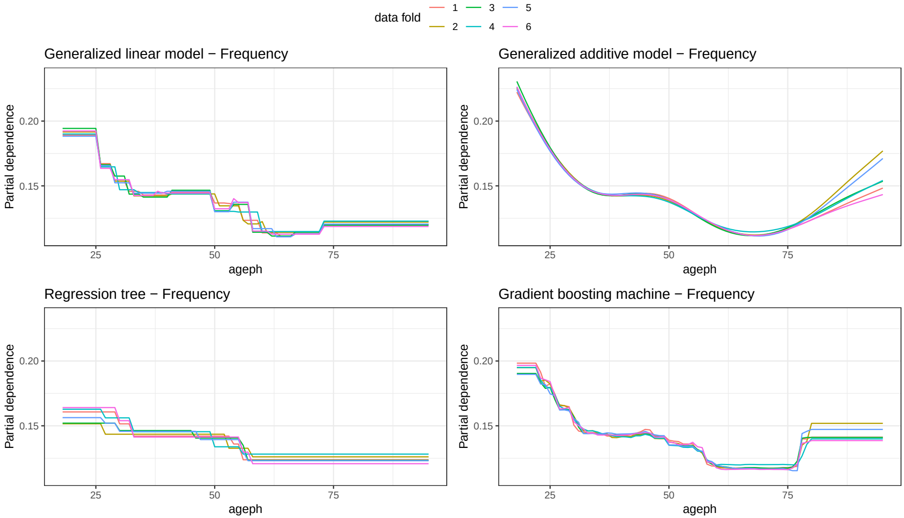
- (Univariate) Partial Dependence Plots (PDPs) to interpret the marginal effect of a feature on the outcome

$$\bar{f}_{\ell}(x_{\ell}) = \frac{1}{n}\sum_{i=1}^{n} f_{\text{model}}(x_{\ell}, \boldsymbol{x}^{i}_{-\ell}).$$

- Global measure such that interaction effects can stay hidden.

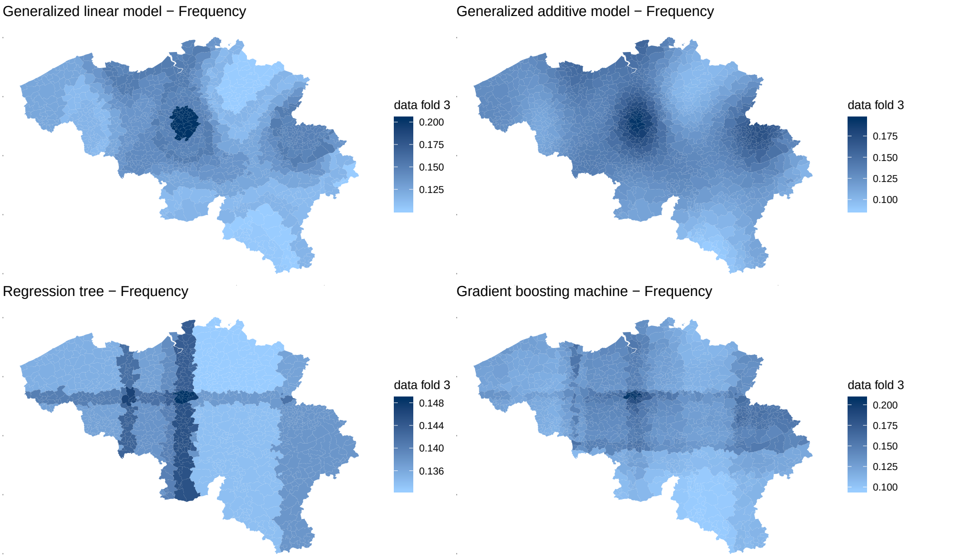
Interpretation tools

PDPs



Interpretation tools

PDPs



Interpretation tools

ICEs

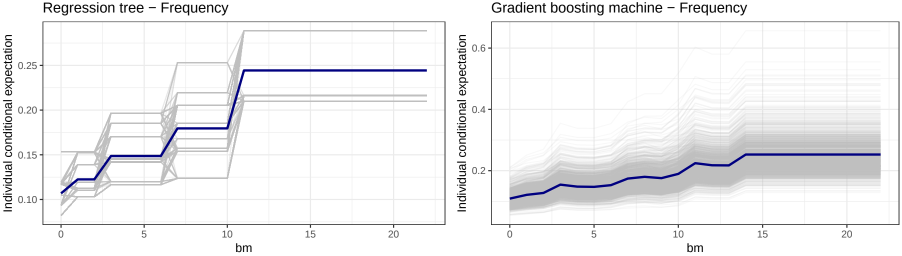
- Individual conditional expectation plots (ICEs)

