Research Article **Open Access**

Zhiyu Quan and Emiliano A. Valdez*

Predictive analytics of insurance claims using multivariate decision trees

https://doi.org/10.1515/demo-2018-0022
Received July 18, 2018; accepted December 5, 2018

Abstract: Because of its many advantages, the use of decision trees has become an increasingly popular alternative predictive tool for building classification and regression models. Its origins date back for about five decades where the algorithm can be broadly described by repeatedly partitioning the regions of the explanatory variables and thereby creating a tree-based model for predicting the response. Innovations to the original methods, such as random forests and gradient boosting, have further improved the capabilities of using decision trees as a predictive model. In addition, the extension of using decision trees with multivariate response variables started to develop and it is the purpose of this paper to apply multivariate tree models to insurance claims data with correlated responses. This extension to multivariate response variables inherits several advantages of the univariate decision tree models such as distribution-free feature, ability to rank essential explanatory variables, and high predictive accuracy, to name a few. To illustrate the approach, we analyze a dataset drawn from the Wisconsin Local Government Property Insurance Fund (LGPIF) which offers multi-line insurance coverage of property, motor vehicle, and contractors' equipments. With multivariate tree models, we are able to capture the inherent relationship among the response variables and we find that the marginal predictive model based on multivariate trees is an improvement in prediction accuracy from that based on simply the univariate trees.

Keywords: Tree-based models, univariate regression trees, random forests, gradient boosting, multivariate regression trees, multivariate tree boosting, predictive model of insurance claims

1 Introduction

A decision tree model, with origins that date back to early 1960's, is a data mining algorithm that can broadly be described by repeatedly partitioning the regions of the explanatory variables and thereby creating a tree-based model for predicting the response. [25] developed the very first naive regression tree algorithm and called it the Automatic Interaction Detection (AID). For more details about the historical development of decision trees including alternative algorithms such as GUIDE and C4.5 algorithms, see [21]. Today, the use of decision trees has become an increasingly popular alternative predictive tool for building classification and regression models. Considered a supervised learning technique, it has many advantages which are especially important for analyzing actuarial data.

First, a decision tree model is considered to be nonparametric and thereby does not require distribution assumptions. Unlike classical statistical methods, decision tree models do not require the input of any probability distributions about the response. Second, it is an effective algorithm that can handle missing data. For many real datasets, the absence and the unrecording of some information is not uncommon. Third, apart from the ability to handle missing data, it can detect non-linear effects and possible interactions among the explanatory variables. Traditional linear models typically capture only linear effects, and detection for non-

Zhiyu Quan: Department of Mathematics, University of Connecticut, E-mail: zhiyu.quan@uconn.edu
***Corresponding Author: Emiliano A. Valdez:** Department of Mathematics, University of Connecticut, E-mail: emiliano.valdez@uconn.edu

