Decision Tree Algorithm

An application to Insurance data

Introduction

- **Objectives:**
  - Estimate the claims frequency of car drivers using a Regression Tree with the R package {rpart}.
  - Purely educative and stands as an introduction to explore the other Tree-based algorithms.
- One among many other predictive modelling approaches.
- Advantage to be easily interpreted and work for both classification and regression tasks.
- Single tree model typically lack in predictive performance comparing to ensemble methods like Random Forest or GBM.

Data in use

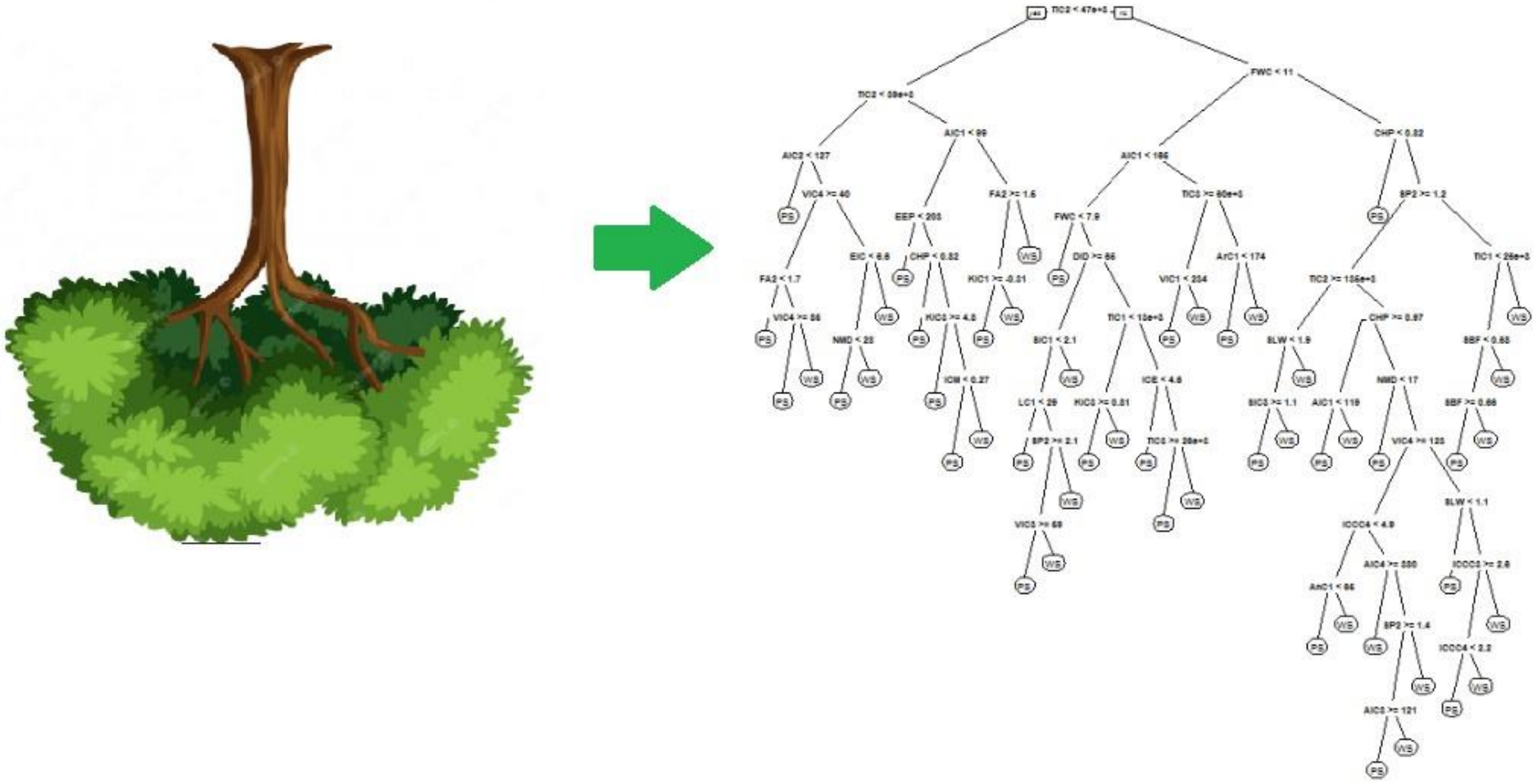
```r
# Define columnn class for dataset
colCls <- c("integer",          # row id
            "character",        # analysis year
            "numeric",          # exposure
            "character",        # new business / renewal business
            "numeric",          # driver age (continuous)
            "character",        # driver age (categorical)
            "character",        # driver gender
            "character",        # marital status
            "numeric",          # years licensed (continuous)
            "character",        # years licensed (categorical)
            "character",        # ncd level
            "character",        # region
            "character",        # body code
            "numeric",          # vehicle age (continuous)
            "character",        # vehicle age (categorical)
            "numeric",          # vehicle value
            "character",        # seats
            rep("numeric", 6),  # ccm, hp, weight, length, width, height (all continuous)
            "character",        # fuel type
            rep("numeric", 3)   # prior claims, claim count, claim incurred (all continuous)
)
```

```
