KPMG
cutting through complexity

Methods for growing classification and regression trees

Robert Meixner

19 December 2014

Introduction

Introduction

Selected methods

- CART
- Random forests

Case study

Areas of application

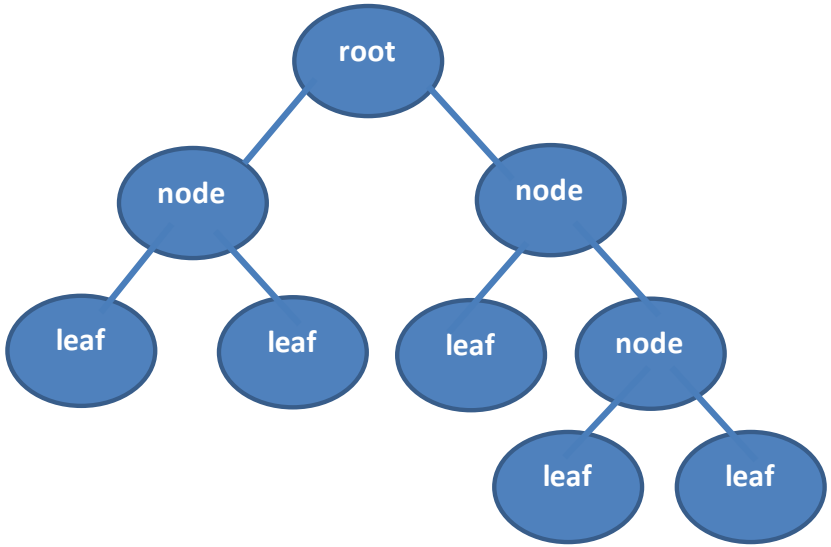
Introduction

Classification and regression trees

- Classification and regression trees are predictive models based on the decision trees

$$\hat{y} = f(\boldsymbol{x})$$

- The dependent variable $y \in Y$ can be:
  - Categorical -> **classification trees**
  - Continuous -> **regression trees**
- The independent variables of vector $\boldsymbol{x} \in X$ can be both continuous and categorical



Single tree algorithms

- Roots of tree algorithms date to 1950s (automatic detection of interactions)
- Algorithms of today type developed in 1980s
- CART algorithm (Breiman at al [1], published in 1984)

Multi tree algorithms

- Multi tree algorithms developed in 1990s and later
- Examples of multi tree algorithms
  - Bagging
  - Random forest
  - Boosted trees

Simple trees

Introduction

Selected methods

- CART
- Random forests

Case study

Areas of application

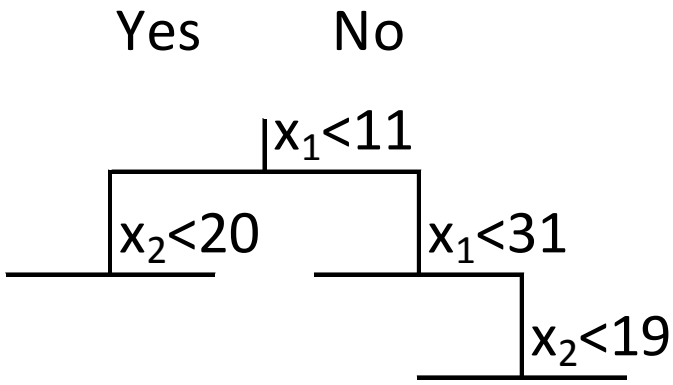
CART - introduction

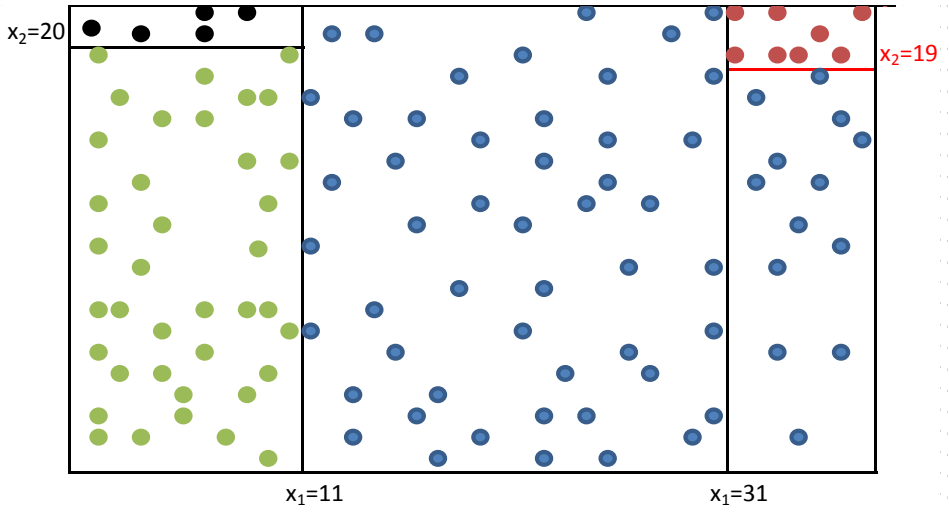
CART algorithm

- CART uses recursive partitioning of the space of independent variables
- The tree is grown based on the following inputs:
  - Training sample
    $$L=\{(\boldsymbol{x}_i, y_i), i=1,\ldots,n\}$$
  - Splitting criterion
  - Stopping criterion
- The tree root contains whole sample
- The sample contained in each node is split into two disjunctive sets
- The selected split is optimal for given node according to selected criteria
- The algorithm is repeated until the stopping criterion is satisfied

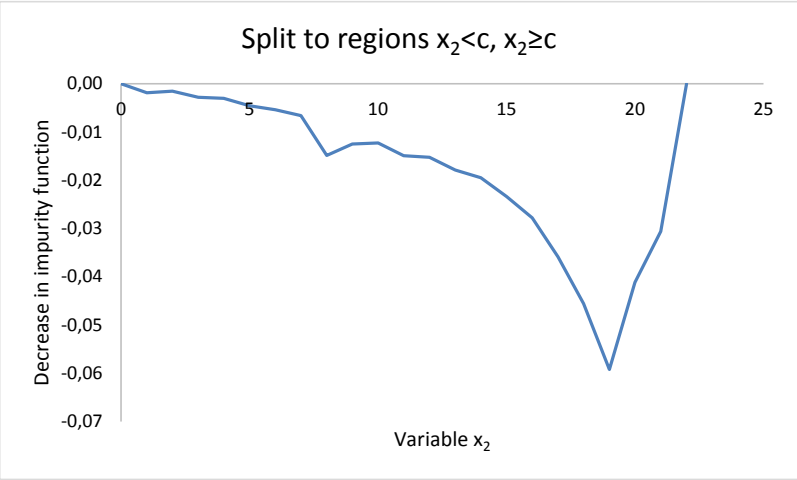
The form of the split is given by:

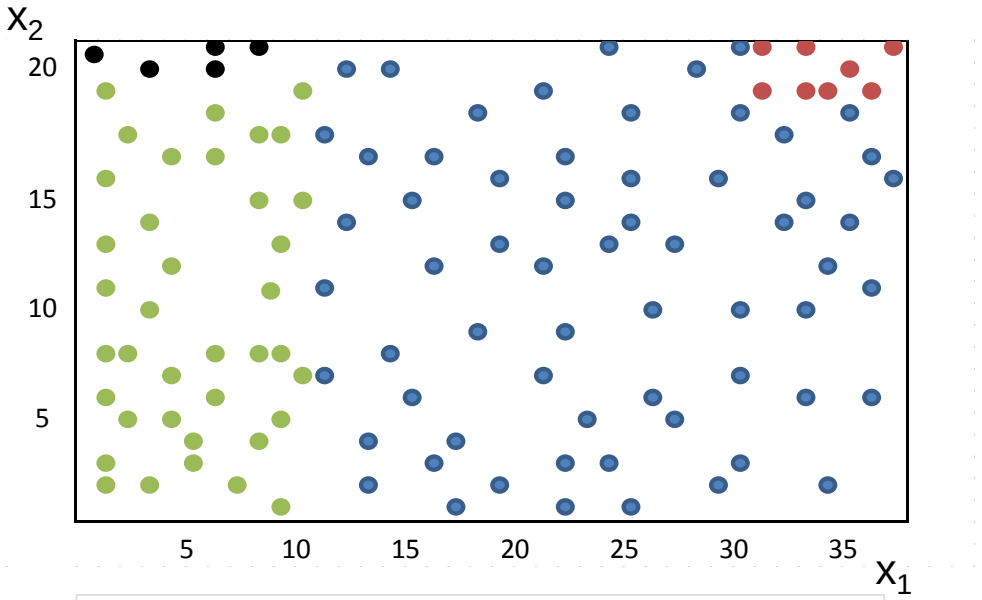
- Condition of type "$x_i<c$", $c \in (-\infty, \infty)$ for continuous variables
- List of classes "$x_i \in \{A_1, \ldots, A_k\}$ for categorical variables
- Each split depends only on one variable



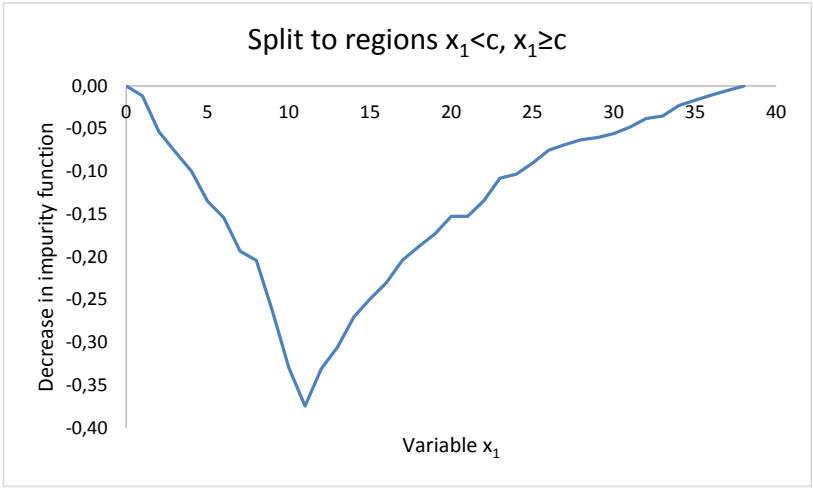


CART – splitting criterion





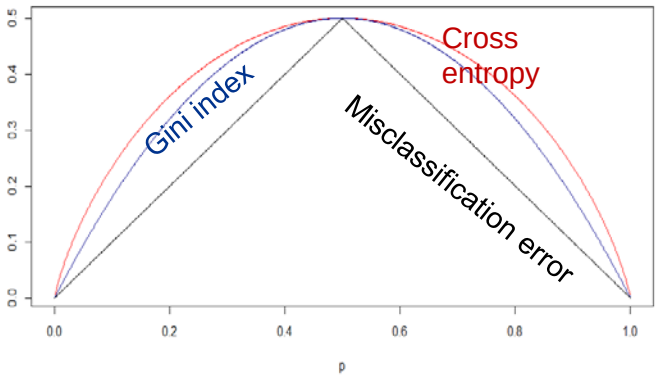
The decrease in impurity function has minimum for split to intervals $x_1<11$, $x_1 \geq 11$



