# An introduction to insurance pricing models and in particular tree-based machine learning models

Julie Huyghe

20 June 2022

ULB UNIVERSITÉ LIBRE DE BRUXELLES

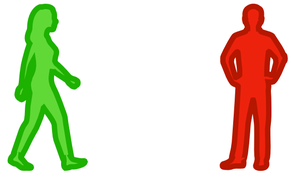
Julien Trufin
Department of Mathematics
Université Libre de Bruxelles

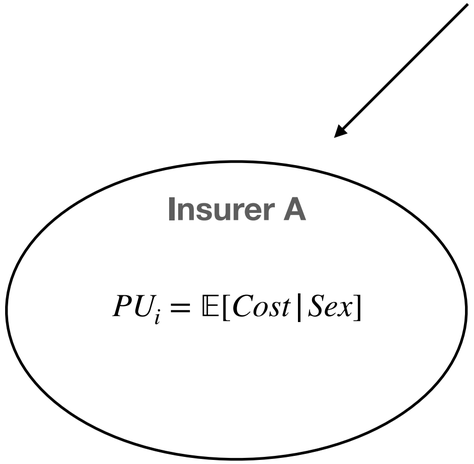
Michel Denuit
Institute of Statistics, Biostatistics and Actuarial Science
UCLouvain

# Pricing in Insurance

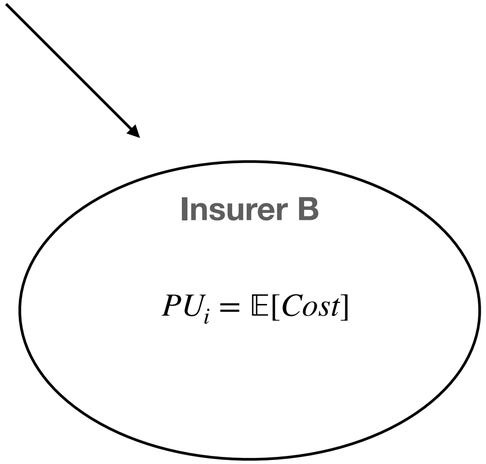
## Why Classifying Risks ?

| | |
|---|---|
| $\mathbb{E}\left[\text{cost} \mid \text{Female}\right]$ | **50** |
| $\mathbb{E}\left[\text{cost} \mid \text{Male}\right]$ | **150** |
| $\mathbb{E}\left[\text{cost}\right]$ | **100** |






SEGMENTATION



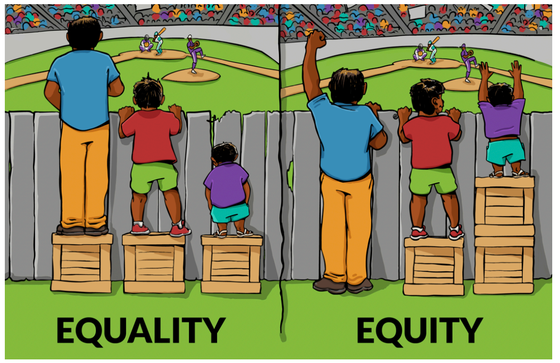

MUTUALISATION

WHICH ONE IS THE BEST MODEL ?

ULB

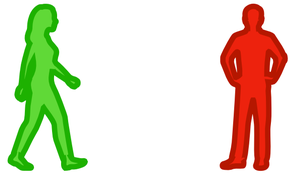
# Pricing in Insurance

## Why Classifying Risks ?




### Philosophical point of view

| | |
|---|---|
| $\mathbb{E}\left[\text{cost} \mid \text{Female}\right]$ | 50 |
| $\mathbb{E}\left[\text{cost} \mid \text{Male}\right]$ | 150 |
| $\mathbb{E}\left[\text{cost}\right]$ | 100 |



