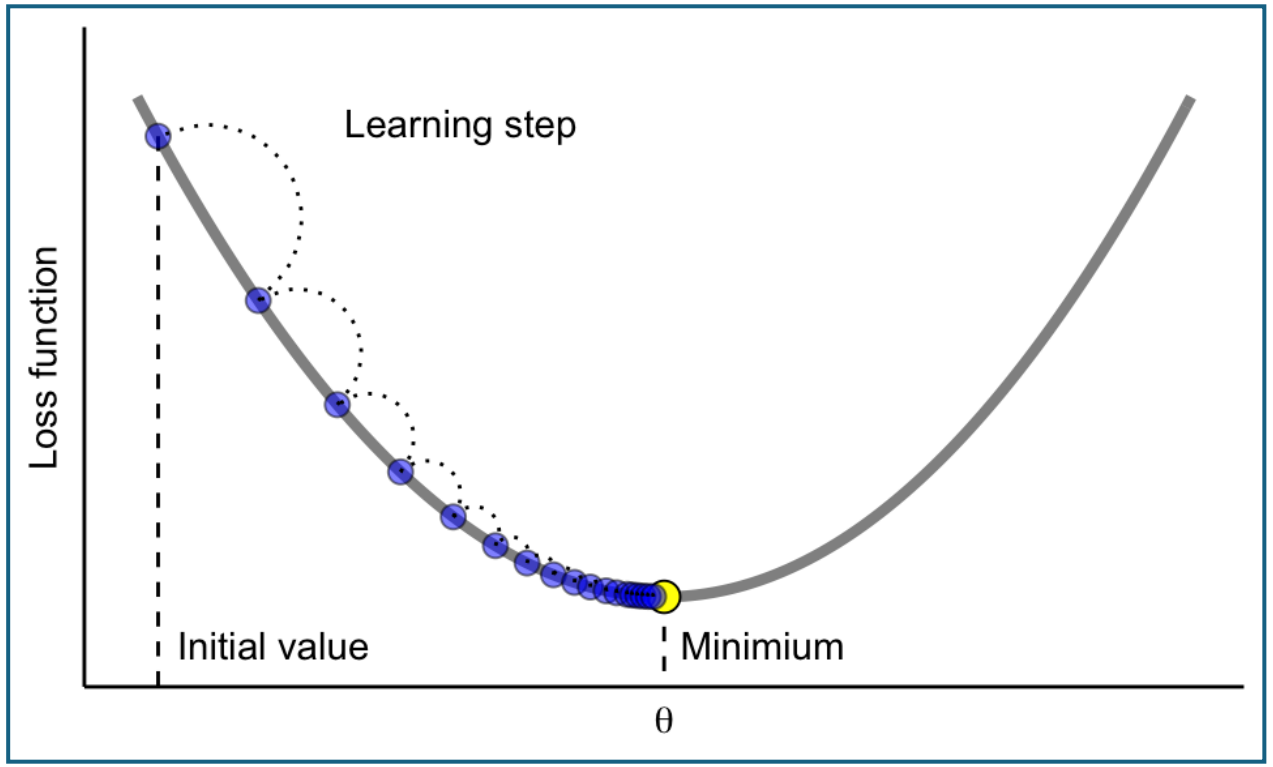
Gradient Boosting Machine

An application to Insurance data

Estimation of Claim Frequency

Introduction



- The **boosting algorithm consists in <u>sequentially</u> fitting regression trees to the residuals from the previous tree.** This specific approach is how gradient boosting minimizes the mean squared error (SSE) and other loss functions such as mean absolute error (MAE), or deviance.
- {gbm} in R takes care of Poisson and Gamma deviance, which is useful for our purpose.

Gradient boosting is considered a ***gradient descent*** algorithm. It is an optimization algorithm capable of finding optimal solutions. The general idea of gradient descent is to tweak parameter(s) iteratively in order to minimize a cost function.