$$\tilde{f}_{\ell,i}(x_\ell) = f_{\text{model}}(x_\ell, \boldsymbol{x}^i_{-\ell}).$$

- ICEs show the effect of a variable on an individual level:
  - to picture the uncertainty of the effect of a variable on the prediction outcome
  - to detect interaction effects.

Interpretation tools

ICEs



The gray lines are ICEs for 1000 random policyholders and the blue line shows the partial dependence curve.

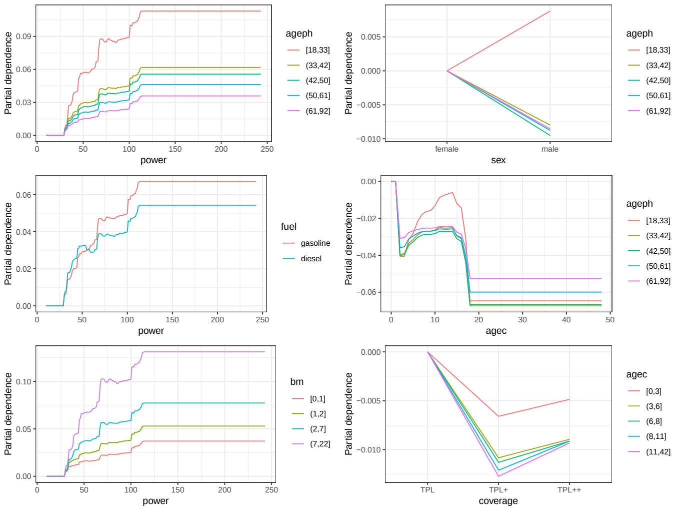
Hunting for interaction effects

Friedman's H-statistic:

$$H_{k\ell}^2 = \frac{\sum_{i=1}^n \{\bar{f}_{kl}(x_k^{(i)}, x_\ell^{(i)}) - \bar{f}_k(x_k^{(i)}) - \bar{f}_l(x_\ell^{(i)})\}^2}{\sum_{i=1}^n \bar{f}_{kl}^2(x_k^{(i)}, x_\ell^{(i)})}.$$

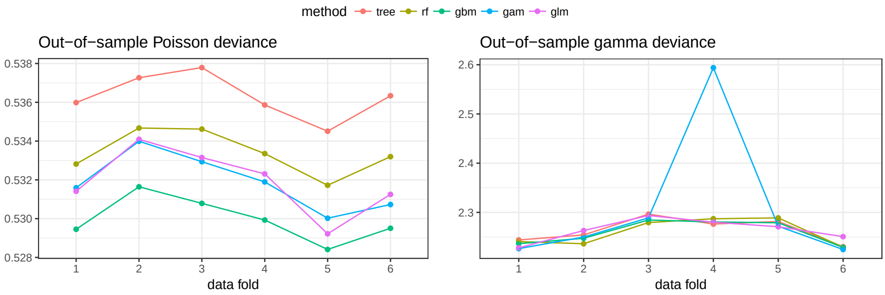
Variables	H-statistic	Variables	H-statistic	Variables	H-statistic
(`lat`, `long`)	0.2687	(`agec`, `coverage`)	0.1185	(`bm`, `power`)	0.0800
(`fuel`, `power`)	0.1666	(`ageph`, `power`)	0.1062	(`ageph`, `lat`)	0.0799
(`agec`, `power`)	0.1319	(`ageph`, `bm`)	0.0961	(`agec`, `ageph`)	0.0785
(`ageph`, `sex`)	0.1293	(`power`, `sex`)	0.0829	(`long`, `sex`)	0.0732
(`coverage`, `long`)	0.1203	(`fuel`, `long`)	0.0828	(`agec`, `bm`)	0.0678

PDPs to picture interaction effects



Model comparison tools

Findings: out-of-sample



Conclusion:

- Poisson deviance for frequency: GBM > GLM > RF > CART
- gamma deviance for severity: GBM ≈ GLM ≈ RF ≈ CART.