**Insurer A**

$PU_i = \mathbb{E}[Cost \mid Sex]$

**Insurer B**

$PU_i = \mathbb{E}[Cost]$

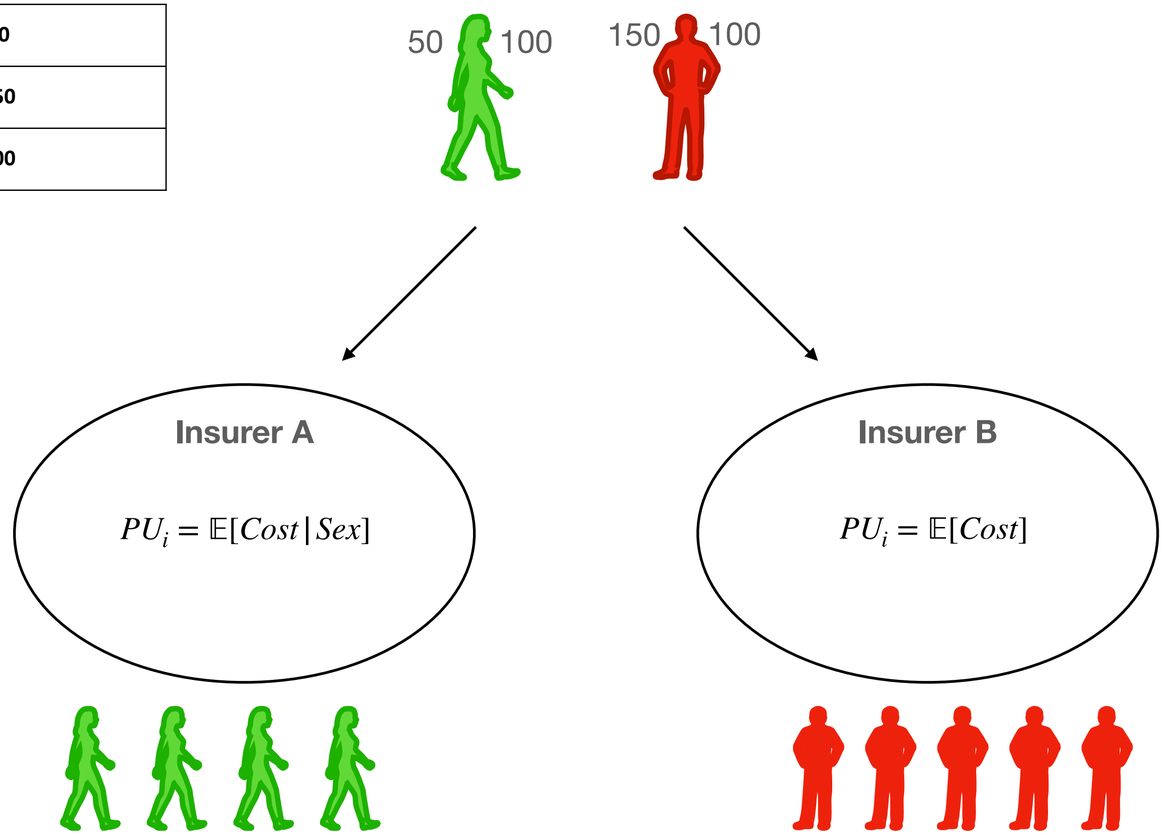
**EGALITY** **VS** **EQUITY/FAIRNESS**

# Pricing in Insurance

## Why Classifying Risks ?

### Insurer's point of view

| | |
|---|---|
| $\mathbb{E}\left[\text{cost} \mid \text{Female}\right]$ | **50** |
| $\mathbb{E}\left[\text{cost} \mid \text{Male}\right]$ | **150** |
| $\mathbb{E}\left[\text{cost}\right]$ | **100** |




**ADVERSE SELECTION PHENOMENON**

# Pricing in Insurance

Let $Y \in \mathbb{R}^m$ be a response vector and $\mathbf{X} \in \mathbb{R}^{m \times p}$ a vector of features

The aim of actuarial ratemaking is to evaluate the pure premium as accurately as possible.
The target is thus the conditional expectation :

$$\mu(\mathbf{X}) = \mathbb{E}[Y \,|\, \mathbf{X}]$$

Generally unknown

$$\hat{\mu}(\mathbf{X})$$

Lot of data
Lot of features

Supervised Learning

ULB

# Pricing in Insurance

## Dataset

$$\begin{pmatrix} y^{(1)} & x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & \cdots & x_p^{(1)} \\ y^{(2)} & x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & \cdots & x_p^{(2)} \\ \vdots & \cdots & \cdots & \cdots & \cdots & \vdots \\ y^{(n)} & x_1^{(n)} & x_2^{(n)} & x_3^{(n)} & \cdots & x_p^{(n)} \end{pmatrix}$$

## Target Vector

$$\mathbf{Y} = \begin{pmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{pmatrix}$$

## Features Matrix

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \cdots & x_p^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(n)} & \cdots & x_p^{(n)} \end{pmatrix}$$

ULB

# Pricing in Insurance

## Goal

$\hat{\pi} : \mathbf{X} \rightarrow Y$ that minimizes the expectation of some loss function $L(Y, \hat{\pi})$

over the joint distribution of all $(Y, \mathbf{X})$ – values, i.e ;

$$\hat{\pi}(\mathbf{X}) = \hat{E}[Y \,|\, \mathbf{X}] = \arg\min_{\pi(\mathbf{X})} E[L(Y, \pi(\mathbf{X}))]$$

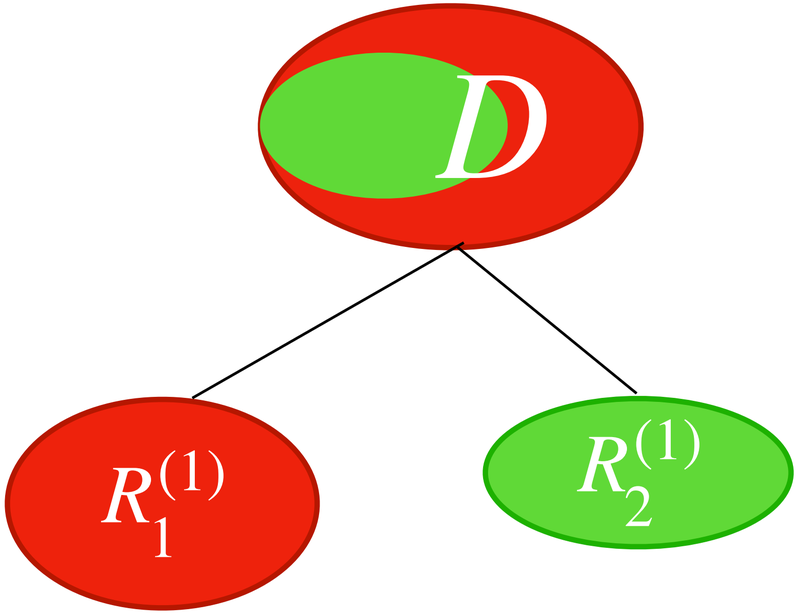
$$\approx \arg\min_{\pi(\mathbf{x})} \frac{1}{n} \sum_{i=1}^{n} L(y_i, \pi(\mathbf{x}_i)) = \arg\min_{\pi(\mathbf{x})} \sum_{i=1}^{n} L(y_i, \pi(\mathbf{x}_i))$$