Tuning parameters

- **Boosting Parameters: Drive the boosting method**
  - **Number of trees:** The total number of trees to fit.
    - Find the optimal number of trees that minimize the loss function of interest with cross validation.
  - **Learning rate:** Controls how quickly the algorithm proceeds down the gradient descent.
    - Smaller values reduce the chance of overfitting but also increases the time to find the optimal fit. This is also called shrinkage.
  - **Subsampling:** Controls whether or not you use a fraction of the available training observations. If <100% of the training observations ➔ stochastic gradient descent.
    - Help to minimize overfitting
- **Tree specific parameters:**
  - **Depth of trees:** The number *d* of splits in each tree, which controls the complexity of the boosted ensemble. If d=1 ⇔ a stump.
  - **Minimum number of observations in terminal nodes:** control the complexity of the tree.

General Tuning Strategy

1. Choose a relatively high learning rate. Generally the default value of 0.1 works but somewhere between 0.05–0.2 should work across a wide range of problems.

2. Determine the optimum number of trees for this learning rate.

3. Fix tree hyperparameters and tune learning rate and assess speed vs. performance.

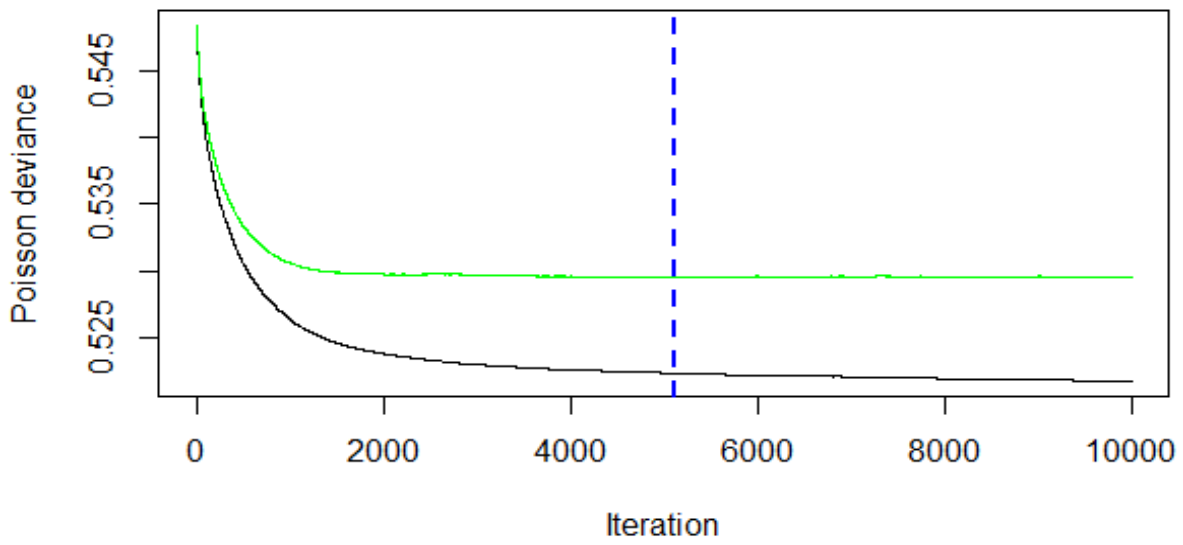
4. **Tune tree-specific parameters** for decided learning rate.

5. Once tree-specific parameters have been found, lower the learning rate to assess for any improvements in accuracy.

6. Use final hyperparameter settings and increase CV procedures to get more robust estimates. Often, the above steps are performed with a simple validation procedure or 5-fold CV due to computational constraints. If you used *k*-fold CV throughout steps 1–5 then this step is not necessary.

Results 1 – OOB Error and Cross-Validation

OOB Error and CV chart




```r
library(gbm)
# set the seed for reproducibility
set.seed(1234)

# Train GBM model
gbm.fit <- gbm(
  formula = as.formula(paste('clm.count ~ offset(log(exposure)) +', paste(features, collapse = ' + '))),
  distribution = "poisson", # Need the poisson distribution here, as we are dealing with a count (frequency)
  data = dta_trn,
  n.trees = 10000,
  interaction.depth = 1, # a stump
  shrinkage = 0.01, # learning rate
  cv.folds = 5, # cross-validation fold
  n.cores = NULL, # will use all cores by default
  verbose = FALSE,
  bag.fraction = 0.75, # delta in Table 1
  #n.minobsinnode = 0.01 * 0.75 * nrow(dta_trn),
  n.minobsinnode = 10
  )
```

Objectives

- Compute an average claim frequency.