CART – splitting criterion

Information criteria used for classification trees

- The homogeneity of data measured by “impurity function”
- The impurity function must comply with the following conditions:
  - Symmetric function
  - Maximum in point (*1/K*,…,*1/K*) (*K* classes of dependent variable)
  - Minimum in points (1,…,0), …, (0,…,1)
- Examples of impurity functions:
  - Misclassification error $I_M = 1 - \max_k \hat{p}_{tk}$
  - Gini index $I_G = \sum_{k=1}^{K} \hat{p}_{tk}(1 - \hat{p}_{tk})$
  - Cross entropy $I_E = -\sum_{k=1}^{K} \hat{p}_{tk} \log \hat{p}_{tk}$



$$\hat{p}_{tk} = \frac{1}{|t|} \sum_{x_i \in t} I(y_i = k)$$

i.e. it is the proportion of observations in the node *t*, for which the dependent variable is class *k*,
I is the indicator function

Information criteria used for regression trees

- The information criteria is the estimate of mean squared error for given node/leaf *t*

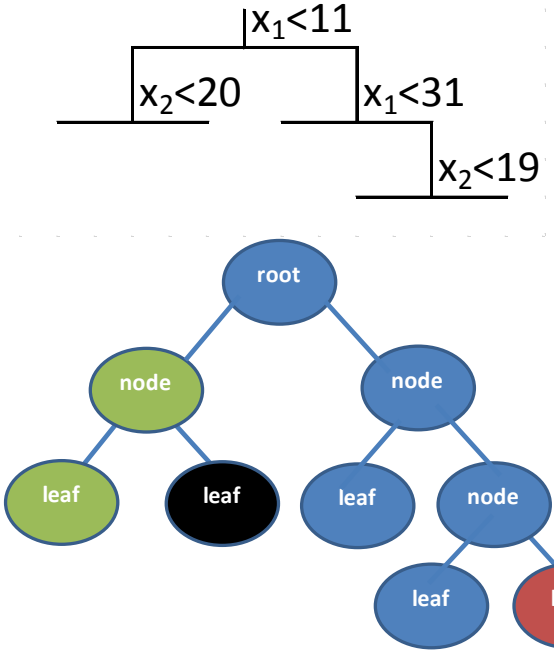
$$\widehat{MSE}(t) = \frac{1}{|t|} \sum_{\boldsymbol{x}_i \in t} (f(\boldsymbol{x}_i) - y_i)^2$$

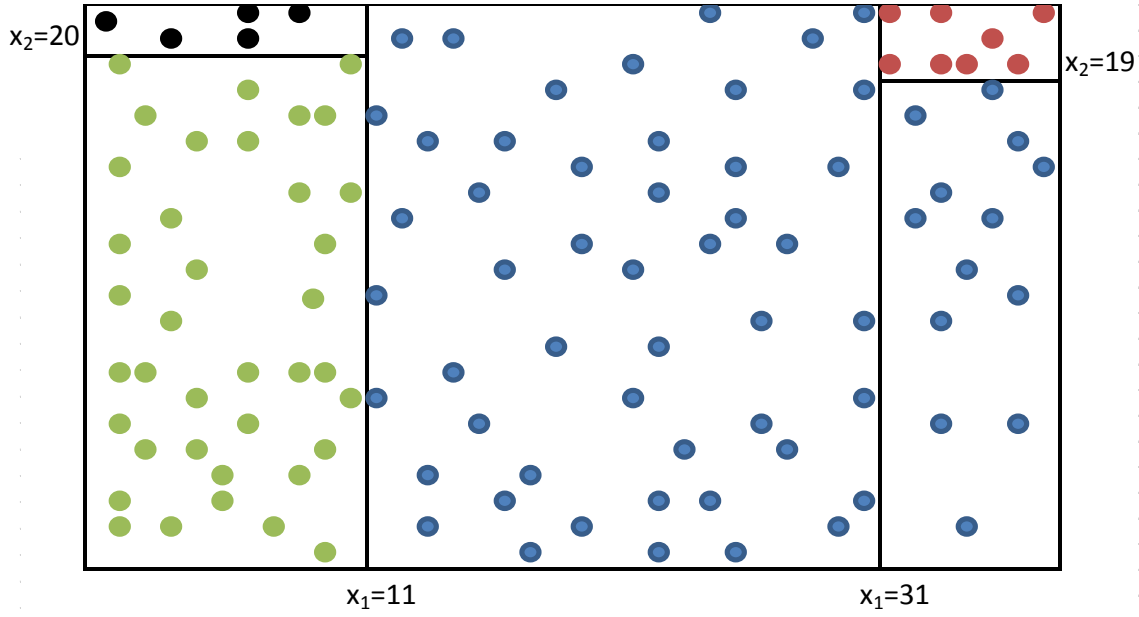
where |*t*| is number of observations (training data) falling into the node *t*

CART – stopping criterion

Stopping criterion

- Avoiding over fitted and unnecessarily complicated models while growing the tree
- Approaches based on measures applied on particular nodes
  - Set minimal number of training data falling to each node
  - Set threshold on the decrease of selected information criteria (sum of squared errors, entropy,…)
- Approaches based on measures applied on tree
  - Set limit on maximal depth of the tree
  - Set maximum number of nodes in the tree





CART – model error

	Classification tree	Regression tree
General model error for classification / regression problem	$R(f) = P(f(\boldsymbol{X}) \neq Y)$	$R(f) = MSE(f) = E(f(\boldsymbol{X}) - Y)^2$
Model error given observation falls to leaf $\tilde{t}$	$r(\tilde{t}) = 1 - \max_k p_{\tilde{t}k}$	$MSE(\tilde{t}) = E((f(\boldsymbol{X}) - Y)^2 \mid \tilde{t})$
Model value (prediction) for observation falling to the leaf $\tilde{t}$	$f(\tilde{t}) = \underset{k}{argmax}\, p_{\tilde{t}k}$	$f(\tilde{t}) = E(Y \mid \tilde{t})$
Model error of tree *T*	$R(T) = \sum_{\tilde{t}} p(\tilde{t}) r(\tilde{t})$	$R(T) = MSE(T) = \sum_{\tilde{t}} p(\tilde{t}) MSE(\tilde{t})$

where $p(\tilde{t}) = P(\boldsymbol{X} \in \tilde{t})$, $p_{\tilde{t}k} = P(Y = k \mid \boldsymbol{X} \in \tilde{t})$

The model value (prediction) given observation falls to the leaf $\tilde{t}$ is estimated as

- **Average value** of training data falling to the leaf $\tilde{t}$ for regression tree

$$\widehat{f(\tilde{t})} = \frac{1}{|\tilde{t}|} \sum_{\boldsymbol{x}_i \in \tilde{t}} y_i$$

- **The most frequent class** of training data falling to the leaf $\tilde{t}$ for classification tree

$$\widehat{f(\tilde{t})} = \underset{k}{argmax} \sum_{\boldsymbol{x}_i \in \tilde{t}} I(y_i = k)$$

CART – estimate of model error

Estimate based on the pure training set - biased downward

Estimate based on the testing set

- While growing a tree, do not use whole sample, randomly select testing sample
- Testing sample typically contains 1/2 or 1/3 of all data
- The unused part of the sample serves as testing set
- Estimate of model error based on the testing set

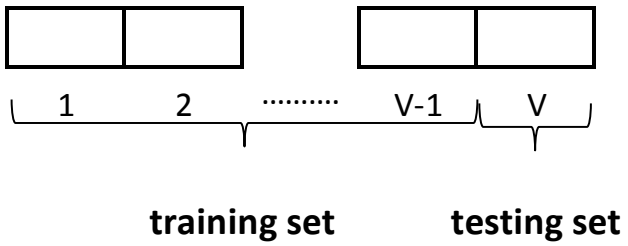
$$\widehat{R(T)} = R^{TS}(T) = \frac{1}{n}\sum_{i=1}^{n} I(f(\boldsymbol{x_i}) \neq y_i) \text{ (classification tree)}$$

$$\widehat{R(T)} = R^{TS}(T) = \frac{1}{n}\sum_{i=1}^{n} (f(\boldsymbol{x_i}) - y_i))^2 \text{ (regression tree)}$$

where *n* is number of samples in the testing set and *I* is indicator function

Estimate based on the V-fold cross validation

- Random split of data to V equally large sets
- Growing tree based on V-1 sets
- Test the model error based on the V-th set
- Repeat V times so each set is once used as testing set
- Estimate of model error based on average error of V trees



$$R^{CV}(T) = \frac{1}{V}\sum_{i=1}^{V} R^{TS}(T_i)$$

where T_i is *i*-th tree

CART - pruning

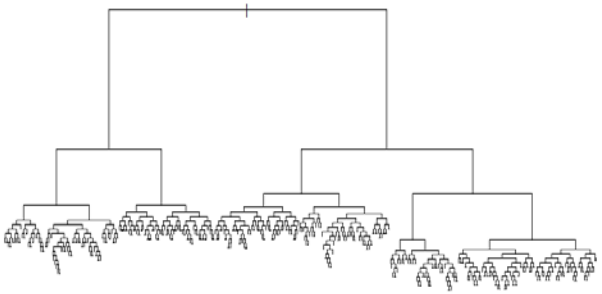
Pruning principle

- Pruning the tree = reduction of tree model by pruning selected branches
- The goal is to grow an optimal tree
- Pruning removes branches with low added value ~ low decrease in information criteria
- Cost complexity function
  - Enhanced approach with penalization of larger trees