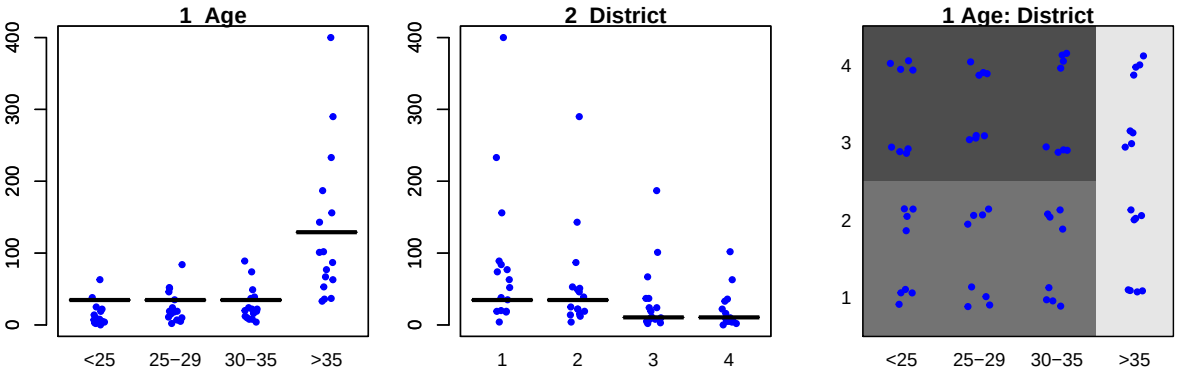

Figure 1: Insurance claims: segmentation of the explanatory variables

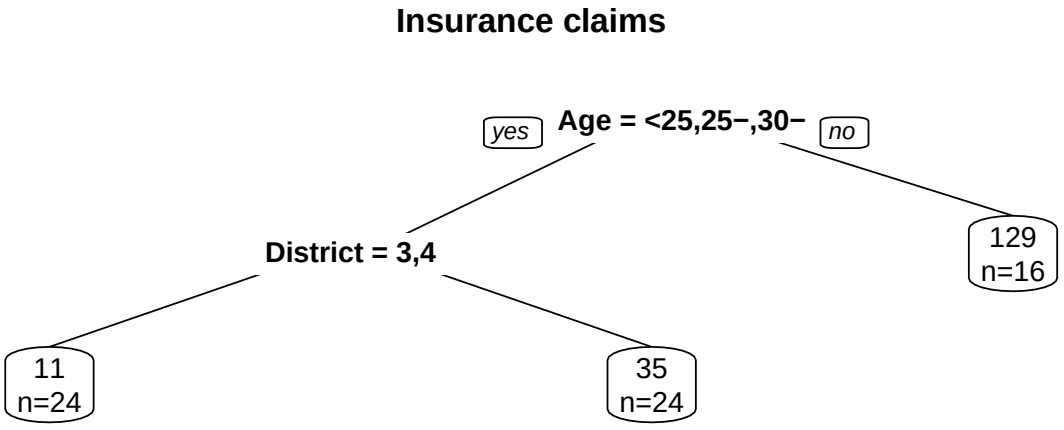

Figure 2: Insurance claims: segmentation of the explanatory variables

linearity as well as interactions requires further analysis. Fourth, it can be considered as a variable selection procedure by assessing the relative importance of the explanatory variables. Such variable selection is usually important in actuarial science for purposes such as risk classification and collection of risk variables. Finally, decision trees, especially with smaller-sized trees, are straightforward to interpret by a visualization of the tree structure in the plot. These advantages are particularly useful for actuarial and insurance data. See [34].

It should be pointed out that comparing prediction accuracy between traditional linear models and decision tree models (and their ensemble methods) may be unsuitable due to the manner in which rules and principles are applied. However, in practice, it is understandable to make such comparison in order to better evaluate the quality between models. Indeed, in the literature where applications are emphasized, there have been some papers that make direct comparison of prediction accuracy between traditional statistical models and machine learning algorithms. For example, see [22], [26], and [37].

For a simple illustration of how regression trees are constructed, consider the insurance claims data obtained from the R package MASS. Here we have 64 observed claims and we consider three potential explanatory variables: Age, District, and Group. In Figure 1, we show how the amount of claims are segmented by Age and District. Note that Group is omitted in this figure because it is not considered significant. Figure 2 shows the final structure of the decision tree. This tree corresponds to the segmentation of Age and District as demonstrated in Figure 1 where it clearly makes the separation according to shade. This illustrates a simple diagram of the separation of nodes within a regression tree model. We can conclude from these figures that claims are generally significantly larger for ages 35 and above, while districts 3 and 4 have slightly lower claims than districts 1 and 2. For similar figures of how decision trees are constructed for classification, see [21] and [34].

The early methods of decision trees have the potential disadvantages of producing irregular patterns resulting in overfitting and bias in variable selection. Innovations and extensions to the original methods, such as random forests and gradient boosting, further improved the capabilities of using decision trees as a predictive model. Random forest refers to the process of generating ensembles of trees with a set of unpruned fully-grown trees. These trees are generated based on a bootstrap sampling of the original data and using a subsample of the explanatory variables. [1] showed that the use of random forests led to significant improvements in prediction accuracy.