## 'data.frame':    40760 obs. of  27 variables:
##  $ row.id        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ year          : chr  "2010" "2010" "2010" "2010" ...
##  $ exposure      : num  1 1 1 0.08 1 0.08 1 1 0.08 1 ...
##  $ nb.rb         : chr  "RB" "NB" "RB" "RB" ...
##  $ driver.age    : num  63 33 68 68 68 68 53 68 68 65 ...
##  $ drv.age       : chr  "63" "33" "68" "68" ...
##  $ driver.gender : chr  "Male" "Male" "Male" "Male" ...
##  $ marital.status: chr  "Married" "Married" "Married" "Married" ...
##  $ yrs.licensed  : num  5 1 2 2 2 2 5 2 2 2 ...
##  $ yrs.lic       : chr  "5" "1" "2" "2" ...
##  $ ncd.level     : chr  "6" "5" "4" "4" ...
##  $ region        : chr  "3" "38" "33" "33" ...
##  $ body.code     : chr  "A" "B" "C" "C" ...
##  $ vehicle.age   : num  3 3 2 2 1 1 3 1 1 5 ...
##  $ veh.age       : chr  "3" "3" "2" "2" ...
##  $ vehicle.value : num  21.4 17.1 17.3 17.3 25 ...
##  $ seats         : chr  "5" "3" "5" "5" ...
##  $ ccm           : num  1248 2476 1948 1948 1461 ...
##  $ hp            : num  70 94 90 90 85 85 70 85 85 65 ...
##  $ weight        : num  1285 1670 1760 1760 1130 ...
##  $ length        : num  4.32 4.79 4.91 4.91 4.04 ...
##  $ width         : num  1.68 1.74 1.81 1.81 1.67 ...
##  $ height        : num  1.8 1.97 1.75 1.75 1.82 ...
##  $ fuel.type     : chr  "Diesel" "Diesel" "Diesel" "Diesel" ...
##  $ prior.claims  : num  0 0 0 0 0 0 4 0 0 0 ...
##  $ clm.count     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ clm.incurred  : num  0 0 0 0 0 0 0 0 0 0 ...
```

- **Data:** *Predictive Modelling Applications in Actuarial Science, Vol.2* (E. Frees & al.): https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/glm/v2-chapter-1.html
- Data already explored in a previous study (cf. EDA for Insurance) stored in another repository where a description of the fields is also available.

Single Tree

Example of a real tree



Classification And Regression Tree algorithm, aka CART, developed by Breiman et al. in 1984 works by partitioning the feature space into a number of smaller (non-overlapping) regions with similar response values using a set of splitting rules. The goal is at each threshold, the minimum sum squared of residuals between the observed value and the predicted is minimal.