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}|,$$

where $|\widetilde{T}|$ is number of leaves and α complexity cost parameter

  - Selection of α depends on the underlying problem



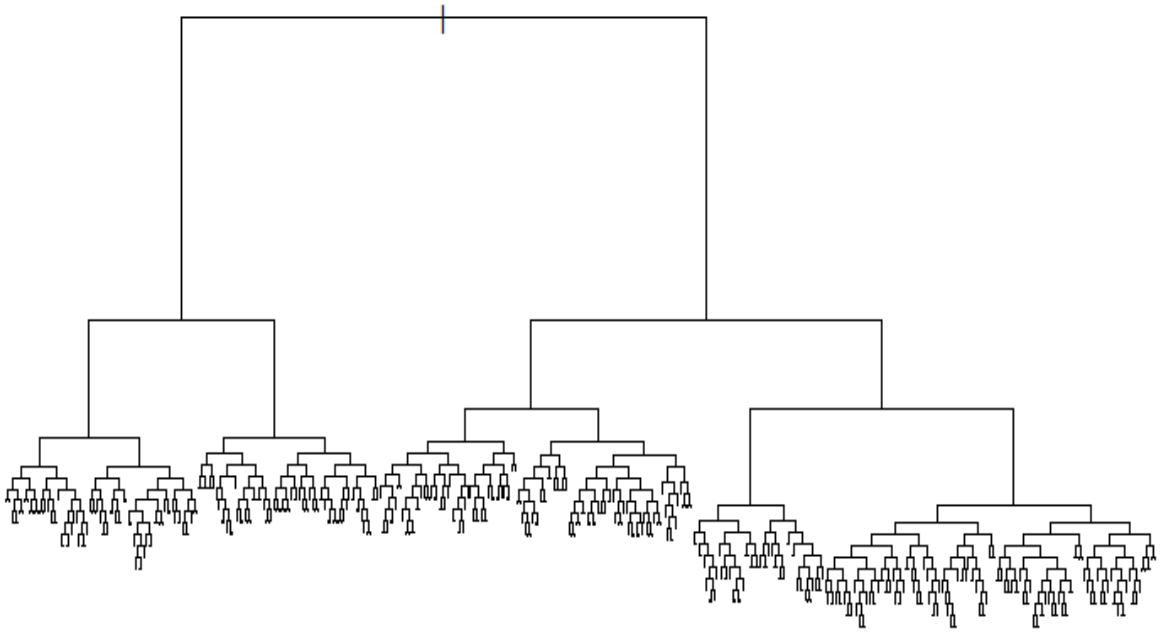
Application of pruning

- Pruning the tree based on the cross validation estimate of error
  - Minimum R^{CV} rule
  - Minimum R^{CV} plus 1 SE rule (standard error)
- Interactive pruning the tree

- Advanced tricks to grow optimal tree - combining growing/pruning based on the training/testing data

CART – cross validation

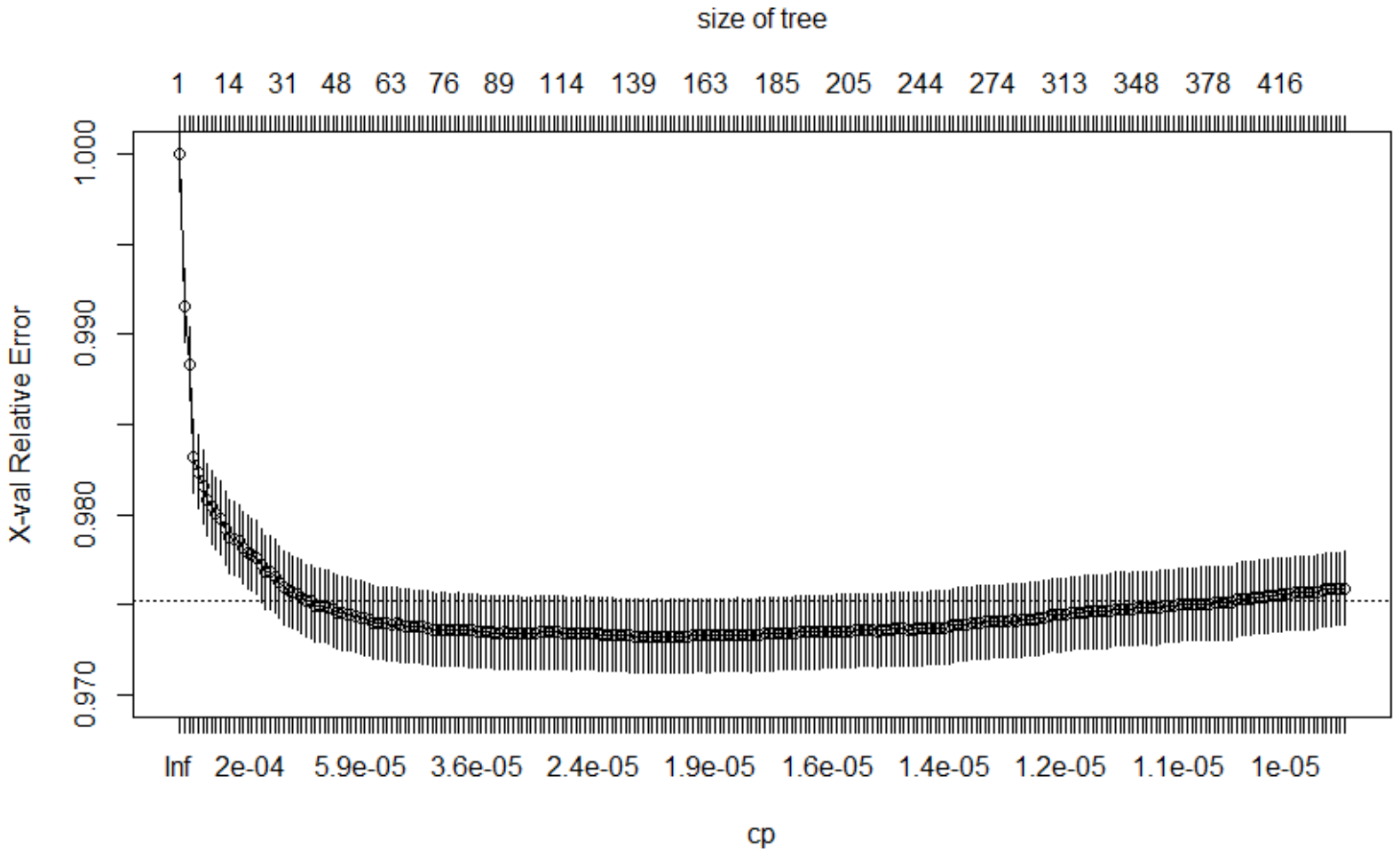
Generating a large (overfitted) tree using minimal stopping criterion. The tree is graphically presented in the following chart:



CART – cross validation

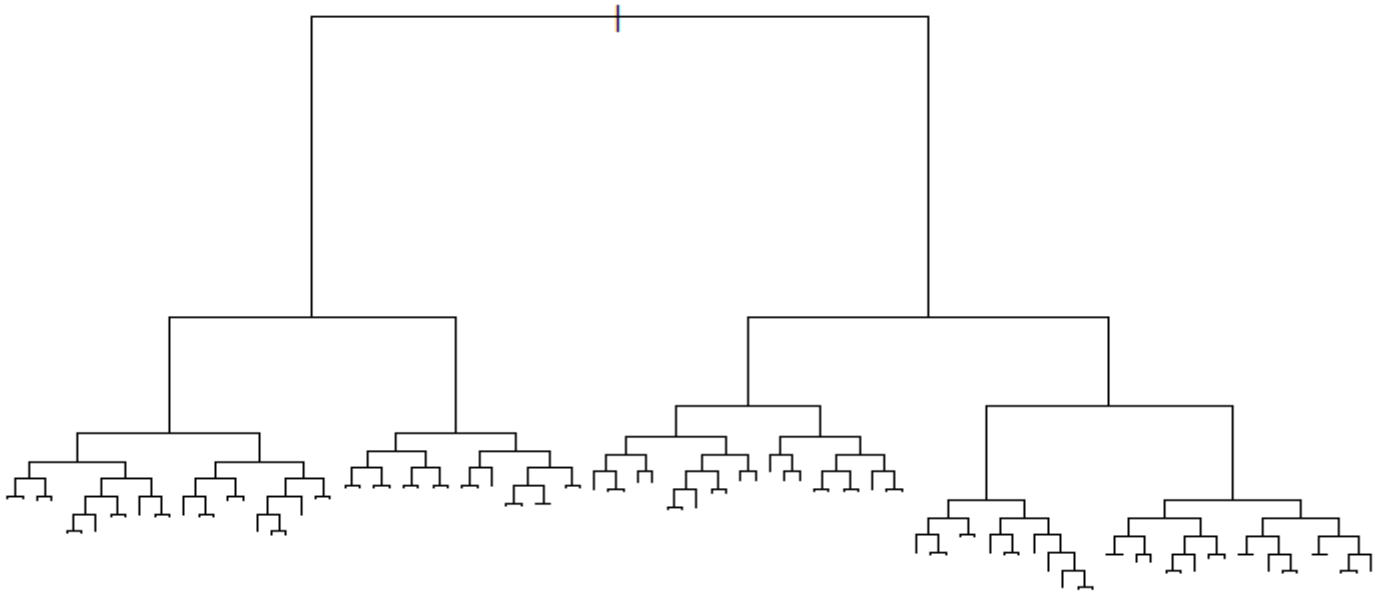
The cross validation plot of the tree is presented in the following chart

The chart indicates overfitting of the model



CART – cross validation

Pruned tree in point that minimizes the cross validation curve



CART– pros and cons

Model features	CART
Fully nonparametric	Yes
Ability to model nonlinear dependency and interactions	Yes
Ability to deal with missing data	Yes
Ability to deal with many variables	Yes
Easy predictions, easy to interpret (important variables, interactions)	Yes
Handling data of mixed types	Yes
Robust to outliers	Yes
Accuracy of results	Moderate
Stability of model in respect to data	No
Extrapolation power	No
Possible overfitting of model	Yes
Identification of global dependency	No

Random forests

Introduction

Selected methods

- CART
- Random forests

Case study

Areas of application

Random forest - principle

Random forest = set of trees, for which:

- Each tree is grown on different bootstrap sample
- Each split in each tree takes into account only a limited number m_{try} of independent variables, other are ignored

Usual number m_{try} of independent variables selected at each split:

- For classification trees $m_{try} = \sqrt{N}$, where *N* is overall number of independent variables
- For regression trees *N/3*
- Bagging is a special case of random forest with m_{try}=*N*

Random forest is not too sensitive to m_{try} , recommended approach is to try usual m_{try} (see above), half and double and choose the best

The predicted value is:

- Class with majority of votes of individual trees for classification forest
- Average value of prediction of individual trees for regression forest

Random forests - Bootstrapping

Bootstrapping approach

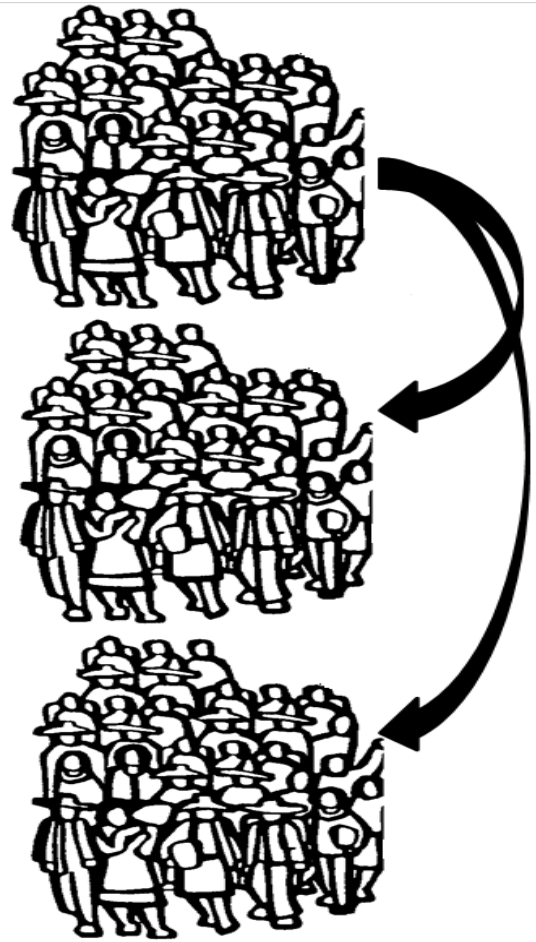
- Given training data $(\boldsymbol{x_1}, y_1)$, $(\boldsymbol{x_2}, y_2)$,…, $(\boldsymbol{x_n}, y_n)$, draw a random set of integers $J_1, J_2, \ldots, J_n$ independently and uniformly from the numbers *1, 2, …, n* with replacement
- Set $(\boldsymbol{x'_i}, y'_i) = (\boldsymbol{x_{J_i}}, y_{J_i})$
- Treat this bootstrap sample as another set of training data and fit a tree to it
- Repeat as many times as needed

The training data must be IID

The bootstrap sample has the same number of observations as the original training data

The bootstrap sample approximates the empirical distribution of training data

Bootstrapping imposes no assumptions about the underlying probability distribution



Random forest - properties

Properties of random forest

- As the number of trees K_T goes to infinity, almost surely

$$E_{\boldsymbol{X},Y}(Y - \frac{1}{K_T}\sum_k f_k\,(\boldsymbol{X}, L_k))^2 \to E_{\boldsymbol{X},Y}(Y - E_L f(\boldsymbol{X}, L))^2$$

where

  - $E_{\boldsymbol{X},Y}(Y - E_L f(\boldsymbol{X}, L))^2$ is defined as generalization error of the random forest *PE(forest)*
  - $f_k(\boldsymbol{X}, L_k)$ is prediction based on *k*-th tree, which was grown on bootstrap sample L_k
  - The left side $E_{\boldsymbol{X},Y}(Y - \frac{1}{K_T}\sum_k f_k\,(\boldsymbol{X}, L_k))^2$ is model error of the random forest with K_T trees
  - Proof is based on the strong law of large numbers (see [3])

- The generalization error of a tree is defined as

$$PE(tree) = E_L E_{\boldsymbol{X},Z}(Y - f(\boldsymbol{X}, L))^2$$

Assume for all *L*, $EY = E_{\boldsymbol{X}} f(\boldsymbol{X}, L)$, then

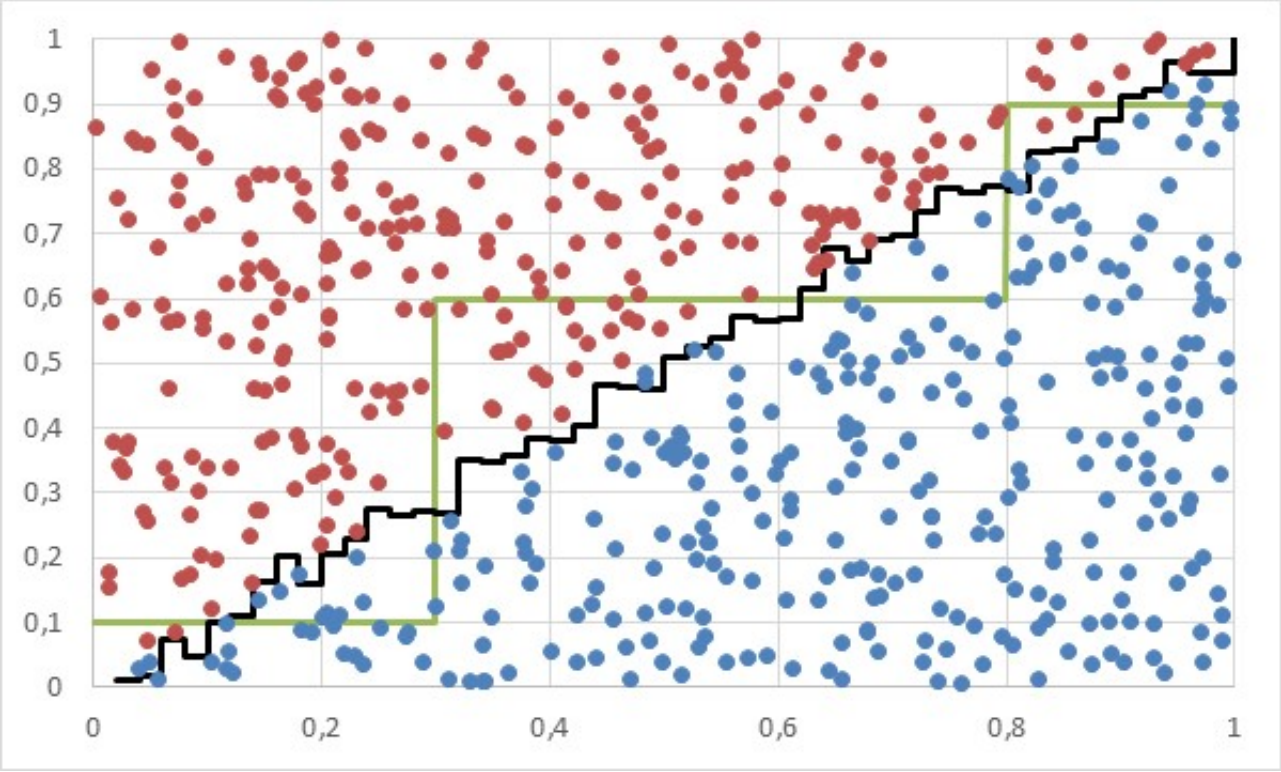
$$PE(forest) \leq \bar{\rho}\, PE(tree),$$

where $\bar{\boldsymbol{\rho}}$ is a weighted correlation between the residuals $Y\text{-}f(\boldsymbol{X}, L`)$ and $Y\text{-}f(\boldsymbol{X}, L)$ and samples *L*, *L`* are independent (proof and details in [3])

Random forest in practice

Practical consequences

- The trees in the forest are not necessary to be pruned
- Lower correlation between trees imposes lower bound to the forest generalization error
- The number of trees to grow – until error no more decreases
- Higher stability of results than a single tree
- More accurate results than a single tree



Random forest – estimation of model error

Estimate of error through out of bag sample (OOB)

- Out of bag sample = training data not used in growing a particular tree (consequent of sampling with replacement)
- Each tree of the forest has its own OOB, generally different from OOB of other trees in the forest
- For each tree approximately 1/3 of data is not used
- The OOB sample can be used as testing set to estimate error for given tree
- For each observation in the training set the average error is calculated from cases when the observation was OOB, the error of random forest model is then estimated as

$$\widehat{R(RF)} = \frac{1}{n}\sum_{i=1}^{n}\frac{\sum_{k=1}^{K_T} I\left((\boldsymbol{x}_i, y_i) \in OOB_k\right)(f(\boldsymbol{x}_i) - y_i)^2}{\sum_{k=1}^{K_T} I\left((\boldsymbol{x}_i, y_i) \in OOB_k\right)}$$

where n is number of observations in training data, K_T number of trees in the random forest and $I\left((\boldsymbol{x}_i, y_i) \in\right.$

Random forests – variable importance

Variable importance index

- Let *I(v(t)=i)* be an indicator function whose value is
  - 1 if *t*-th node splits according to the independent variable *i*,
  - 0 otherwise
- Let d_t^2 be a decrease in MSE (generally any selected function) for including node *t*
- The importance J_i^2 of *i*-th independent variable in the tree is

$$J_i^2 = \sum_{t=1}^{|T^*|} d_t^2 I(v(t) = i)$$

Where |*T**| is the number of nodes in the tree

- For forest average J_i^2 is used over all trees in the forest

Partial dependency plots

- One way partial dependence plot for *i*-th independent variable
- i-th independent variable is shown against the estimate of expected value of independent variable for given for $X_i=x_i$

Random forest – variable importance

Based on OOB estimate of model error:

- For each independent variable X_i, $i = 1,\ldots,N$
  - Randomly permute *i*-th variable in training data to generate a new set of samples
  - Calculate OOB estimate of prediction error with the new sample
  - The measure of importance of random variable X_i is the increase in error between predictions based on permuted and original OOB

Proximity matrix

- Record when $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ appears in the same node and normalize by number of trees in the forest -> record the result in the matrix (proximity matrix)
- Proximity matrix can be considered as a kind of similarity measure, can be used by other methods

CART– pros and cons

Model features	CART	Random forest
Fully nonparametric	Yes	Yes
Ability to model nonlinear dependency and interactions	Yes	Yes
Ability to deal with missing data	Yes	Yes
Ability to deal with many variables	Yes	Yes
Easy predictions, easy to interpret (important variables, interactions)	Yes	Partially
Handling data of mixed types	Yes	Yes
Robust to outliers	Yes	Yes
Accuracy of results	Moderate	High
Stability of model in respect to data	No	Yes
Extrapolation power	No	No
Possible overfitting of model	Yes	No
Identification of global dependency	No	No