Boosting algorithms have increased in popularity in machine learning because they help to find a good balance between bias and variance through the tuning parameters. For decision tree models, boosting algorithms build trees sequentially so that for each new iteration, a tree is grown using the residuals from previously grown trees. This procedure combines weak learners to produce strong learner. Early methods of boosting decision trees, as discussed in [10], used optimization based on gradient descent algortihms and this gave rise to the term *gradient boosting*.

Here we cite some interesting applications of decision tree models in actuarial science and insurance. Interestingly, for example, [27] provided an alternative look of the life table construction using tree-based models and concluded that tree-based methods have inherent characteristics that capture intrinsic data structure useful for identifying primary risk factors. In their lecture notes on data analytics for non-life insurance pricing, [39] used classification trees to determine whether a policy belongs to a male or female driver given some policy characteristics. [13] used the idea of gradient boosting (GB) to predict auto accident loss cost and concluded that this method provided more superior predictive accuracy than that of traditional Generalized Linear Models (GLMs). [19] introduced Delta Boosting (DB) as an alternative boosting algorithm and showed that this algorithm is optimal under a variety of loss functions. Using claims data on collision coverage for vehicle insurance from a Canadian insurer, the article also demonstrated that the DB algorithm outperforms the GB algorithm. [38] applied classification and regression trees to calculate reserves on individual claims data. [4] applied the Poisson regression tree and its boosting ensemble to examine the quality of mortality models in understanding different causes of death.

As evident from our previous discussion, applications of decision tree models have practically been based on a single-valued response variable. This paper extends the concept of decision trees in the case where we have a multiple-valued, or multivariate, response variable. In the literature in recent years, we have seen a large potential for actuarial and insurance applications where we encounter multivariate responses. To illustrate, here are some sources of dependencies that we often encounter in actuarial and insurance problems: (a) a single policyholder may have several insurance coverages, (b) a policyholder may own a bundle of insurance contracts such as homeowners and automobile, (c) a taxicab company may own an umbrella coverage for several automobiles, (d) a corporation may own several types of insurances for its employees such as workers compensation and health insurance, and (e) time and spatial dependencies are typical insurance data structures.

To demonstrate the potential benefits of fitting multivariate decision trees, we examine a multivariate response variable with six different components; each component is associated with one of the six types of property and casualty insurance coverages for local government units established by the Wisconsin Local Government Property Insurance Fund (LGPIF). Earlier works on this dataset have used the concept of parametric copulas to analyze the multivariate structure of the data. Our paper examines the benefits of using multivariate decision tree models without having to specify probability distributions. Yet another advantage of using decision tree models is avoiding the use of a two-part frequency-severity model. When compared to univariate decision tree models, we find that multivariate decision tree models have generally a better predictive accuracy.

The remainder of this paper has been organized as follows. In Section 2, we discuss the concept of univariate regressions and its extensions. In section 3, we describe the concept of decision trees when the response variable is multivariate. In Section 4, we describe the dataset used for our empirical investigation and provide some preliminary data exploration. Results of model calibration and model validation are presented in Section 5 and Section 6, respectively. Finally, we provide concluding remarks in Section 7.

2 Univariate decision trees and its extensions

In this section, we introduce the concept of the univariate regression tree and its extensions. Here we assume that we have a dataset consisting of a vector of p explanatory variables, denoted by $\mathbf{x}_i = (x_{i1}, x_{i2} \ldots, x_{ip})$ and a response variable, y_i, for each of N observations. This dataset is best represented as $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, N$. For our paper, we discuss the three of the most widely used univariate decision trees: CART (Classification and Regression Tree), random forests, and gradient boosted regression trees.