Model comparison tools

Findings: model lift

- We combine frequency and severity into a (technical) tariff or risk premium.
- We compare the GLM, GAM, decision tree, random forest and GBM constructed tariffs.
- Managerial tools:
  - total loss vs. total premiums
  - loss ratio lift, double lift, Gini index.
- Conclusion: GBM > GLM > RF > CART.

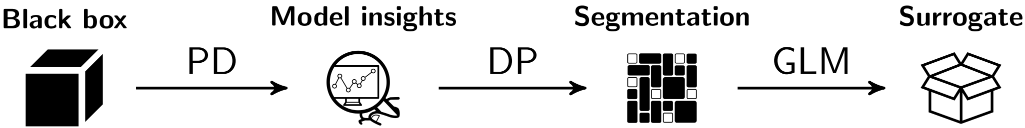
maidrr: Model-Agnostic Interpretable Data-driven suRRogate

How about using the GBM to inform feature engineering for a GLM?

In fact, the GBM could be replaced by any ML method (RF, NN, etc.).

We developed maidrr, see the working paper on arxiv by Henckaerts, Antonio and Côté.

maidrr: Model-Agnostic Interpretable Data-driven suRRogate



(1) group values of feature x_j based on univariate PD $\bar{f}_j(x_j) \Rightarrow$ use dynamic programming (DP) algo for clustering

(2) find relevant interactions x_a and x_b based on H-statistic

(3) cluster similar $\bar{f}_{a,b}(x_a, x_b) \Rightarrow$ use DP

(4) fit GLM on segmented features.

maidrr: Model-Agnostic Interpretable Data-driven suRRogate

Some technical details

Values of feature x_j with a similar PD show a similar relation to the prediction target

- group values/levels of x_j based on the univariate PD $\bar{f}_j(x_j)$
- let m_j denote the unique number of observed values for x_j
- let $x_{j,q}$ denote its qth value for $q \in \{1, \ldots, m_j\}$ and define $z_{j,q} = \bar{f}_j(x_{j,q})$
- allocate elements of m_j dimensional input vector to k_j clusters by minimizing within-cluster sum of squares.

Adjacency constraints can be imposed (e.g. for ordinal variables).

maidrr: Model-Agnostic Interpretable Data-driven suRRogate

Some technical details (cont.)

We choose the number of groups (k_j) for feature x_j via a penalized loss function:

$$\frac{1}{m_j}\sum_{q=1}^{m_j}\left(z_{j,q} - \tilde{z}_{j,q}\right)^2 + \lambda_{\text{marg}} \cdot \log(k_j)$$

with $\tilde{z}_{j,q}$ the average PD effect for the group to which value/level $x_{j,q}$ belongs.

λ_{marg} is a tuning parameter, independent of j.

maidrr: Model-Agnostic Interpretable Data-driven suRRogate

Some technical details (cont.)

The interaction between features x_a and x_b is captured by subtracting both one-dimensional PDs from the two-dimensional PD:

$$\bar{f}_{a,b}(x_a, x_b) = \frac{1}{n}\sum_{i=1}^{n} f_{\text{pred}}(x_a, x_b, \boldsymbol{x}^i_{-a,-b}) - \frac{1}{n}\sum_{i=1}^{n}\sum_{\ell \in \{a,b\}} f_{\text{pred}}(x_\ell, \boldsymbol{x}^i_{-\ell}).$$

DP algorithm without adjacency constraint allows to cluster similar $\bar{f}_{a,b}(x_a, x_b)$ values.

Optimal number of groups is again chosen using a penalized loss, with separate tuning parameter λ_{intr} for the interaction effects.

Comparison tools

Other surrogates, accuracy, local interpretations

The paper reports our findings on a benchmark study with 6 insurance data sets.

Decision tree (DT) surrogate

- original data as features and the GBM predictions as target
- maximum tree depth restricted to four.