Single Tree

```{r}
# Remove the predictors that are not in use for the analysis
dta_trn <- subset(dta, select = -c(row.id, drv.age, clm.incurred, hp, length, width, height, sev, veh.val.q15, veh.val.q35,
                                   driver.age.q35, driver.age.q49, driver.age.q59,
                                   efq, esv, epp
                                   ))
```

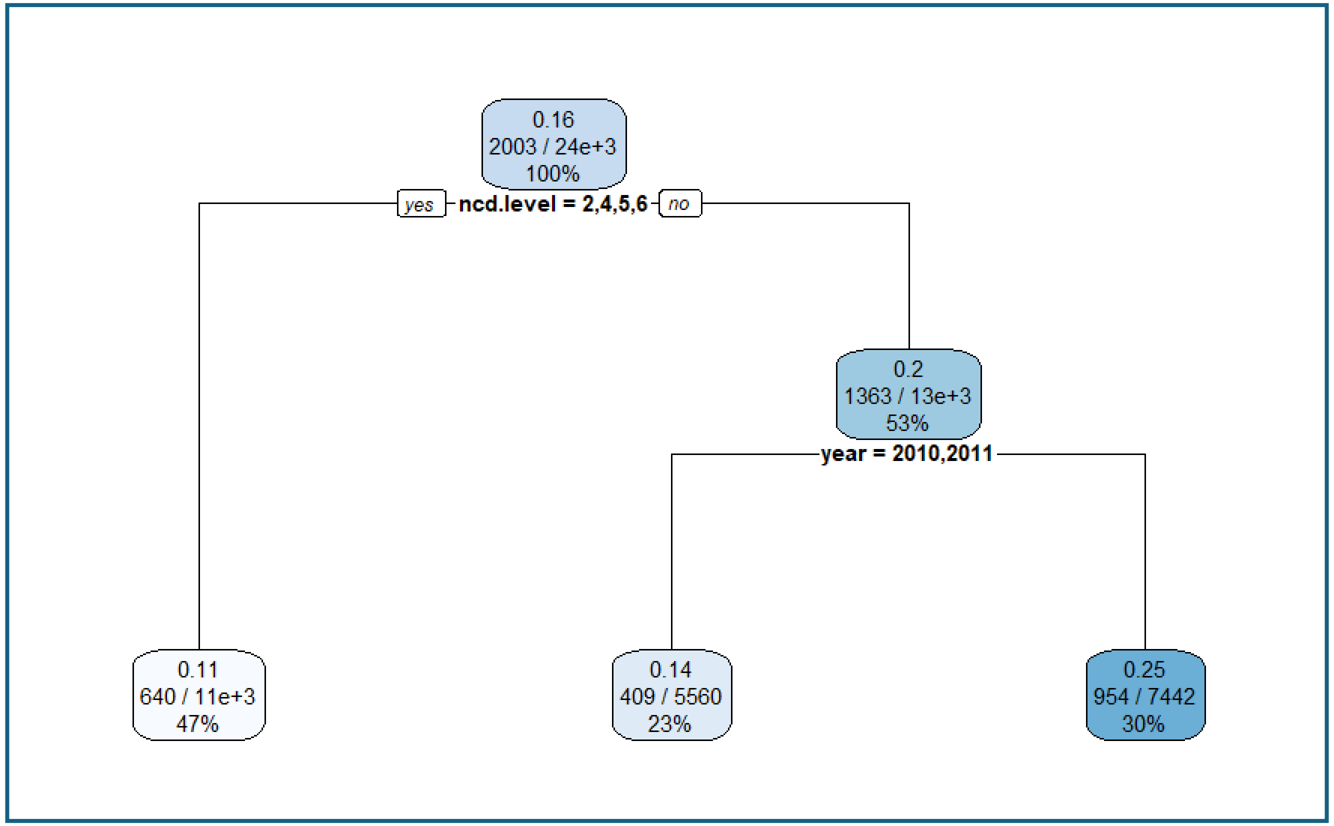
```{r}
library(rpart)        # direct engine for decision tree application
library(caret)        # meta engine for decision tree application
fit <- rpart(formula =
                cbind(exposure,clm.count) ~ . ,
                data = dta_trn,
                subset = train,
                method = 'poisson'
                )
print(fit)
```