Case study

Introduction

Selected methods

- CART
- Random forests

Case study

Areas of application

Case study – data description

- The input data represent a database of insurance policies, which lapsed or did not lapse (1=lapse, 0=not lapsed)
- The database was artificially created based on random generator, binomial distribution was used to simulate the lapse event
- The database contains 200 000 records
- The average lapse rate amounts to app. 20%
- The database contains additional information (variables) for each policy. The variables are presented in the table below:

Variable	Description
A	Age of policyholder at entry
M	Marital status
E	Earnings of the policy holder
T	Type of insurance
I	Sum insured
O	Office of underwriting
Y	Year of underwriting
C	Children

The random generator contains interaction of the following variables:

- A:T
- M:T
- C:T
- O:Y

Case study – data description

The values of the variables are split into the following categories:

Variable	Description of categorical parameters
A	A1: 18-29, A2: 30-39, A3: 40-49, A4: 50-59, A5: 60+
M	M0: single/divorced, M1: married
E	E1: <10; E2: 10-20; E3: 20-30; E4:30+ (thousands CZK)
T	T1: whole life non-profit, T2: whole life with-profit, T3: endowment non-profit, T4: endowment with-profit, T5: unit link
I	I1: 0-500, I2: 500-1000, I3: 1000+ (thousands CZK)
O	O1, O2, O3, O4, O5
Y	Y1: 96-97, Y2: 98-99, Y3: 00-01, Y4: 02-03, Y5: 04-05
C	C0: no children, C1: some children

Case study –models applied

- In order to construct and test the models, the database was randomly split into two subsets:
  - Training data to produce the models (100 000 observations)
  - Testing data to validate the models (100 000 observations)
- Following models were produced
  - 3 simple tree models based on CART (1 large tree subsequently pruned twice)
  - Several random forests models based on different input parameters, selected final model had the lowest estimated model error estimate
  - Reference GLM model was fitted based on the training data as well to achieve comparability of the results
- All models were subsequently tested on the testing data
- The construction of the models was based on the following assumptions/models:
  - No cleaning of the input data was carried out as all database records are assumed to be reliable, no data are missing
  - All variables contained in the database may have an effect on the lapse rate
  - The tree based models ware created based on software implemented in R software
  - GLM model took into account binomial distribution of input data and interactions of independent variables

Case study – comparison of results

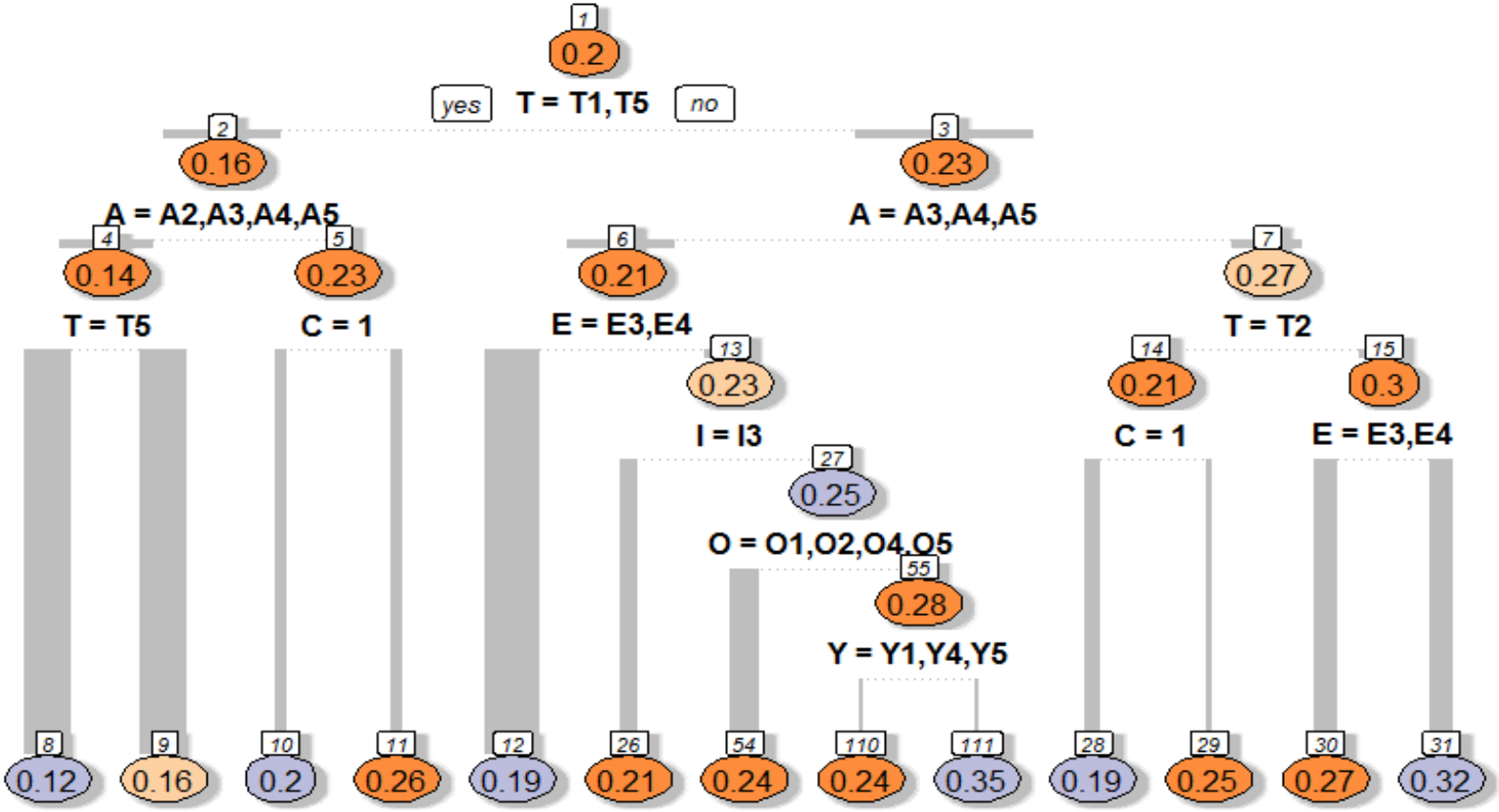
Comparison of MSE of particular models				
Model	Number of nodes (incl leaves)	Depth	MSE – testing data	Success rate
CART 1	413	14	0.160022	44%
CART 2	63	7	0.159289	59%
CART 3	23	5	0.159560	54%
Random Forest	NA (100 trees)	NA	0.158503	75%
GLM	NA	NA	0.158142	82%
Pure average model	1	1	0.162228	0%
Variance of data	NA	NA	0.157248	100%

Success rate is calculated as decrease in model MSE in comparison with pure average model, relatively to maximal possible decrease (given by variance of underlying distribution)

- 0% success means MSE of the pure average model
- 100% means MSE of variance of theoretical distribution

Case study – tree model

Final/most pruned tree model based on CART method is presented below (CART 2)



Case study – tree model

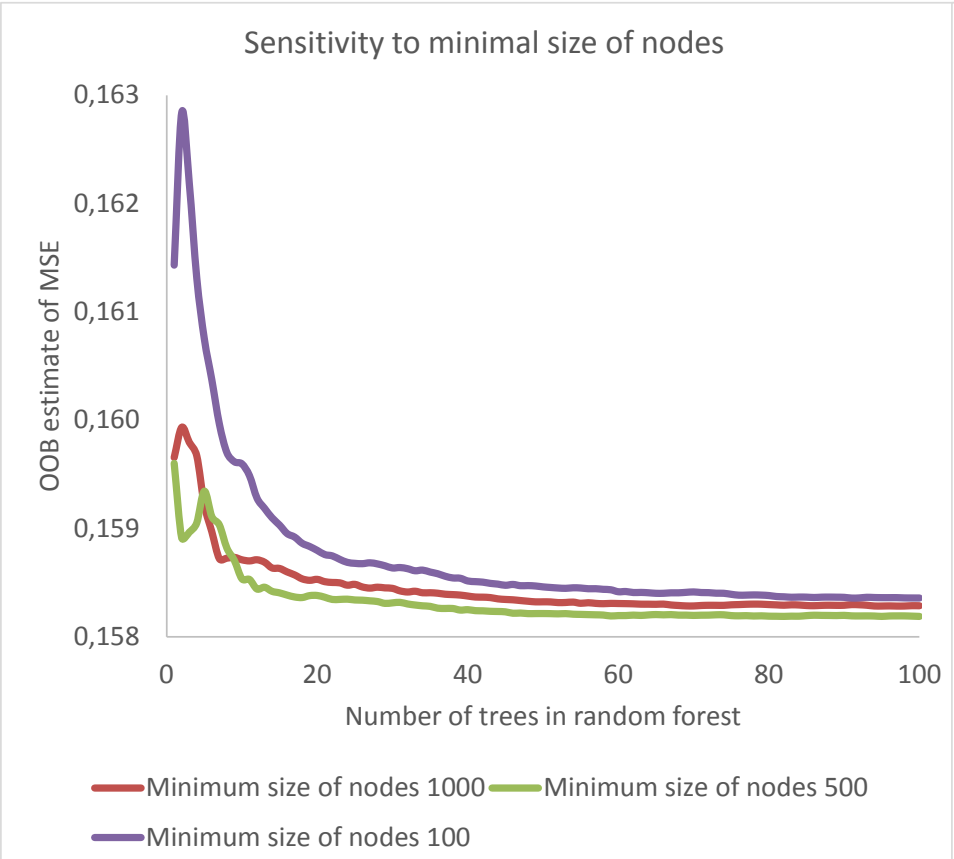
Interpretations of results (note that the database is artificially created and hence these results are not relevant for real business; the interpretations are based on the previous tree plot):