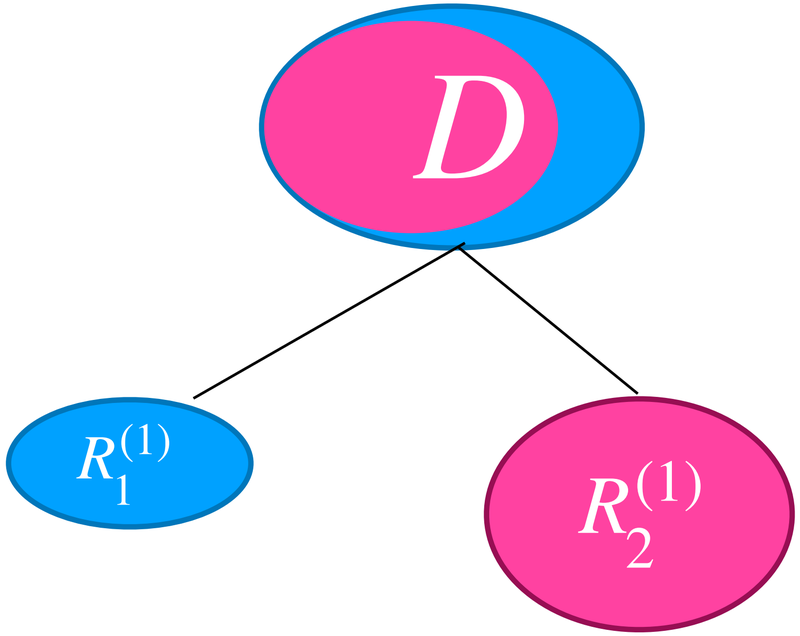
# Regression trees

# Regression trees

## Idea

To begin, the idea is to partition the predictor space into two regions : $R_1^{(1)}$ and $R_2^{(1)}$




How to choose $R_1^{(1)}$ and $R_2^{(1)}$?

Wich prediction in each of $R_1^{(1)}$ and $R_2^{(1)}$?

ULB

# Regression trees

## Idea

Training set $= \{(y_i, \mathbf{x}_i); i = 1, \cdots, n\}$ denoted $\mathbf{D}$.

Search every distinct value of every predictor to find **the predictor** and **split value** that partitions the predictor space into two regions :

$R_1^{(1)}$ and $R_2^{(1)}$ such that

$$\sum_{i=1}^{n} L(y_i, \pi(\mathbf{x}_i)) = \sum_{i:\mathbf{x}_i \in R_1^{(1)}} L(y_i, \bar{c}_1^{(1)}) + \sum_{i:\mathbf{x}_i \in R_2^{(1)}} L(y_i, \bar{c}_2^{(1)})$$

$$\bar{c}_1^{(1)} = \text{ave}\left(y_i \,|\, \mathbf{x}_i \in R_1^{(1)}\right)$$

$$\bar{c}_2^{(1)} = \text{ave}\left(y_i \,|\, \mathbf{x}_i \in R_2^{(1)}\right)$$

$$\pi(\mathbf{x}_i) = \bar{c}_1^{(1)} I_{\{\mathbf{x}_i \in R_1^{(1)}\}} + \bar{c}_2^{(1)} I_{\{\mathbf{x}_i \in R_2^{(1)}\}}$$

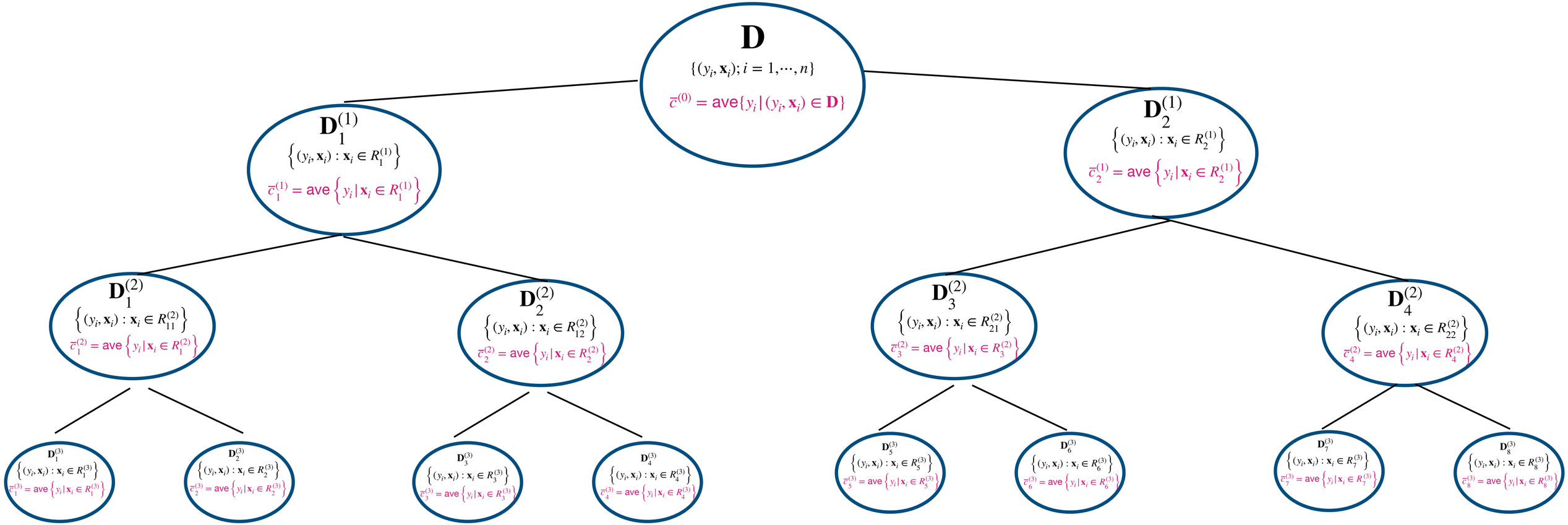
→ The training set $\mathbf{D}$ is then partitioned into two groups $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$, where