$$D^{\text{Poi}} = \frac{2}{n} \sum_{i=1}^{n} y_i \cdot \ln \frac{y_i}{\text{expo}_i \cdot \hat{f}(x_i)} - \{y_i - \text{expo}_i \cdot \hat{f}(x_i)\},$$

As we want to predict a frequency, we need to specify using the Poisson deviance by calling the method "poisson" and setting the response as a two-column matrix including the exposure.

In this first example, we take just some potential predictors and leave all the hyperparameters at their default level.

Single Tree

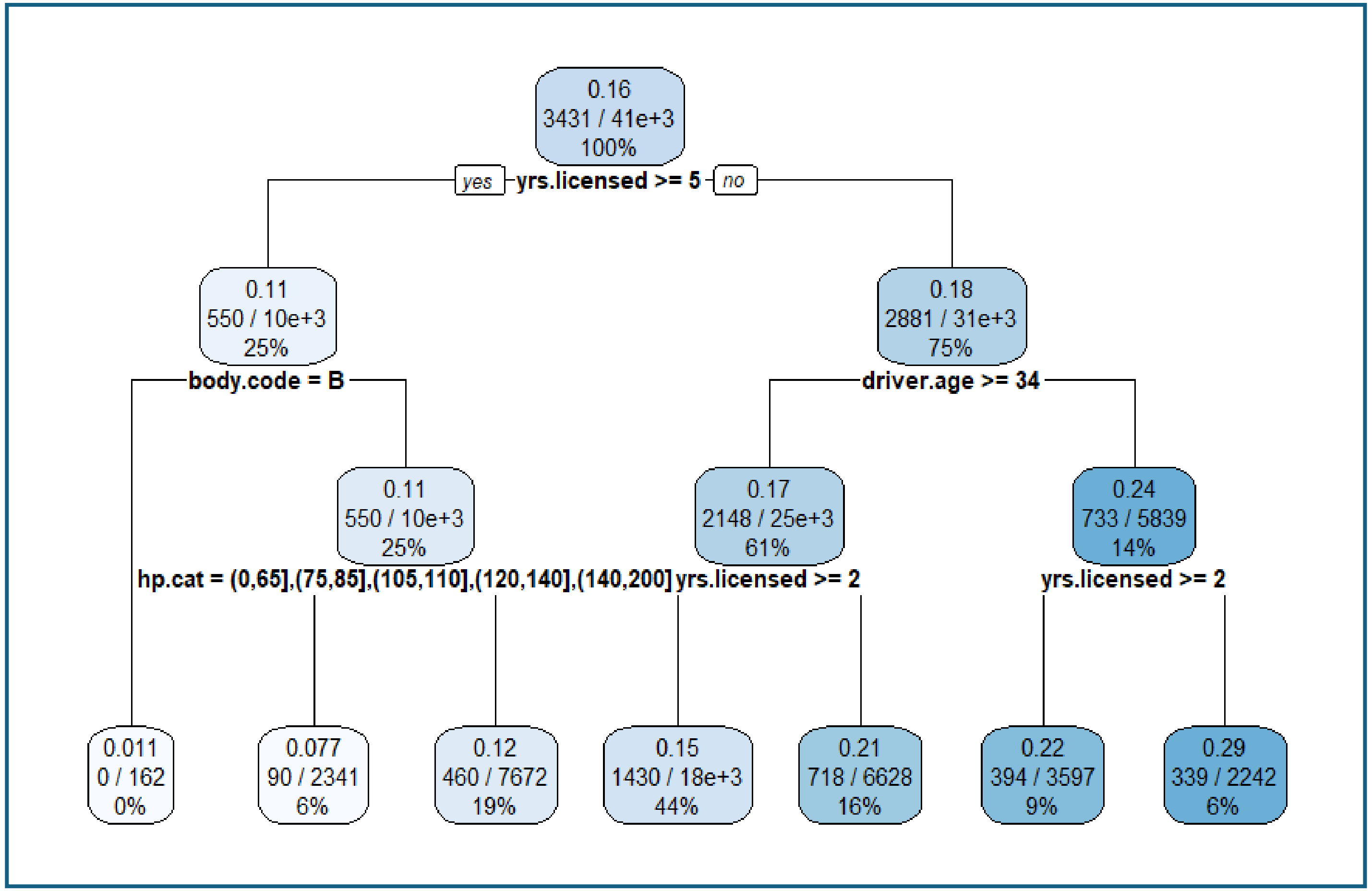



```
Warning: package ‘ggplot2’ was built under R version 4.1.3n= 24495

node), split, n, deviance, yval
      * denotes terminal node

1) root 24495 9848.918 0.1606008
  2) ncd.level=2,4,5,6 11493 3598.377 0.1108765 *
  3) ncd.level=1,3 13002 6079.696 0.2034859
    6) year=2010,2011 5560 2112.325 0.1412178 *
    7) year=2013,2012 7442 3866.075 0.2509145 *
```