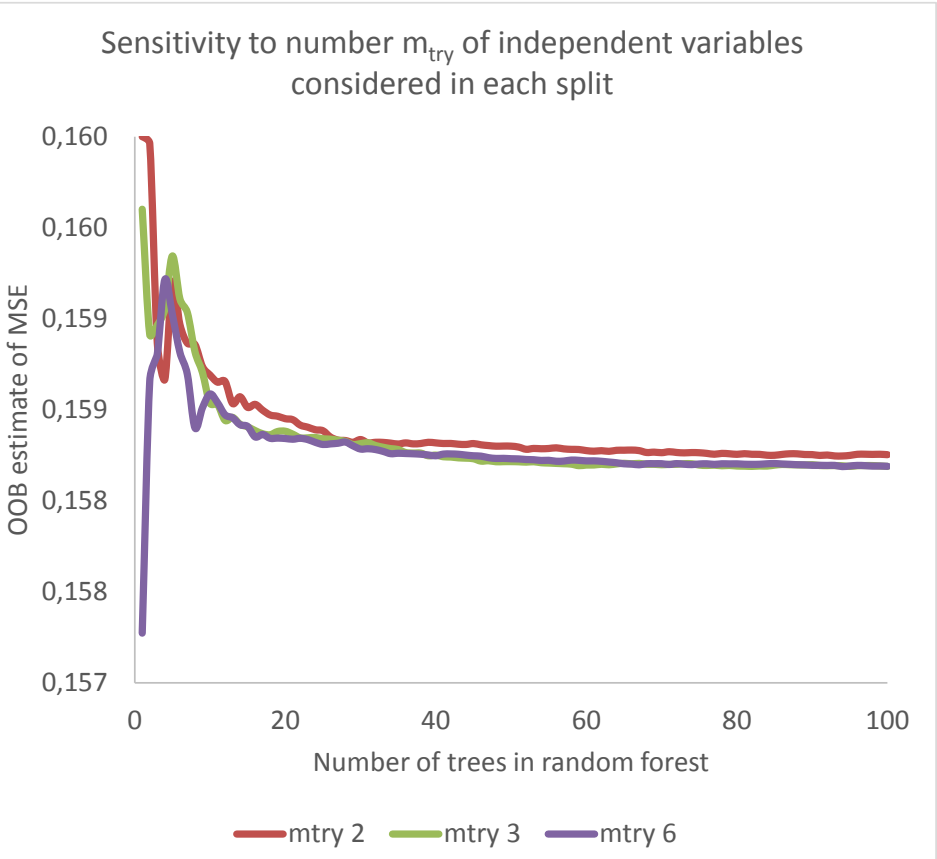
- The lapse rate depends significantly on the product type. The endowment products have significantly higher lapse rate than unit linked and WLNP
- The most significant interaction between the variables is between the policyholder's age and type of product
- Policyholders with children have lower lapse rate, this is significant especially for younger policyholders
- Young policyholders have higher lapse rate than older policyholders
- Policyholders with higher earnings have lower lapse rates - for endowment and WLWP products

Case study – random forest sensitivity

The sensitivity analysis of random forest relates to:

- Number m_{try} of independent variables considered in each split
- Minimal size of node
- Number of trees in the forest

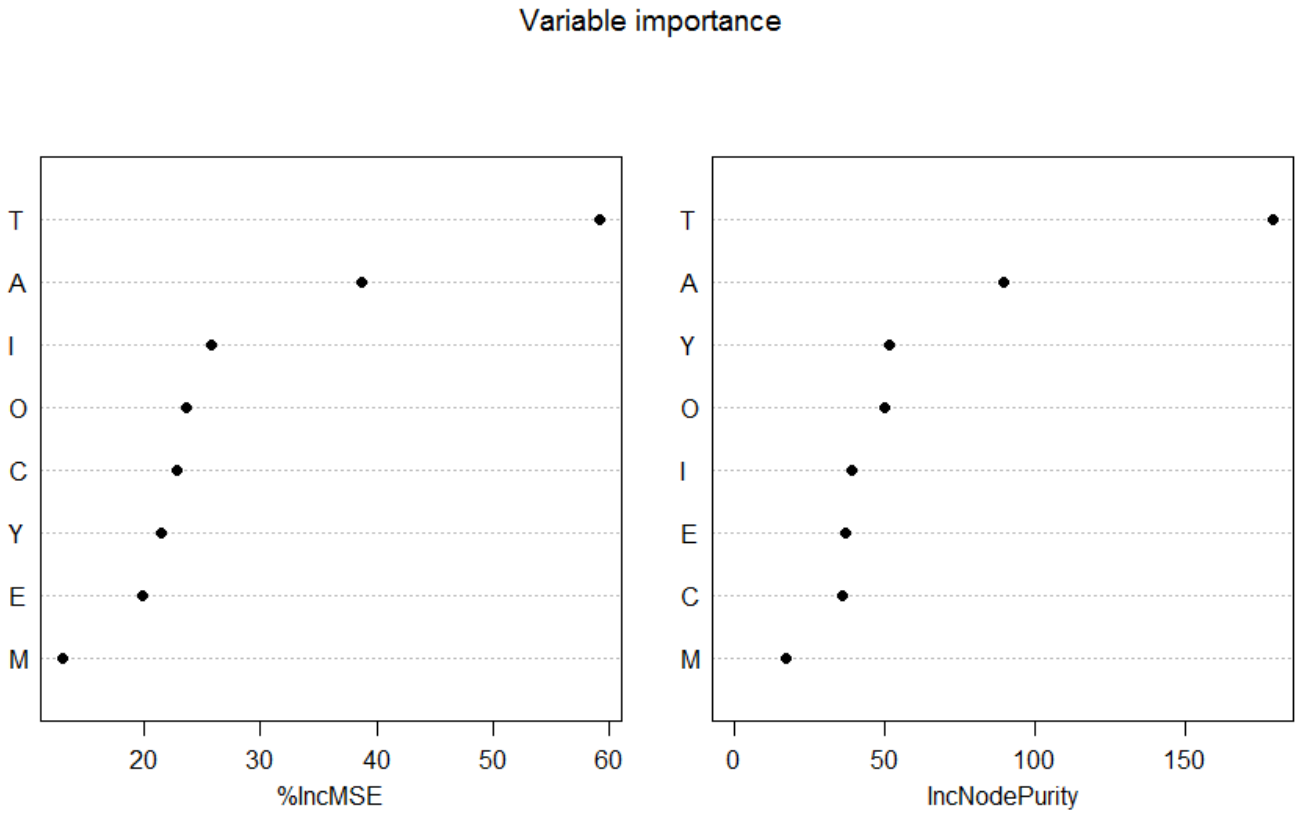




Case study – variable importance in random forest

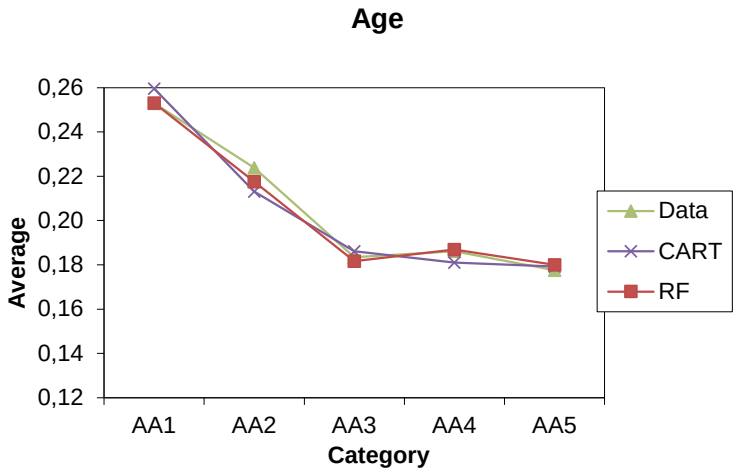
The following chart contains variable importance analysis according to:

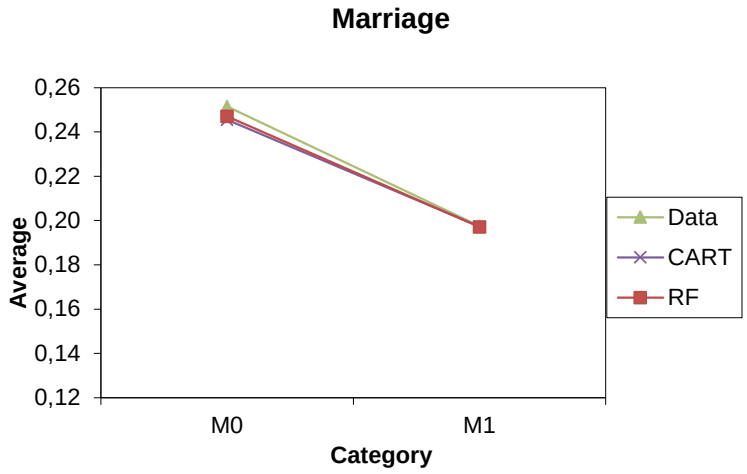
- % Increase in MSE due to permutation of independent variable
- Decrease in sum of squares in nodes split by independent variable