$$\mathbf{D}^{(1)} = \left\{(y_i, \mathbf{x}_i) : \mathbf{x}_i \in R_1^{(1)}\right\} \qquad \mathbf{D}^{(2)} = \left\{(y_i, \mathbf{x}_i) : \mathbf{x}_i \in R_2^{(1)}\right\}$$

ULB

# Regression trees

## Algorithm




$$\hat{\pi}(\mathbf{x}) = \sum_{i=1}^{M} \hat{\bar{c}}_m I_{\{\mathbf{x} \in R_m\}}$$

Here $M = 2^3 = 8$

ULB

# Regression trees

## Algorithm

Repeat the previous step within each of groups $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ and so on

**Stop**

- When the number of samples in the split falls below some threshold
- When the depth of the tree reached a certain limit
- When the improvement in the loss function becomes too small
- Pruning (with complexity parameter)

…

$$\hat{\pi}_{tree}(\mathbf{x}) = \sum_{m=1}^{M} \hat{\bar{c}}_m I_{\{\mathbf{x} \in R_m\}}$$

ULB

# Regression trees

## Advantages & Disadvantages

### Advantages

- Easy to interpret
- Intuitively, the most important predictors are those that appear
  - → Higher in the tree ;
  - → Several times in the tree.
- Do not require to specify the form of the regression function (Non-parametric regression)

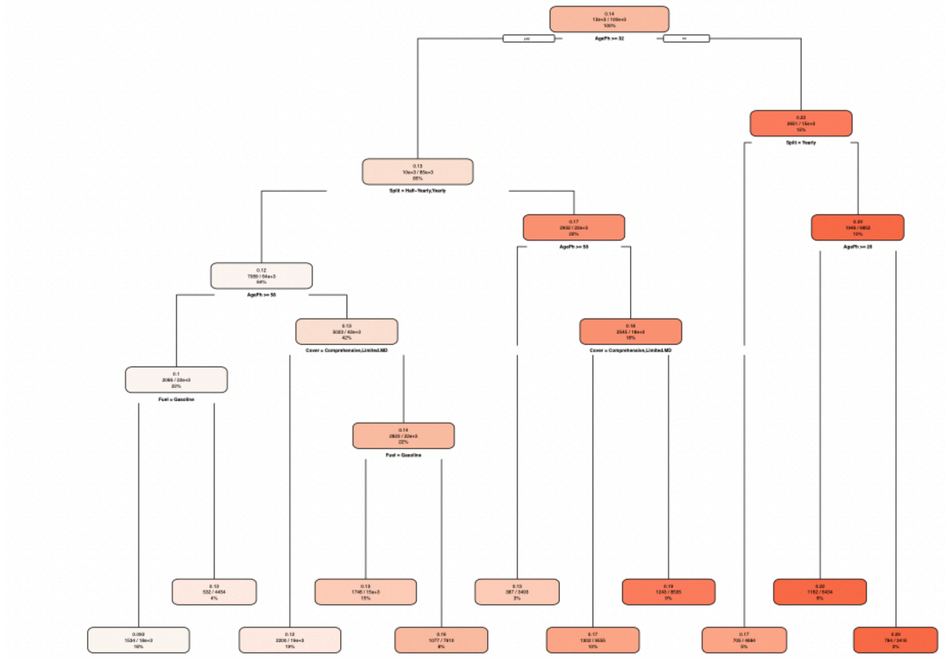
### Disadvantages

- The final tree can be very large and overfit the training set
- Instability of the Model (slight changes in the data can drastically change the structure of the tree).
- Finite number of possible predicted outcomes (determined by number of terminal nodes).
- Partition the data in rectangular regions of the predictor space (not optimal predictive performance)

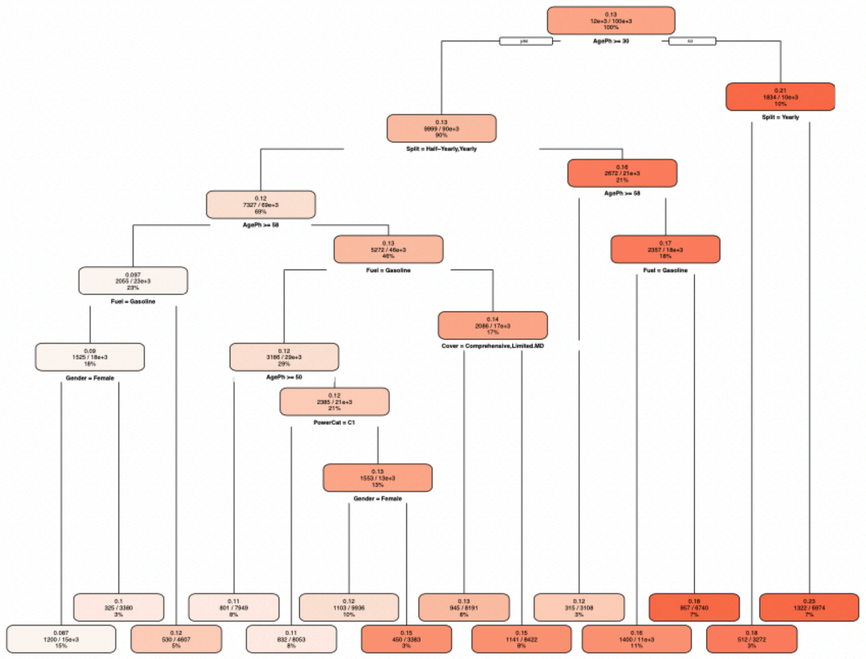
ULB

# Regression trees

## Advantages & Disadvantages



data1 = data[1:100000,]



data2 = data[10000:110000,]