Methodology

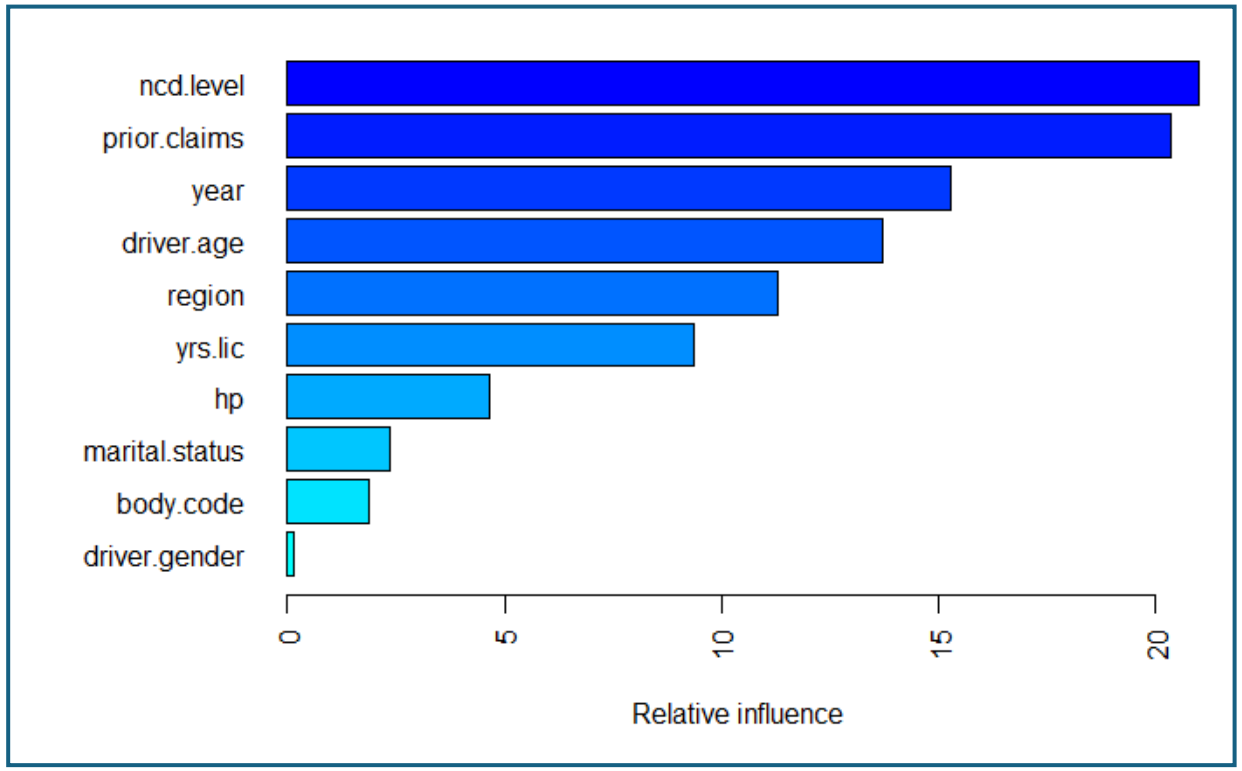
- We can set up a n cv-fold with the {gbm} package in R. Here 5 are used.
- 10000 iterations were performed.
- Trees set up as a stump.

Results

- At this learning rate of 0.01, the best cross-validation iteration was achieved with 5120 trees giving a for a Deviance of 0.529.