2.1 CART (Classification and Regression Trees)

First introduced by [2], this method uses recursive partitioning to build decision trees applicable for predicting either a continuous response, in the case of regression, or a categorical response, in the case of classification. In our subsequent discussion, we focus on regression trees for predicting continuous response variable. We adopt notation from [15]. In this algorithm, a regression tree, denoted by $T(\mathbf{x}, \Theta)$, is produced by partitioning the space of the explanatory variables into M disjoint regions $R_1, R_2, \ldots, R_M$ and then assigning a constant c_m for each region R_m, for $m = 1, 2, \ldots, M$. Given a regression tree, each observation can then be modeled based on the expression

$$f(\mathbf{x}_i|\Theta) = \sum_{m=1}^{M} c_m \mathbf{1}_{R_m}(\mathbf{x}_i), \tag{1}$$

where $\Theta = \{R_m, c_m\}_{m=1}^{M}$ denotes the partition with the assigned constants. Under CART, the constants c_m are determined by minimizing the sum of squares error (SSE) loss function:

$$L(y_i, \widehat{y}_i) = \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2, \tag{2}$$

where $\widehat{y}_i = \widehat{f}(\mathbf{x}_i|\Theta) = \sum_{m=1}^{M} \widehat{c}_m \mathbf{1}_{R_m}(\mathbf{x}_i)$ is the predicted value of the response variable. It can be shown that the optimal value, $\hat{c}_m$, is the average of y_i in the region R_m:

$$\widehat{c}_m = \text{average}(y_i|\mathbf{x}_i \in R_m) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} y_i, \tag{3}$$

where N_m is the number of observations in region R_m.

The regions in the regression tree are determined according to an algorithm called recursive binary splitting. The initial step in this algorithm is to find one explanatory variable $X_{\cdot j}$ which best divides the data into two subregions, for example, $R_1(j, s) = \{\mathbf{x}_i|X_{\cdot j} < s\}$ and $R_2(j, s) = \{\mathbf{x}_i|X_{\cdot j} \geq s\}$ in the case of a continuous explanatory variable. This division is determined as the solution to

$$\underset{j,s}{\text{argmin}} \sum_{i:\mathbf{x}_i \in R_1(j,s)} (y_i - \hat{c}_{R_1(j,s)})^2 + \sum_{i:\mathbf{x}_i \in R_(j,s)} (y_i - \hat{c}_{R_2(j,s)})^2, \quad \text{for any } j \text{ and } s.$$

Subsequently, the algorithm looks for the next explanatory variable with the best division into two subregions and this process is applied recursively until reaching a minimum size of observations in the terminal region or some other predefined threshold. The algorithm can handle other types of numerical explanatory variables, such as those with rank order, as well as categorical variables. Furthermore, regression trees can deal with missing values in the explanatory variables using 'surrogate splitting' which involves finding a surrogate variable that best approximates the original split. Observations with missing values are assigned according to the split on the surrogate variable rather than on the original splitting variable.

For many instances, the result is a fully grown tree, T_0, with many terminal regions that may lead to overfitting and unnecessary model complexity. This complexity can be controlled by using *cost-complexity*

pruning to trim the fully grown tree T_0. From equations (2) and (3), define the loss in region R_m by

$$L_m(T) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \hat{c}_m)^2.$$

For any subtree $T \subset T_0$, denote the number of terminal regions in this subtree by $|T|$. To control the number of terminal regions, we introduce the tuning parameter $\alpha \geq 0$ to the loss function by defining the new cost function as

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m L_m(T) + \alpha|T|.$$

Clearly according to this cost function, the tuning parameter penalizes large number of regions. The idea then is to find the subtrees $T_\alpha \subset T_0$ for each α, and choose the subtree that minimizes $C_\alpha(T)$. Furthermore, the tuning parameter α governs the tradeoff between size of the tree and its goodness of fit to the data similar to the regularization parameter in a penalized regression. Large values of α result in smaller trees and as the notation suggests, $\alpha = 0$ leads to the fully grown tree T_0. The estimation of this tuning parameter α is done using K-fold cross-validation.

Algorithm 1 summarizes the details of implementing the CART procedure using the R-package `rpart`. See [17] and [36].