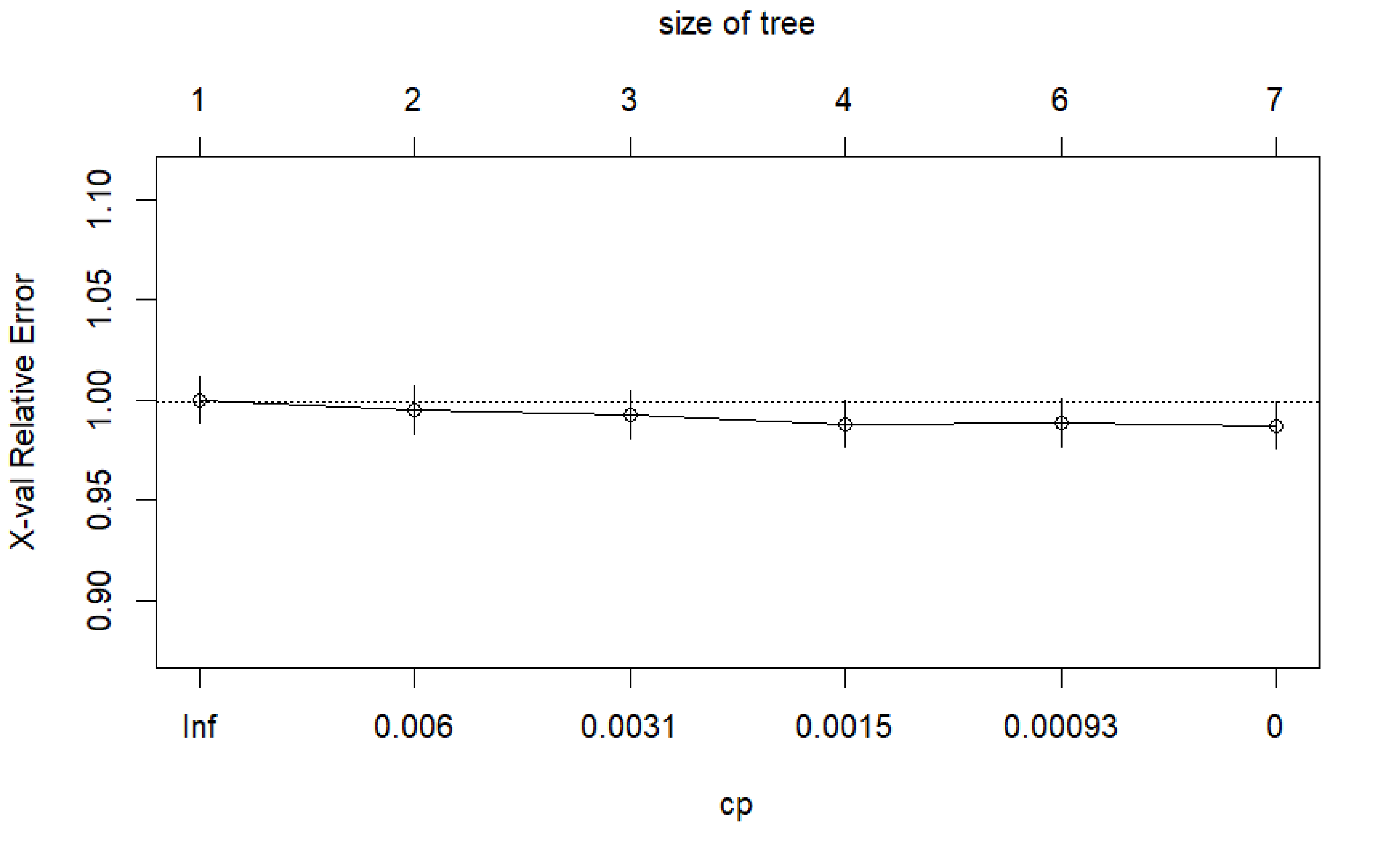
- We have 24495 observations
- “ncd” (No-Claim Discount) is the first variable that optimize the reduction of the Poisson Deviance.
- The sample is split in 2 regions: 47% and 53%
- The first region, we have a claims frequency of 11%.




```{r}
# A second model
fit2 <- rpart::rpart(formula =
                  cbind(exposure, clm.count) ~
                  driver.age  + hp.cat
                   + fuel.type + driver.gender + body.code + yrs.licensed + vehicle.value,
                   data = dta_trn,
                   method = 'poisson',
                   control = rpart.control(
                     maxdepth = 3,
                     cp = 0 )
)
print(fit2)
```