Results 2 – Variable Importance

VIP chart

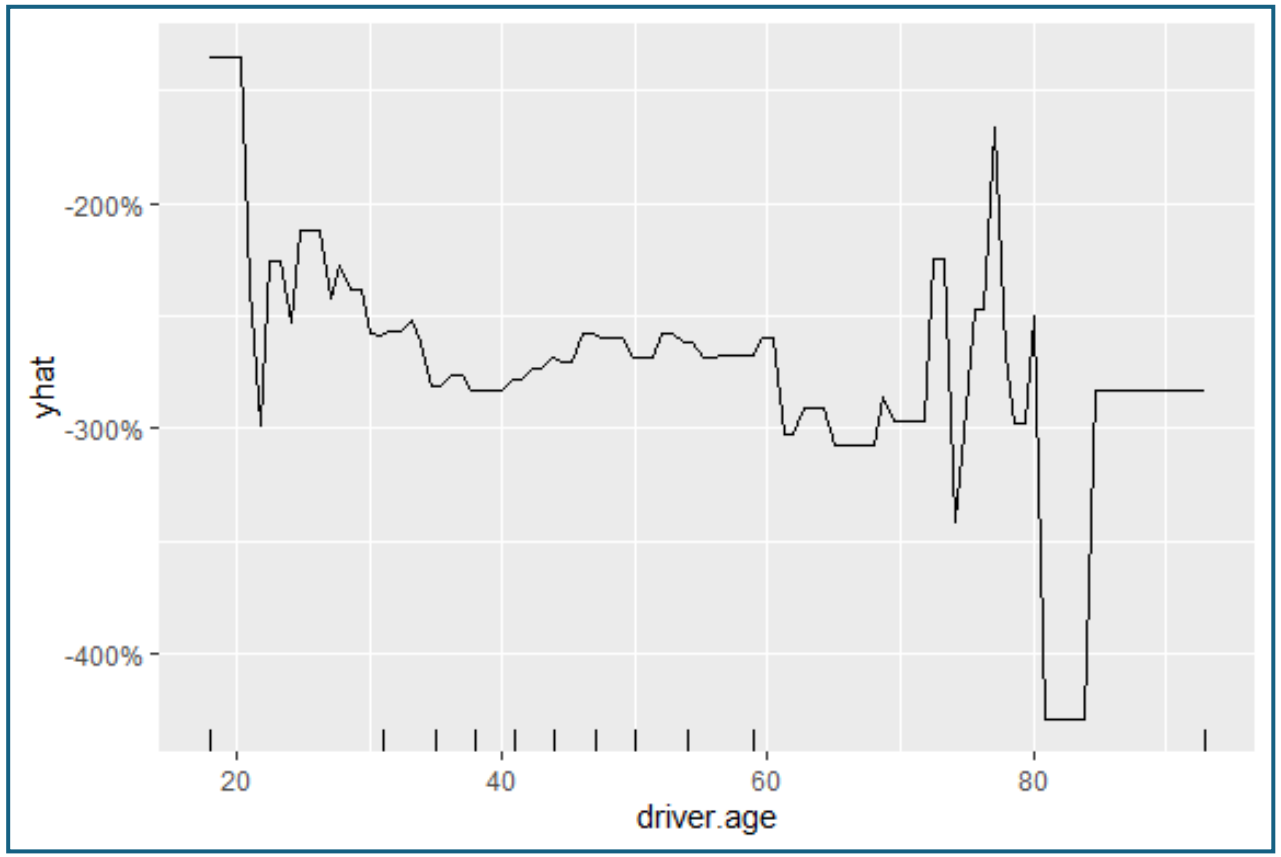


Results

- Like for Random Forest, we can publish a Variable Importance chart.
- Variable importance is determined by calculating the relative influence of each variable:
  - If selected to split the data during the tree building process,
  - how much the error decreased as a result.
- The information on the past claims carried by “ncd.level” and “prior.claims” are the most important.