Linear model (LM) surrogate

- original data as features and the GBM predictions as target.

Comparison tools

Other surrogates, accuracy, local interpretations

	ausprivauto	bemtpl	frempl	fremtpl	norauto	pricingame	avg.
GLM	**0.10**	**0.49**	**1.80**	**0.92**	**0.03**	**0.48**	**0.64**
LM	0.22	1.15	18.39	6.35	0.07	2.53	4.79
DT	0.25	1.68	4.82	2.66	0.28	2.13	1.97

$$\Delta D^{\mathrm{Poi}} = 100 \times \left(D^{\mathrm{Poi}}\{y, f_{\mathrm{surro}}(\boldsymbol{x})\} / D^{\mathrm{Poi}}\{y, f_{\mathrm{gbm}}(\boldsymbol{x})\} - 1 \right).$$

Comparison tools

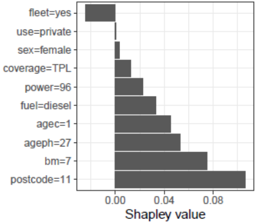
Other surrogates, accuracy, local interpretations

	ausprivauto	bemtpl	frempl	fremtpl	norauto	pricingame	avg.
GLM	0.86	**0.94**	**0.91**	**0.78**	**0.99**	**0.93**	**0.90**
LM	**0.89**	0.83	0.62	0.30	0.95	0.88	0.75
DT	0.75	0.74	0.88	0.75	0.84	0.76	0.78

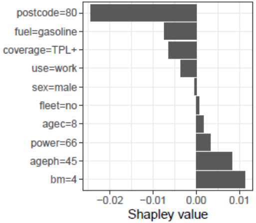
$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left\{ f_{\text{surro}}(\boldsymbol{x}_i) - f_{\text{gbm}}(\boldsymbol{x}_i) \right\}^2}{\sum_{i=1}^{n} \left\{ f_{\text{gbm}}(\boldsymbol{x}_i) - \mu_{\text{gbm}} \right\}^2}.$$

Comparison tools

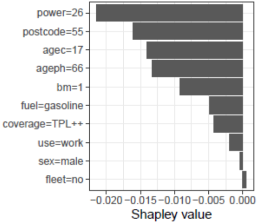
Other surrogates, accuracy, local interpretations



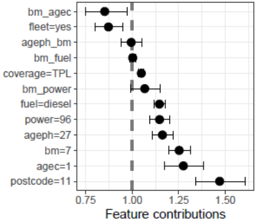
(a) GBM: high risk



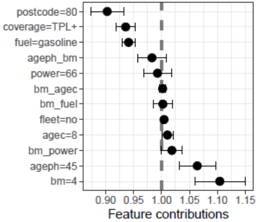
(b) GBM: medium risk



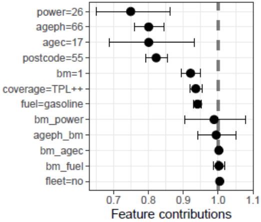
(c) GBM: low risk



(d) GLM: high risk



(e) GLM: medium risk



(f) GLM: low risk

(Provocative) Statement & Thanks!

The mindset of the actuary - research ambition

> The narrative must be that actuaries are entering the data science world not entirely to compete but also to bring the element of the actuarial profession where we build integrity and transparency into any work that we do, and how documentation of that is possible.

Quote from What data science means for the future of the actuarial profession, British Actuarial Journal, June 2018.

R packages

R packages developed by Roel Henckaerts (as part of his PhD):

- for `distRforest` see https://henckr.github.io/distRforest/
- for `maidrr` see https://henckr.github.io/maidrr

References and acknowledgements

- For detailed list of references, please consult the papers.
- Visuals are from
  Machine learning for everyone. In simple words. With real-world examples. Yes, again.
- More interpretation tools available in the (online) book by Christophe Molnar, see `https://christophm.github.io/interpretable-ml-book/`.

Appendix

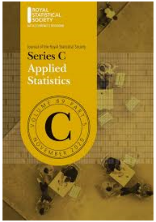
Regularized GLMs for insurance pricing

Sparse Regression with MUlti-type Regularized Feature modelling by Devriendt, Antonio, Reynkens & Verbelen (2020, Insurance: Mathematics and Economics)

- automatic feature selection and binning of risk factors via regularization (i.e. lasso and friends)
- R package `smurf` on CRAN
- end product is a GLM!