Algorithm 1: CART R-package: rpart

Input: Training dataset $\mathbf{x}$, y, K
Output: Best subtree T_α
1 Grow a full tree T_0 on a training dataset using recursive binary splitting. Use the stopping criterion *minsplit* which is the minimum number of observations in a region for a split to be attempted;
2 Prune the full tree T_0 to subtrees T_α using cost-complexity pruning;
3 Divide the training dataset into K folds to determine the optimal tuning parameter α;
4 **for** $k = 1, \ldots, K$ **do**
5 Repeat steps 1 and 2 on all except for the k-th fold;
6 Compute the mean squared prediction error on the hold out k-th fold using T_α;
7 **end**
8 Average the results for each value of α and pick α that minimizes the average prediction error;
9 Return the best subtree T_α;

2.2 Random forests

Random forest regression was first developed by [1] and it refers to an ensemble of unpruned regression trees $\{T(\mathbf{x}, \Theta_b), b = 1, 2, \ldots, B\}$ which are generated based on a bootstrap sampling from the original training dataset. As a result, we can think of $\{\Theta_b\}$ as independent and identically distributed random vectors. With B as the total number of bootstrap samples, we define the model for the response variable as the average of all the regression trees in the random forest:

$$f_B(\mathbf{x}|\Theta) = \frac{1}{B} \sum_{b=1}^{B} f(\mathbf{x}|\Theta_b)$$

By the Strong Law of Large Numbers, as $B \to \infty$, we have

$$E_{\mathbf{x},y}(y - f_B(\mathbf{x}|\Theta))^2 \to E_{\mathbf{x},y}(y - E_\theta f(\mathbf{x}|\Theta))^2 \quad a.s.$$

Random forests produce a limiting value of the generalization error and this explains why they do not overfit as more trees are added.

For each bootstrap sample, the regression trees are produced using a random subset of explanatory variables in order to decrease the similarity among the trees. The average prediction of multiple regression trees

is expected to have a lower variance than that of individual regression trees. While larger random set of explanatory variables can improve the predictive capability of individual trees, it can also increase the similarity among the trees and could therefore void any gains from averaging multiple predictions. The size of the random subset of explanatory variables can be optimally chosen through cross validation; some use rule of thumbs such as the square root of the total number of explanatory variables, $\sqrt{p}$, in the training dataset.

The bootstrap resampling of the data for training each tree also increases the variation between the trees. The accuracy of a random forest depends on the strength of the individual tree and a measure of the similarity between them.

For implementing random forests in R, see [20]. The procedure to produce random forests is summarized in Algorithm 2.

Algorithm 2: R-package: randomForest

Input: Training dataset $\mathbf{x}$, y, B

Output: $\{T(\mathbf{x}; \Theta_b), b = 1, 2, \ldots, B\}$

1 **for** $b = 1, \ldots, B$ *(ntree)* **do**

2 Draw a bootstrap sample of size *sampsize* from the training data;

3 Grow a full tree $T_b(\mathbf{x}; \Theta_b)$ on the bootstrap sample using recursive binary splitting and selecting *mtry* variables at random from the p explanatory variables with stopping criterion *nodesize*;

4 **end**

5 Return the ensemble of trees $\{T(\mathbf{x}; \Theta_b), b = 1, 2, \ldots, B\}$;

6 Average $f_B(\mathbf{x}|\Theta) = \frac{1}{B}\sum_{b=1}^{B} f(\mathbf{x}|\Theta_b)$;

2.3 Gradient boosted regression trees

Developed by [9], gradient boosting algorithm involves building several regression trees sequentially. For each new iteration in the sequential process, a regression tree is grown using information from previously grown regression trees. In other words, each subsequent regression tree focuses on learning from the residuals obtained from previous trees. The result is a set of S regression trees $T_s(\mathbf{x}; \Theta_s)$, for $s = 1, \ldots, S$. The gradient boosted regression tree model is expressed as the sum of such trees:

$$F_S(\mathbf{x}|\Theta) = \sum_{s=1}^{S} f_s(\mathbf{x}|\Theta_s),$$

where $\Theta_s = \{R_{ms}, c_{ms}\}_{m=1}^{M_s}$ and $f_s(\mathbf{x}|\Theta_s)$ are the correponding models produced by the trees $T_s(\mathbf{x}; \Theta_s)$. Here, S is also referred to as the number iterations in the process. For each step s, we find the optimal Θ_s by solving the problem:

$$\widehat{\Theta}_s = \arg\min_{\Theta_s} \sum_{i=1}^{N} L(y_i, F_{s-1}(\mathbf{x}_i|\Theta) + f_s(\mathbf{x}_i|\Theta_s)). \tag{4}$$