**90%  of the observations are the same, but structure of de tree is completely different !**

# Bagging & Random Forest

# Bagging & Random Forest

## Bagging

For t = 1 to T :

1) Generate a Bootstrap D_t sample of the original data
2) Fit a tree on this Sample D_t
3) Give a prediction : $\hat{\pi}(\mathbf{x})$

$$\hat{\pi}_{bag}(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} \hat{\pi}_t(\mathbf{x})$$

➡ Aggregating bootstrap trees reduces the variance in the prediction and hence makes the prediction more stable.

➡ A bagged model is less interpretable than a model that is not bagged.
A bagged tree is no longer a tree.

## Random Forest

For t = 1 to T :

1) Generate a Bootstrap D_t sample of the original data
2) Fit a tree such that :
   For each node :
   - Select randomly m predictors
   - Choose de best predict amoung theses m predictors
   - Split the node into two daughters node

3) Give a prediction :

$$\hat{\pi}_{rf}(\mathbf{x}) = \frac{1}{T}\sum_{i=1}^{T} \hat{\pi}_t(\mathbf{x})$$

Reducing m reduce the correlation between any pair of trees
Reduce the variance of $\hat{\pi}_{rf}(\mathbf{x})$

ULB

# Boosting & Gradient Boosting

# Boosting & Gradient Boosting

## Boosting Methods

Several trees are also used. But this is not the same philosophy as bagging or RF.

Each tree will explain what is not yet explained.

**Boosting Methods** combining many weak rules to approximate

A weak learner is a learning algorithm which is slightly better than random guessing.

**A weak learner can be constructed by a small regression tree !**

ULB

# Boosting & Gradient Boosting

## Additive models & Boosting

The **standard linear regression model** assumes a linear form for the conditional expectation $E[y \mid \mathbf{x}]$

$$E[y \mid \mathbf{x}] \approx \hat{\pi}(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$$

An **additive model** extends the linear model, by assuming an additive predictor of the form :

$$E[y \mid \mathbf{x}] \approx \hat{\pi}(\mathbf{x}) = \sum_{j=1}^{p} f_j(x_j)$$

The **additive model can be generalized** by considering functions of potentially all the inputs variables

$$E[y \mid \mathbf{x}] \approx \hat{\pi}(\mathbf{x}) = \sum_{t=1}^{T} f_t(\mathbf{x}) = \sum_{t=1}^{T} \beta_{t,a_t} h(a_t, \mathbf{x})$$

**Estimation of the parameters** amounts to solve

$$\min_{\{\beta_{t,\mathbf{a}_t},\mathbf{a}_t\}_1^T} \sum_{i=1}^{n} L\left(y_i, \sum_{t=1}^{T} \beta_{t,\mathbf{a}_t} h(\boldsymbol{x}_i; \mathbf{a}_t)\right).$$

ULB

# Boosting & Gradient Boosting

## Boosting

### Boosting

Initialize $\pi_0(\mathbf{x}) = 0$ (for example)

For $t = 1$ to $T$, do

1) Estimate $\beta_{t,\mathbf{a}_t}$ and $\mathbf{a}_t$ by minimizing :

$$\sum_{i=1}^{n} L(y_i, \pi_{t-1}(\mathbf{x}_i) + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t))$$

2) Update $\pi(\mathbf{x}) = \pi_{t-1} + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t)$

Output : $\hat{\pi}(\mathbf{x}) = \pi_T(\mathbf{x})$

### Remark

If loss function = squared error

$$\sum_{i=1}^{n} L(y_i, \pi_{t-1}(\mathbf{x}_i) + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t)) = \left( \underbrace{y_i - \pi_{t-1}(\mathbf{x}_i))}_{r_{i,t-1}} - \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t) \right)^2$$

$r_{i,t-1}$ equals the residuals for observation $i$ at the iteration $t-1$

# Boosting & Gradient Boosting

## Boosting

## Boosting

Initialize $\pi_0(\mathbf{x}) = 0$ (for example)

For $t = 1$ to $T$, do

1) Estimate $\beta_{t,\mathbf{a}_t}$ and $\mathbf{a}_t$ by minimizing :

$$\sum_{i=1}^{n} L(y_i, \pi_{t-1}(\mathbf{x}_i) + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t))$$

Solution to step (2.1) in Algorithm 1 can be difficult to obtain in some cases.

2) Update $\pi(\mathbf{x}) = \pi_{t-1} + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t)$

The gradient boosting algorithm aims to solve this issue by using a two-step procedure for step (2.1).