Appendix

GLMs and GAMs for telematics insurance pricing



Unravelling the predictive power of telematics data in car insurance pricing by Verbelen, Antonio & Claeskens (2018, JRSS C)

- black box collected data on group of young drivers
- compositional data ('parts of a whole') on kilometers driven across road types, time slots
- GAMs for claim frequencies, with specific attention to effects of compositional data and interpretation.

Appendix

Gradient Boosting Machines (GBMs)

initialize fit to the optimal constant model: $f_0(\boldsymbol{x}) = \arg\min_b \sum_{i=1}^{n} \mathscr{L}(y_i, b)$;
for $t = 1, \ldots, T$ **do**
randomly subsample data of size $\delta \cdot n$ without replacement from data $\mathcal{D}$;
for $i = 1, \ldots, \delta \cdot n$ **do**

$$\rho_{i,t} = -\left[\frac{\partial \mathscr{L}\{y_i, f(\boldsymbol{x}_i)\}}{\partial f(\boldsymbol{x}_i)}\right]_{f=f_{t-1}}$$

fit a tree of depth d to the pseudo-residuals $\rho_{i,t}$ resulting in regions $R_{j,t}$ for $j = 1, \ldots, J_t$;
for $j = 1, \ldots, J_t$ **do**

$$\hat{b}_{j,t} = \arg\min_b \sum_{i:\boldsymbol{x}_i \in R_{j,t}} \mathscr{L}\{y_i, f_{t-1}(\boldsymbol{x}_i) + b\}$$

update $f_t(\boldsymbol{x}) = f_{t-1}(\boldsymbol{x}) + \lambda \sum_{j=1}^{J_t} \hat{b}_{j,t} \mathbb{1}(\boldsymbol{x} \in R_{j,t})$;
$f_{\text{gbm}}(\boldsymbol{x}) = f_T(\boldsymbol{x})$;

Algorithm 2: Procedure to build a (stochastic) gradient boosting machine.

Appendix

Tuning and hyper-parameters

	Tuning parameters	Hyper-parameters
Regression tree	complexity parameter cp coefficient of variation gamma prior γ	$\kappa = 0.01$
Random forest	number of trees T number of split candidates m	$cp = 0 \quad \gamma = 0.25$ $\kappa = 0.01 \quad \delta = 0.75$
Gradient boosting machine	number of trees T tree depth d	$\lambda = 0.01$ $\kappa = 0.01 \quad \delta = 0.75$

Appendix

maidrr algorithm in pseudo code

Algorithm 1 maidrr

Input: data, f_{pred}, λ_{marg}, λ_{intr}, k and h

for $j = 1$ **to** p **do**

  calculate the PD effect $\bar{f}_j$ via Eq. (1)

  apply the DP algorithm to feature x_j with $k_j^* = \underset{k_j \in \{1,\ldots,k\}}{\arg\min}$ Eq. (2) for $\lambda = \lambda_{\text{marg}}$

  x_j^c represents the grouped version of x_j in categorical format with k_j^* groups

end for

feature selection: $F = \{j \mid k_j^* > 1\}$

upfront interaction selection: $I = \{(l, m) \mid l \in F$ **and** $m \in F$ **and** $H(x_l, x_m) \geq h\}$

for all (a, b) **in** I **do**

  calculate the PD effect $\bar{f}_{a,b}$ via Eq. (3)

  apply the DP algorithm to interaction (x_a, x_b) with $k_{ab}^* = \underset{k_j \in \{1,\ldots,k\}}{\arg\min}$ Eq. (2) for $\lambda = \lambda_{\text{intr}}$

  $x_{a:b}^c$ represents the grouped version of $x_{a:b}$ in categorical format with k_{ab}^* groups

end for

interaction selection: $I = I \setminus \{(l, m) \mid k_{lm}^* = 1\}$

fit a GLM to the target with features x_j^c for $j \in F$ and interactions $x_{a:b}^c$ for $(a, b) \in I$

Output: surrogate GLM