Results 3 – Partial Dependence Plot

PDP Plot



Results

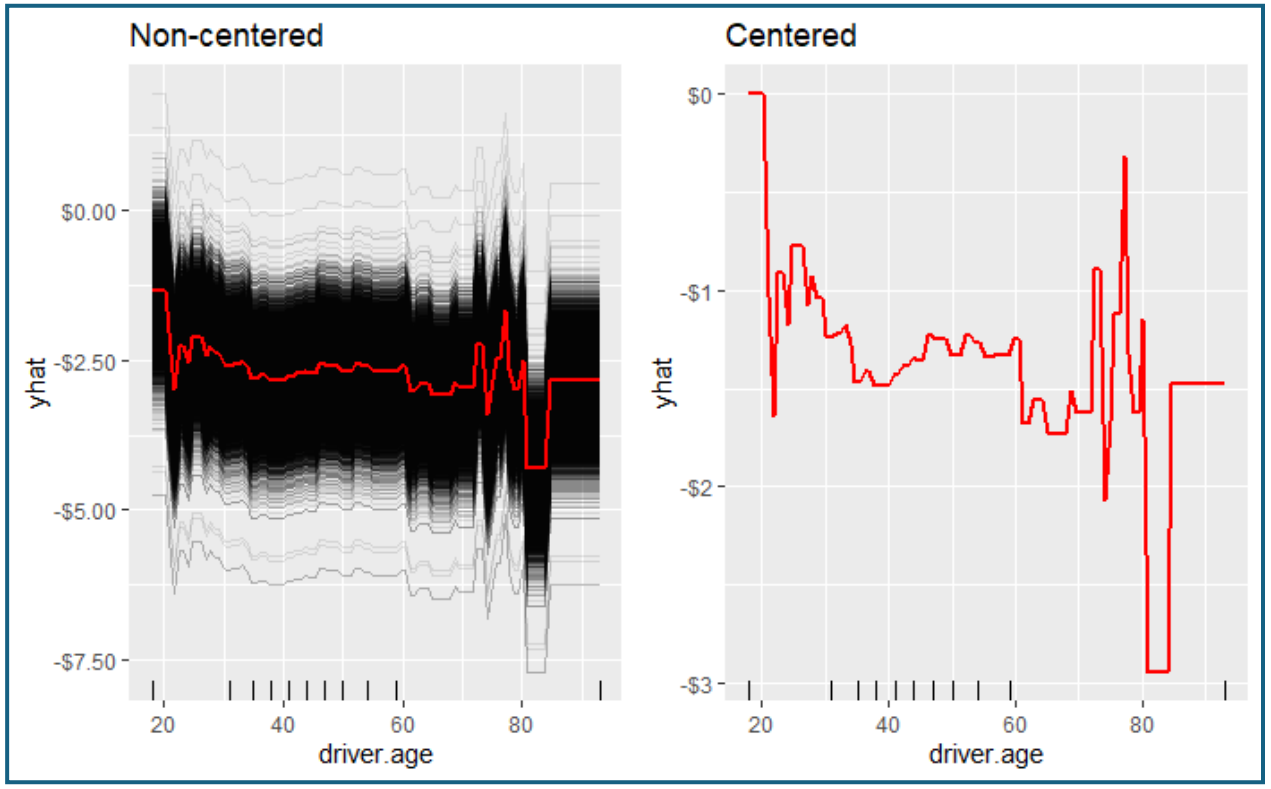
We want to look at the effect of one variable only on the claim frequency, here the driver age.

Globally, the average change of frequency decreases as the driver's age increases.

However, we observe a peak for the older driver, over 70yo.