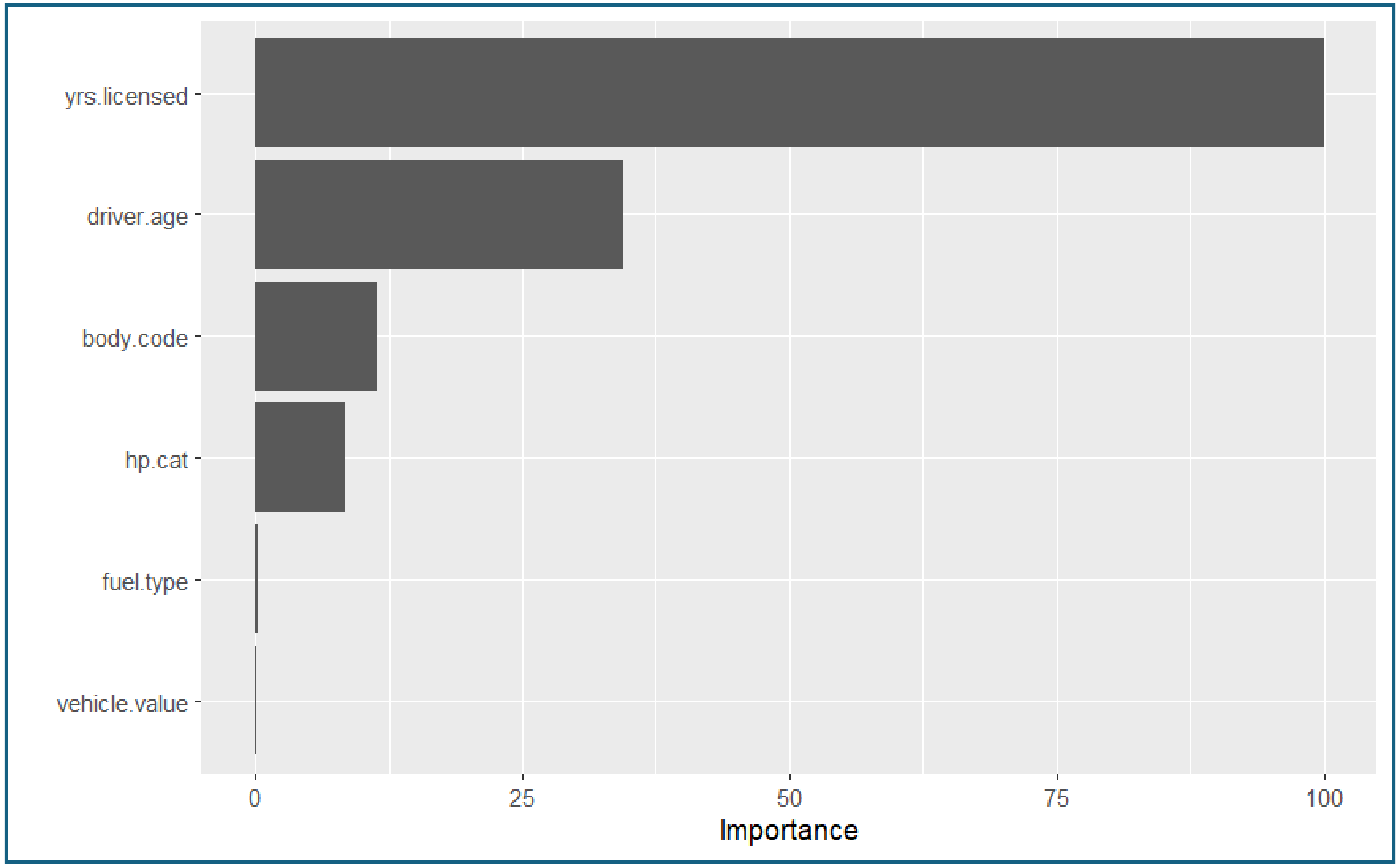
- We can get a more complex tree by adjusting the hyper-parameters:
  - 'control' is an argument that provide a list of hyperparameter value.
  - 'maxdepth' represents the maximum depth of the tree, set up at 3.
  - 'cp' is the complexity parameter, that specify the proportion by which the overall error should improve for a split to be attempted. We force rpart to generate a full tree by setting that parameter at 0.