Output : $\hat{\pi}(\mathbf{x}) = \pi_T(\mathbf{x})$

# Boosting & Gradient Boosting

## Gradient Boosting Method

### Boosting

Initialize $\pi_0(\mathbf{x}) = 0$ (for example)

For $t = 1$ to $T$, do

1) Estimate $\beta_{t,\mathbf{a}_t}$ and $\mathbf{a}_t$ by minimizing :

$$\sum_{i=1}^{n} L(y_i, \pi_{t-1}(\mathbf{x}_i) + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t))$$

2) Update $\pi(\mathbf{x}) = \pi_{t-1} + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t)$

Output : $\hat{\pi}(\mathbf{x}) = \pi_T(\mathbf{x})$

### Gradient Boosting Method

Initialize $\pi_0(\mathbf{x}) = \arg\min_{\beta} \sum_{i=1}^{n} L(y_i, \beta)$

For $t = 1$ to $T$, do

1) Estimate the negative gradients
i.e the pseudo-residuals

$$r_i = \left. \frac{\partial L(y_i, \pi(\mathbf{x}_i)}{\partial \pi(\mathbf{x}_i)} \right|_{\pi(\mathbf{x}_i))=\pi_{t-1}(\mathbf{x}_i)} \qquad i = 1, \cdots, n$$

Fit $h(\mathbf{x}_i, \mathbf{a}_t)$ to $r_i$ to get $\mathbf{a}_t$

Estimate $\beta_{t,\mathbf{a}_t}$ and $\mathbf{a}_t$ by minimizing :

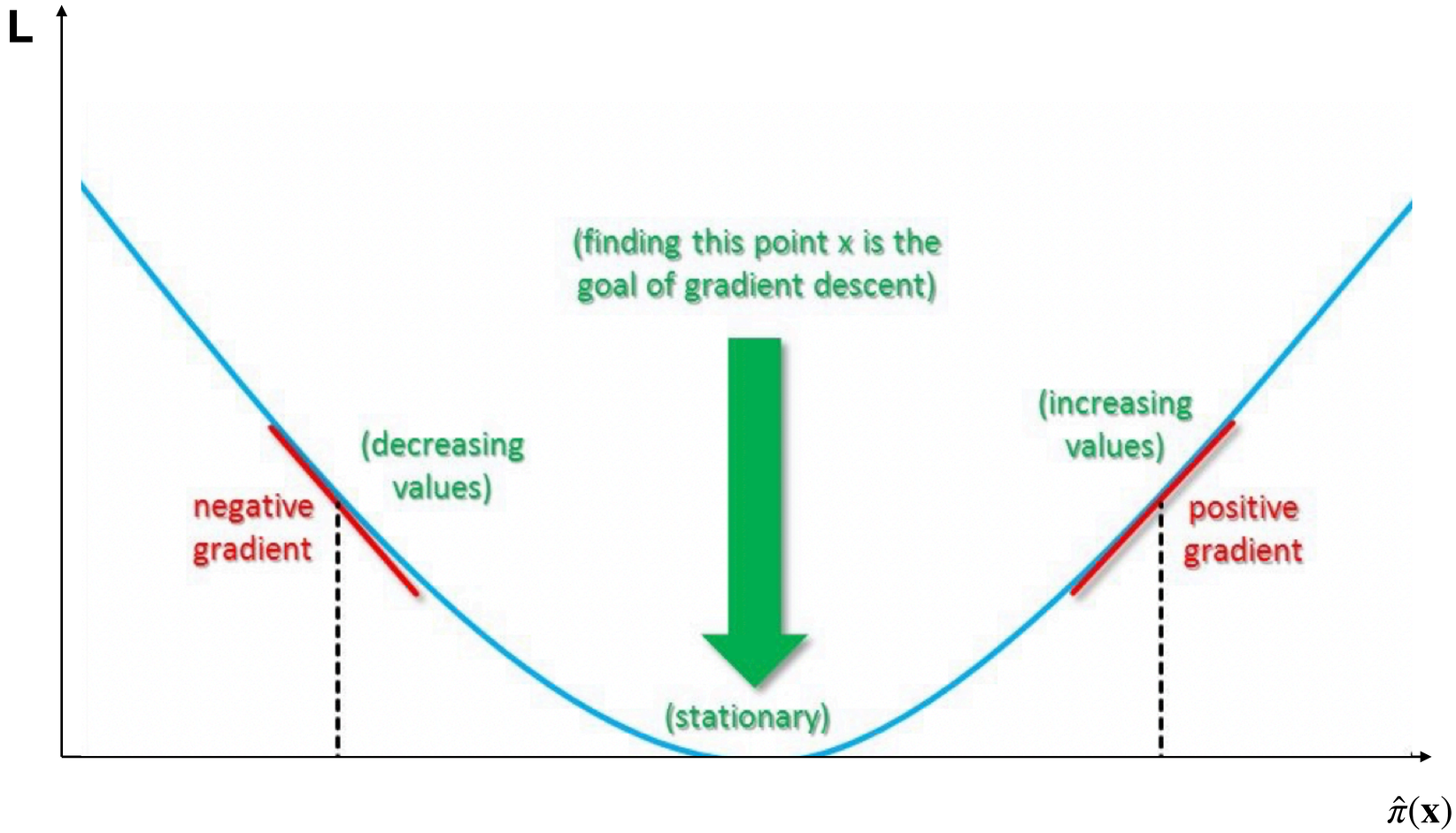
$$\sum_{i=1}^{n} L(y_i, \pi_{t-1}(\mathbf{x}_i)) + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t)$$

2) Update $\pi(\mathbf{x}) = \pi_{t-1} + \beta_{t,\mathbf{a}_t} h(\mathbf{x}_i, \mathbf{a}_t)$