Results 4 – Individual Conditional Expectation

ICE Plot



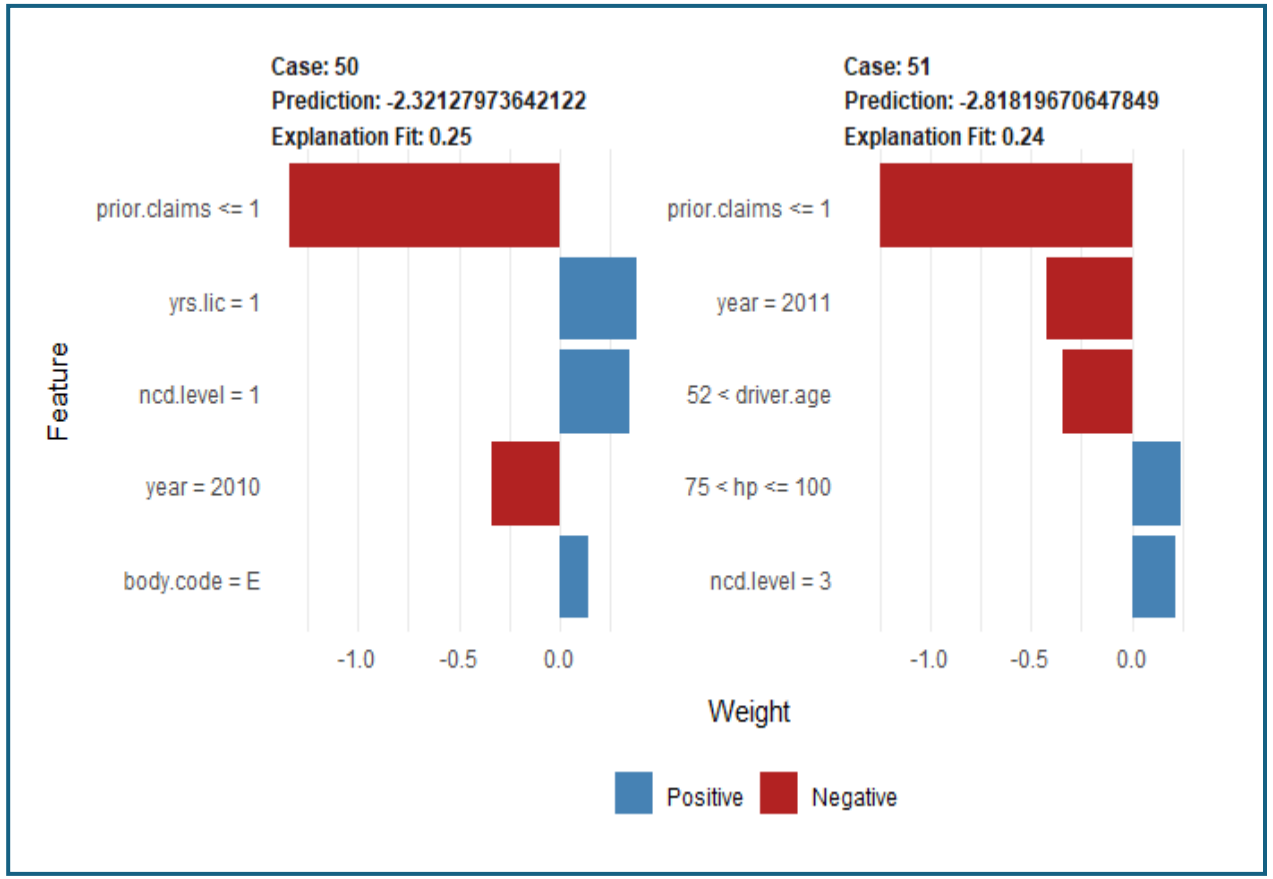
Results

ICE curves are an extension of PDP plots but, rather than plotting the average marginal effect on the response variable, we plot the change in the predicted response variable for each observation as we vary each predictor variable.

Here we tried with the driver age. The shape is the same as seen the previous slide.

Results 5 – LIME package

LIME



A procedure to understand why a prediction resulted in a given value for a single observation.

GBM – Pros & Cons

PROS

- Highest predictive accuracy
- Flexibility:
  - Different Loss function,
  - Large range of hyperparameter.
- No pre-processing required
- Handle missing data

CONS

- Minimize all errors, outliers included ➔ can cause overfitting (CV needed),
  - Objective: to find the optimal number of trees that minimize the loss function of interest with cross validation.
- Computationally expensive,
- Less interpretable.

Sources

- *Predictive Modelling Applications in Actuarial Science, Vol.2* (E. Frees & al.)
- *Hands on ML with R (B. Broecke)* https://bradleyboehmke.github.io/HOML/