Pruning



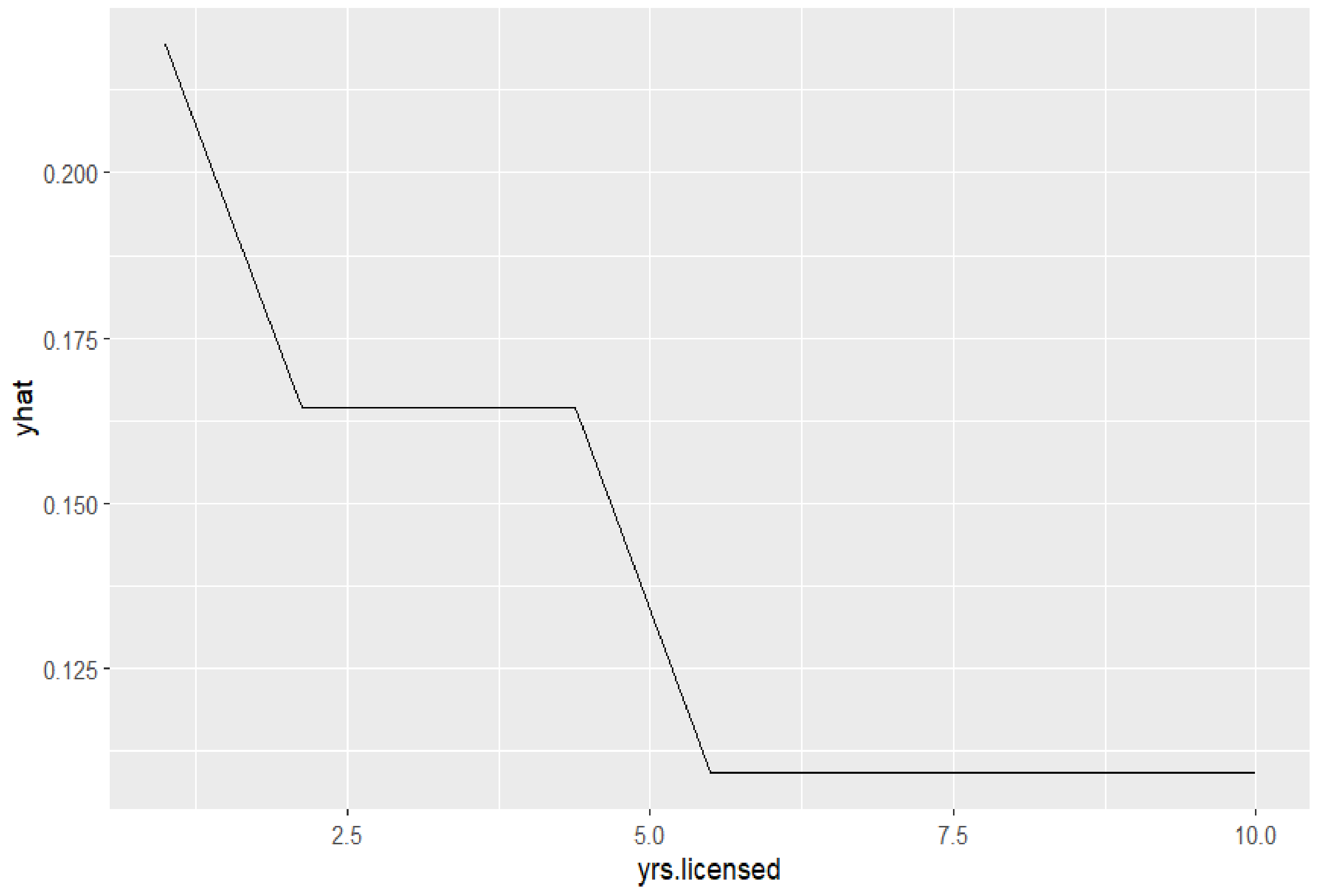
- Behind the scenes, {rpart} is automatically applying a range of cost complexity values to prune the tree. To compare the error for each value, it performs a 10-fold cross validation so that the error associated with a given value is computed on the hold-out validation data.
- The results are summarized in the below graph where the y-axis represents the cross-validation error, the x-axis the cost-complexity value and the upper-x is the number of terminal nodes for the tree.

Interpretation



- Variable Importance Plot
- Feature importance is represented by the reduction in the loss function attributed to each variable at each split.

Partial Dependance Plot



- We use partial dependence plots (PDPs) to get an insight on the relation between a feature and the target. It shows the marginal effect that one or two features have on the predicted outcome of a machine learning model (J. H. Friedman).The plot can show whether the relationship between the target and a feature is linear, monotonic or more complex. In our case, the function 'partial' from the {pdp} package performs the essential steps to generate such a PD effect.
- Here, we observe that the more experienced the driver is, the less incline to get an accident.

Conclusion

- A very intuitive and flexible modeling approach.
- Unfortunately, it suffers from high variance.
- Combination of trees, like Bagging and more complex algorithm such as Random Forest and GBM provides better results.

Sources

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