Output : $\hat{\pi}(\mathbf{x}) = \pi_T(\mathbf{x})$

# Boosting & Gradient Boosting

## Gradient Boosting Method




ULB

# Adaptive Boosting Trees (ABT)

# BOOSTING COST-COMPLEXITY PRUNED TREES ON TWEEDIE RESPONSES: THE ABT MACHINE FOR INSURANCE RATEMAKING

Julie Huyghe
Department of Mathematics
Université Libre de Bruxelles (ULB)
Brussels, Belgium

Julien Trufin
Department of Mathematics
Université Libre de Bruxelles (ULB)
Brussels, Belgium

Michel Denuit
Institute of Statistics, Biostatistics and Actuarial Science
UCLouvain
Louvain-la-Neuve, Belgium

ULB

# Adaptive Boosting Trees

## Tweedie family

| | Type | Name |
|---|---|---|
| $\xi < 0$ | Continuous | - |
| $\xi = 0$ | Continuous | Normal |
| $0 < \xi < 1$ | Non existing | - |
| $\xi = 1$ | Discrete | Poisson |
| $1 < \xi < 2$ | Mixed, non-negative | Compound Poisson sums with Gamma-distributed severities |
| $\xi = 2$ | Continuous, positive | Gamma |
| $2 < \xi < 3$ | Continuous, positive | - |
| $\xi = 3$ | Continuous, positive | Inverse Gaussian |
| $\xi > 3$ | Continuous, positive | - |

Tweedie deviance loss function is given by

$$L(y, \widehat{\mu}(\boldsymbol{x})) = \begin{cases} (y - \widehat{\mu}(\boldsymbol{x}))^2 \text{ for } \xi = 0 \\ 2\left(y \ln \frac{y}{\widehat{\mu}(\boldsymbol{x})} - (y - \widehat{\mu}(\boldsymbol{x}))\right) \text{ for } \xi = 1 \\ 2\left(-\ln \frac{y}{\widehat{\mu}(\boldsymbol{x})} + \frac{y}{\widehat{\mu}(\boldsymbol{x})} - 1\right) \text{ for } \xi = 2 \\ 2\left(\frac{\max\{y,0\}^{2-\xi}}{(1-\xi)(2-\xi)} - \frac{y\widehat{\mu}(\boldsymbol{x})^{1-\xi}}{1-\xi} + \frac{\widehat{\mu}(\boldsymbol{x})^{2-\xi}}{2-\xi}\right) \text{ for } \xi > 1 \text{ and } \xi \neq 2. \end{cases}$$

ULB

# Adaptive Boosting Trees

## Idea

Instead of allowing for trees with constant interaction depth at each iteration

Let the complexity of the newly added tree adapt to the structure remaining to be learned from the data .

- Larger trees are fitted at earlier stages and they progressively simplify until reducing to the single-node tree where the ABT machine stops.

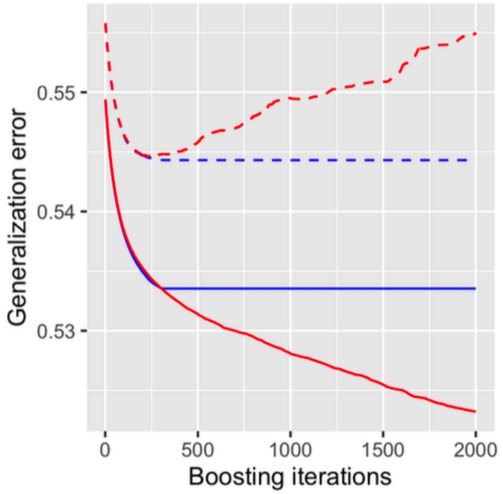
- The stopping criterion is thus built within the ABT algorithm and no computationally- intensive cross validation step is needed.

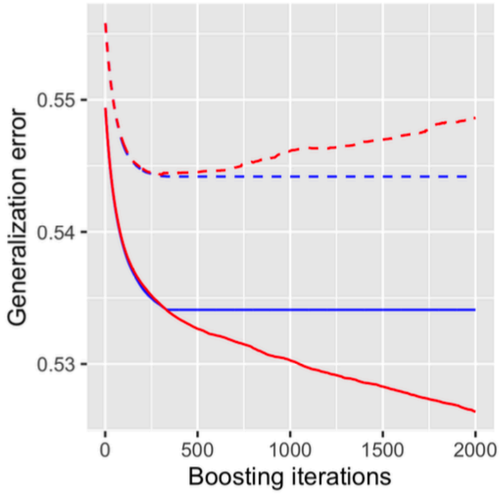
- This adaptive boosting algorithm enables to reduce overfitting