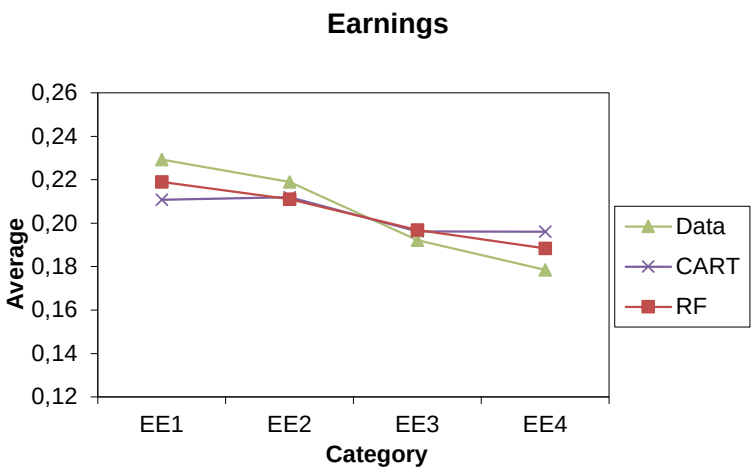


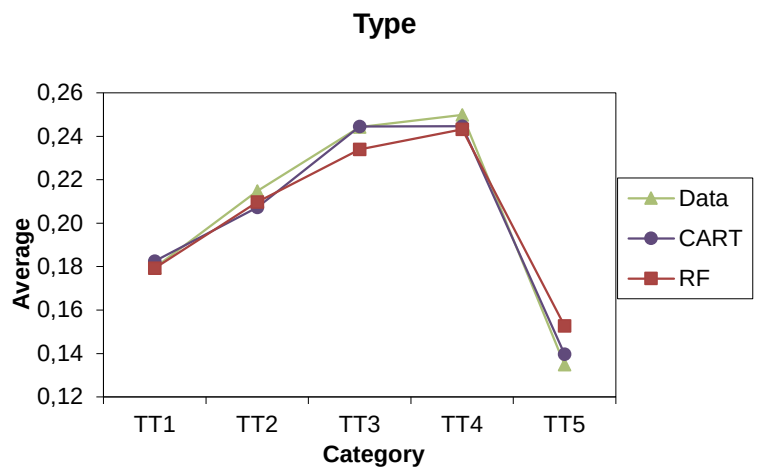
Case study – partial dependency plots

- Partial dependency plots for testing data, CART 1 and random forest model



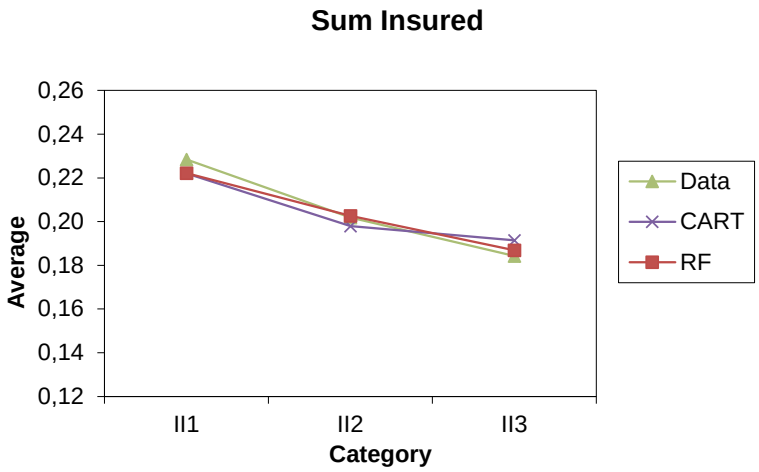


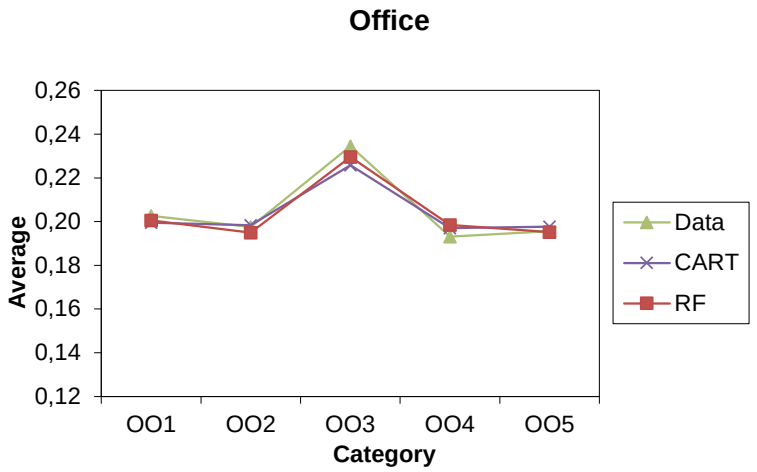


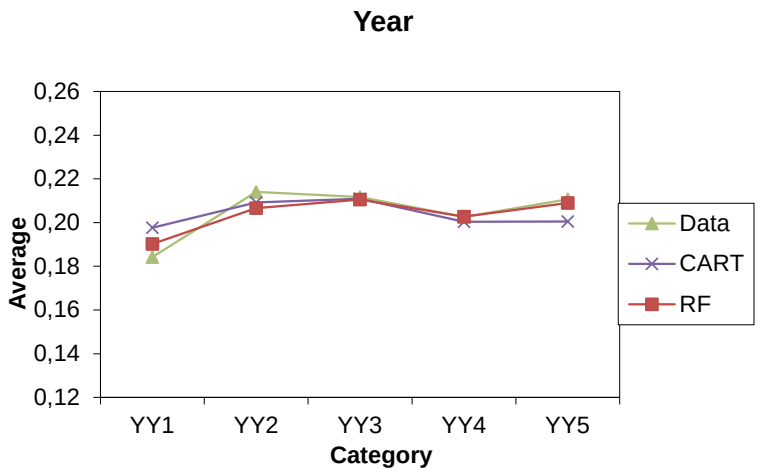


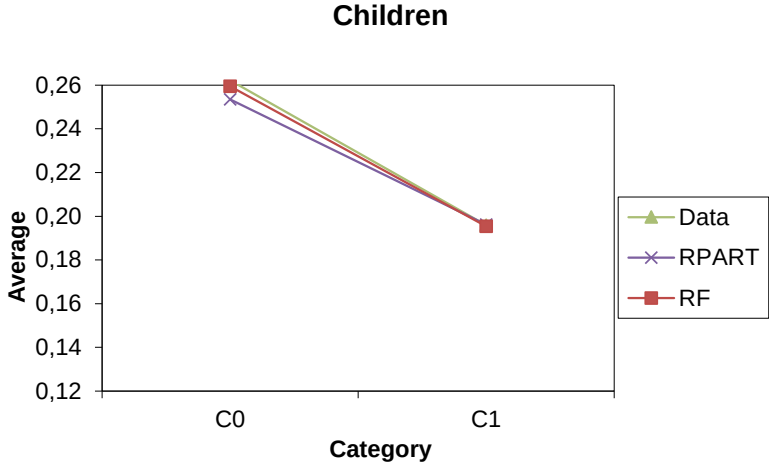
Case study – partial dependency plots

- Partial dependency plots for testing data, CART and random forest model









Case study – comparison of results

In order to compare fit of both models for particular combinations of variables, MSE for each segment S_i of the database are compared (segment is a set of all records with identical combination of values of independent variables):

$$MSE_i = \frac{1}{|S_i|} \sum_{\boldsymbol{x}_j \in S_i} (y_j - f(\boldsymbol{x}_j))^2$$

where:

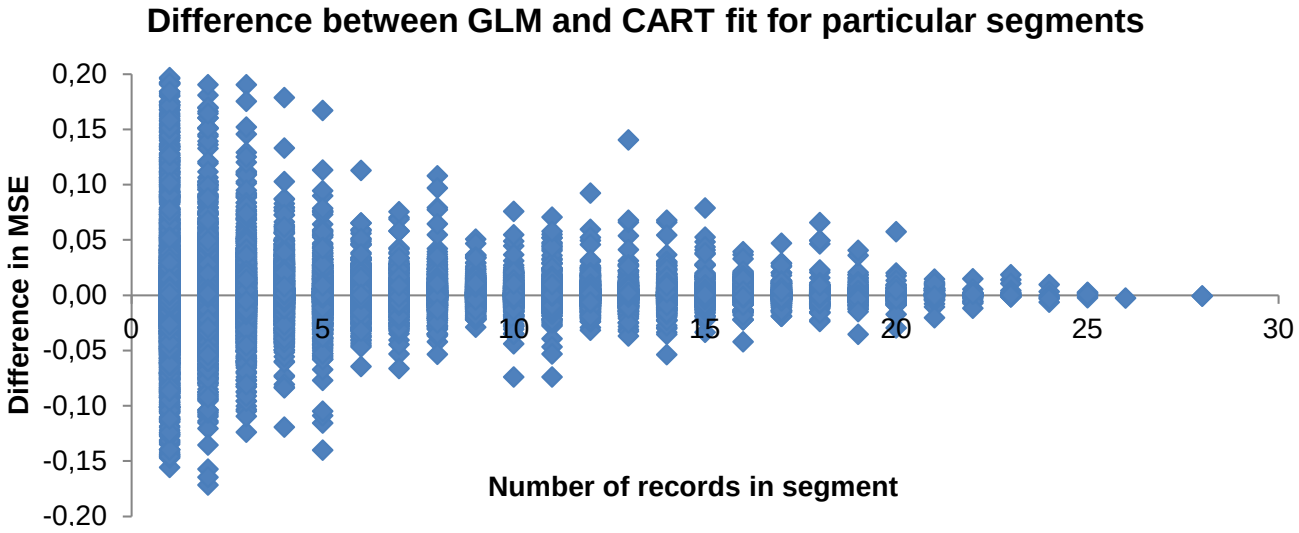
- $|S_i|$ is number of records in segment *i*
- y_j is lapse for *j*-th observations
- $f(\boldsymbol{x}_j)$ is predicted lapse rate in *j*-th segment for GLM/tree models

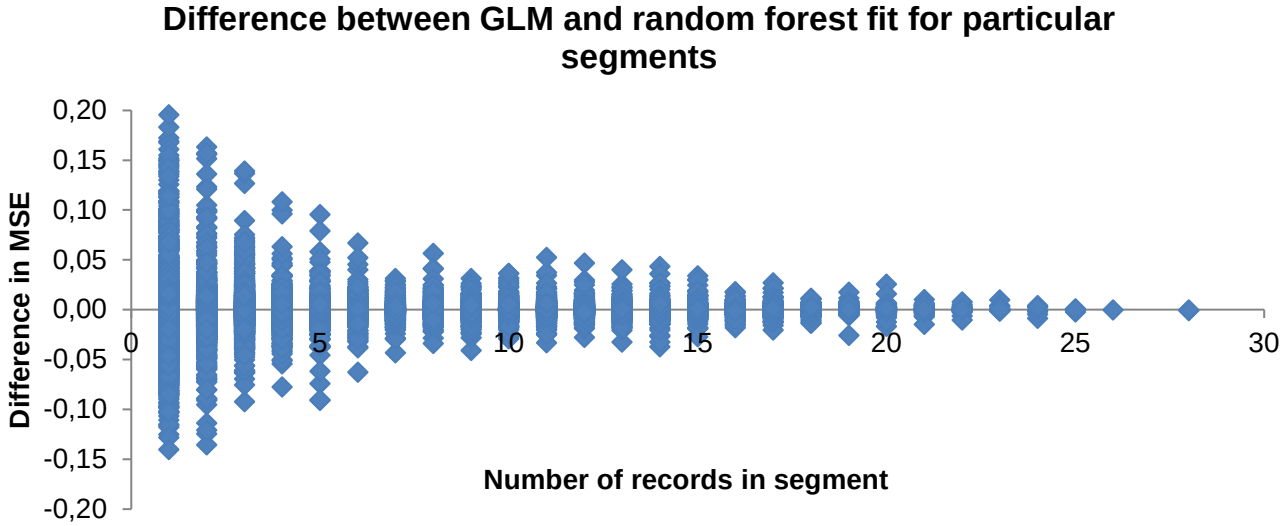
Also the statistics N_iMSE_i is used for each segment, which represents share of *i*-th segment on the overall error of the data model:

$$MSE = \frac{\sum_i n_i MSE_i}{n}$$

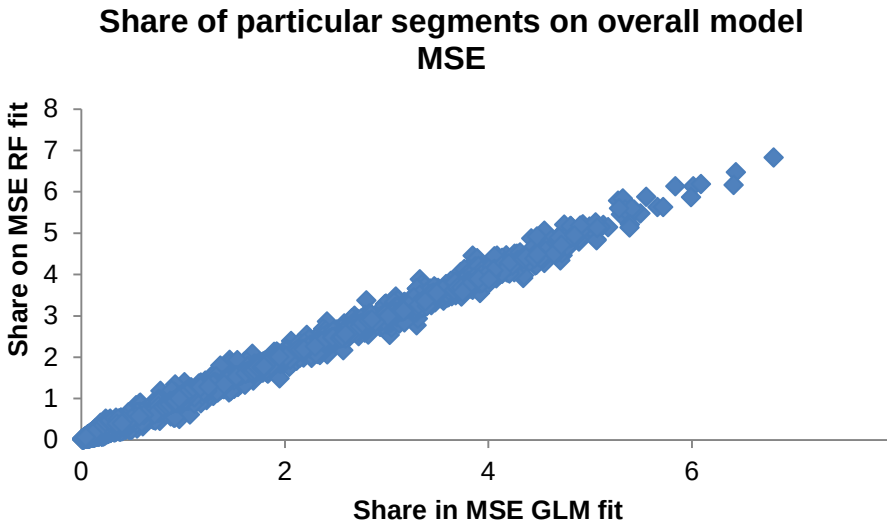
where *n* is number of testing data

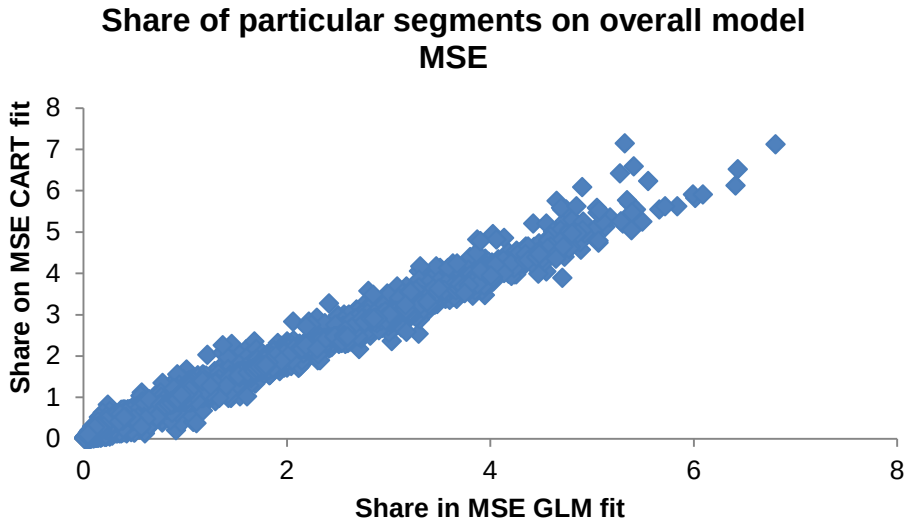
Case study – comparison of GLM and tree model

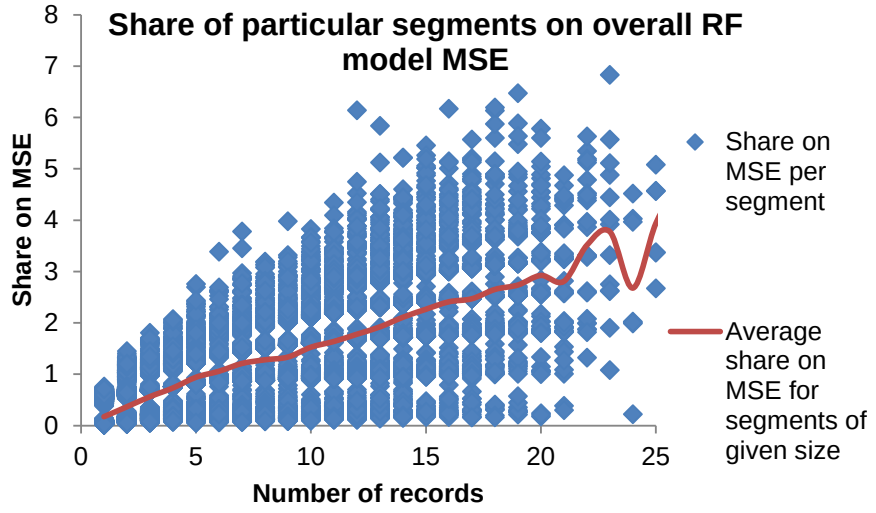


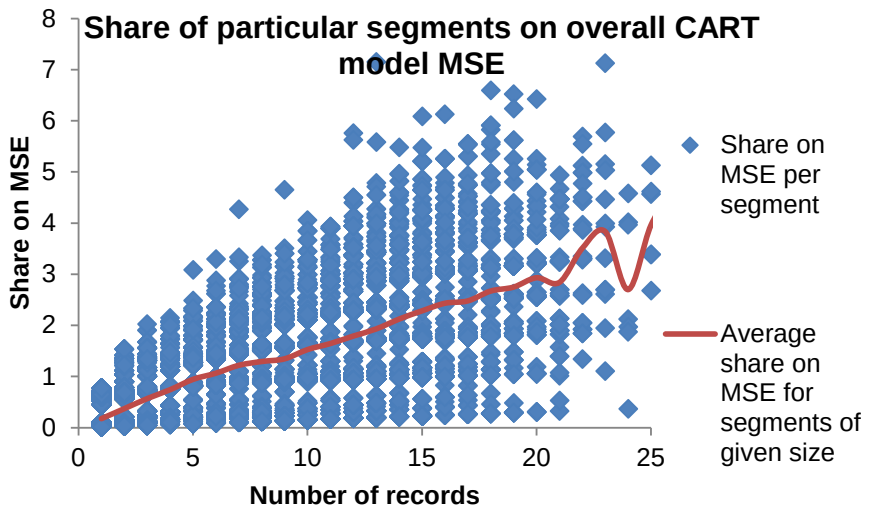


Case study – comparison of GLM and tree model









Areas of application

Introduction

Selected methods

- CART
- Random forests

Case study

Areas of application

Areas of application

Product development

- Pricing
- Design of product features

Claim handling process

- Estimation of RBNS (provision related to reported but not settled claims)
- Estimation claim frequency and claim severity

Approach to client

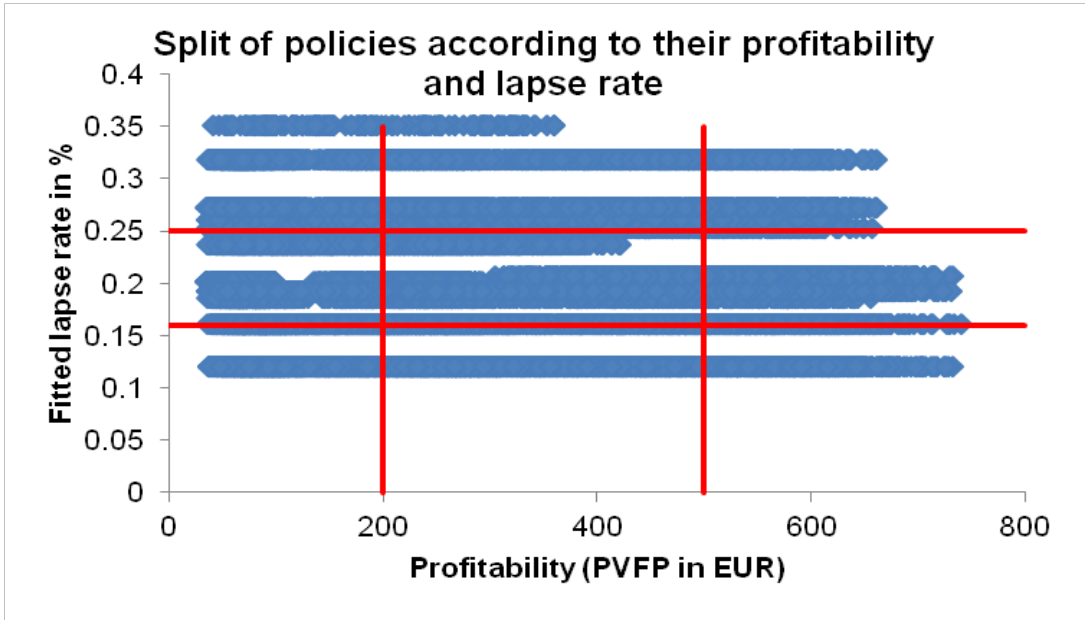
- Retention analyses
- Marketing campaigns
- Optimization of distribution channels

Fraud detection

Areas of application - example

Potential application of lapse analysis

- The analysis of lapses rates enriched by analysis of profitability provides a better ground for future steps
- The following chart split the policies according to the expected lapse rate and profitability of contract measured by PVFP (present value of future profit)
- PVFP is randomly generated, assuming profitability is correlated with sum insured, policyholder's age and product type)



Examples of future actions:

- Focusing retention campaigns on policies with high profitability an high probability of lapse
- Focusing marketing/selling campaigns on policies with high profitability and low probability of lapse
- Tailored offers & product development for specific segments

KPMG
cutting through complexity

Thank you

Presentation by Robert Meixner

Leo Breiman:

"RF is an example of a tool that is useful in doing analyses of scientific data.

But the cleverest algorithms are no substitute for human intelligence and knowledge of the data in the problem.

Take the output of random forests not as absolute truth, but as smart computer generated guesses that may be helpful in leading to a deeper understanding of the problem. "

Literature

[1] Leo Breiman et all: Classification and Regression Trees, 1984

[2] Cosma Shalizi: Classification and Regression Trees, 2009

http://www.stat.cmu.edu/~cshalizi/350/lectures/22/lecture-22.pdf

[3] Leo Breiman: Random Forersts, 2001

https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

[4] Jerome H. Friedman: Greedy Function Approximation: A gradient Boosting Machine, 1999

http://statweb.stanford.edu/~jhf/ftp/trebst.pdf