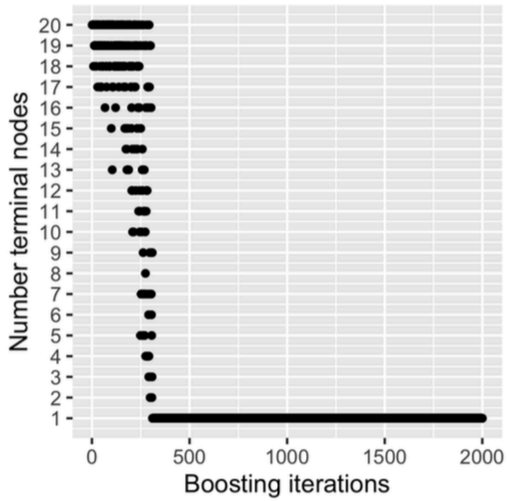
ULB

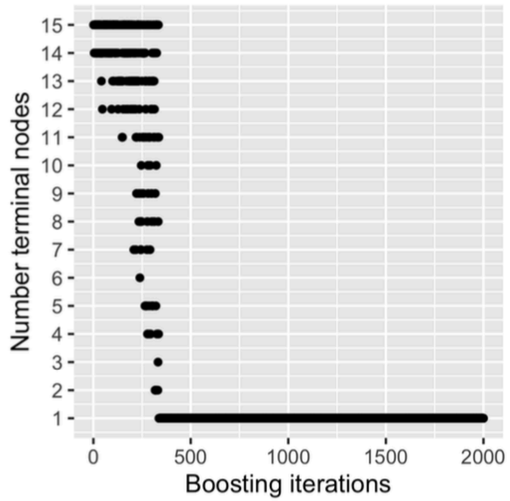
# Adaptive Boosting Trees

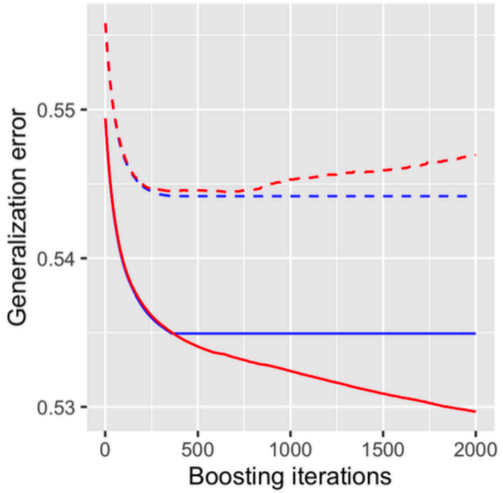
## Idea

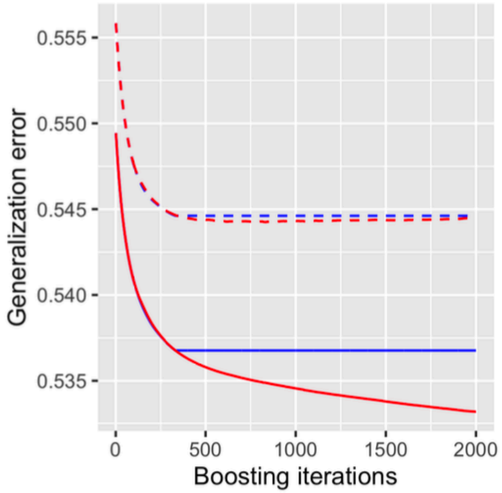


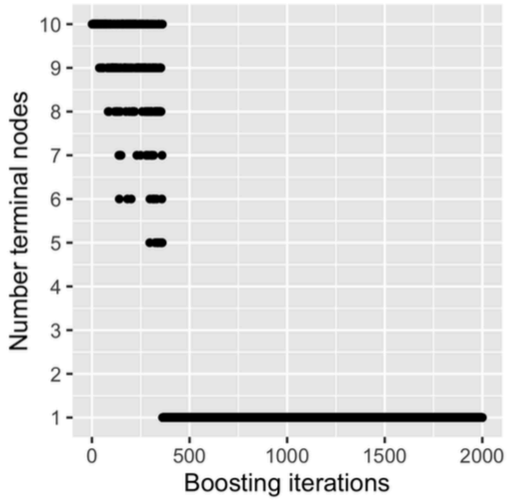



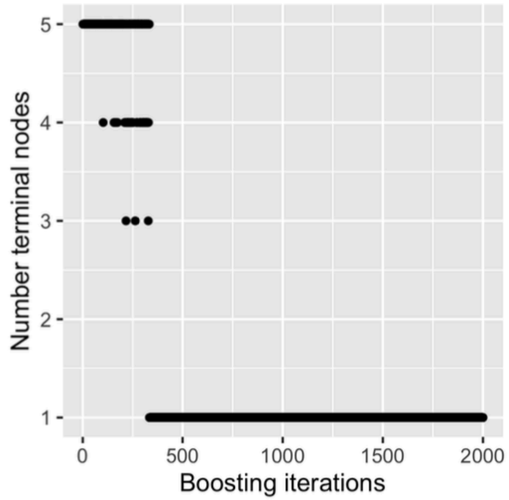




ULB

# Fairness in Machine Learning

# The Fairness of Machine Learning in Insurance: New Rags for an Old Man?

Laurence Barry[a] & Arthur Charpentier[b]

[a] Chaire PARI, Fondation Institut Europlace de Finance, Paris

[b] Université du Québec à Montréal (UQAM), Montréal (Québec), Canada

May 2022

**Abstract**

Since the beginning of their history, insurers have been known to use data to classify and price risks. As such, they were confronted early on with the problem of fairness and discrimination associated with data. This issue is becoming increasingly important with access to more granular and behavioural data, and is evolving to reflect current technologies and societal concerns. By looking into earlier debates on discrimination, we show that some algorithmic biases are a renewed version of older ones, while others show a reversal of the previous order. Paradoxically, while the insurance practice has not deeply changed nor are most of these biases new, the machine learning era still deeply shakes the conception of insurance fairness.

"*Technology is neither good nor bad; nor is it neutral*" [Kranzberg, 1986]

ULB

# Thanks a lot for your attention !

# Bibliography


Course « Assurance non vie II »
Julien Trufin

« Boosting cost complexity  pruned trees on tweedy responses » : The ABT machine for insurance ratemaking  Julie Huyghe, Julien Trufin, Michel Denuit

« Response versus gradient boosting trees, GLMs and neural networks under Tweedie loss and log-link »
Donatien Hainaut, Julien Trufin, Michel Denuit

« The fairness of machine learning in Insurance : New rags for an old man ? »
Laurence Barry, Arthur Charpentier