We note that

$$F_s(\mathbf{x}_i|\widehat{\Theta}) = F_{s-1}(\mathbf{x}_i|\widehat{\Theta}) + \sum_{m=1}^{M_s} \widehat{c}_{ms}\mathbf{1}_{R_{ms}}(\mathbf{x}_i),$$

where $\widehat{c}_{ms} = \arg\min_{c} \sum_{\mathbf{x}_i \in R_{ms}} L(y_i, F_{s-1}(\mathbf{x}_i|\Theta) + c)$.

Under the sum of squared errors (SSE) loss function, it simplifies to the regression tree that best predicts the current residuals $y_i - F_{s-1}(\mathbf{x}_i|\widehat{\Theta})$, and $\widehat{c}_{ms}$ is the mean of these residuals in each corresponding region.

For other differentiable loss functions, the solution to equation (4) can be obtained by numerical optimization via gradient boosting as described in [9]. The regression trees $T_s(\mathbf{x}; \Theta_s)$ produced at each step are analogous to the components of the negative gradient:

$$g_{is} = -\nabla_{f_{s-1}} L(y_i, F_{s-1}(\mathbf{x}_i|\Theta)) = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i|\Theta))}{\partial F(\mathbf{x}_i|\Theta)}\right]_{F(\mathbf{x}_i|\Theta)=F_{s-1}(\mathbf{x}_i|\Theta)} \quad \text{for } i = 1, \ldots, N.$$

Therefore, solving equation (4) is equivalent to solving the following:

$$F_s(\mathbf{x}_i|\Theta) = F_{s-1}(\mathbf{x}_i|\Theta) + \gamma_s \sum_{i=1}^{N} g_{is}$$

where

$$\gamma_s = \arg\min_{\gamma} \sum_{i=1}^{N} L\left(y_i, F_{s-1}(\mathbf{x}_i|\Theta) + \gamma g_{is}\right)$$

Gradient boosting regression models can be implemented using the R package `gbm`. See [29] and [30]. The process is summarized in in Algorithm 3.

Algorithm 3: R-package: gbm

Input: Training dataset $\mathbf{x}$, y, B
Output: $F_S(\mathbf{x}|\Theta) = \sum_{s=1}^{S} \lambda f_s(\mathbf{x}|\Theta_s)$
for $s = 1, \ldots, S$ *(ntree)* **do**
  for $i = 1, \ldots, N * p$ *(p is bag.fraction)* **do**
    Compute $g_{is} = -\nabla_{F_{s-1}} L(y_i, F_{s-1}(\mathbf{x}_i|\Theta))$;
    Fit a regression tree $T_s(\mathbf{x}; \Theta_s)$ to the targets g_{is} giving terminal regions $R_1, R_2 \ldots, R_{M_s}$: $interaction.depth = M_s$ and stopping criteria is $n.minobsinnode$;
    for $m = 1, \ldots, M_s$ **do**
      compute $\widehat{c}_{ms} = \arg\min_{c} \sum_{\mathbf{x}_i \in R_{ms}} L(y_i, F_{s-1}(\mathbf{x}_i|\Theta) + c)$;
    end
  end
  Update $F_s(\mathbf{x}_i|\Theta) = F_{s-1}(\mathbf{x}_i|\Theta) + \lambda \sum_{m=1}^{M_s} c_{ms} \mathbf{1}_{R_{ms}}(\mathbf{x})$ where $shrinkage$ λ is used to reduce the impact of each additional fitted base-learner, regression tree, $T_s(\mathbf{x}; \Theta_s)$.
end
Return $F_S(\mathbf{x}|\Theta) = \sum_{s=1}^{S} \lambda f_s(\mathbf{x}|\Theta_s)$;

3 Extensions to multivariate decision trees

Decision trees discussed in the previous section are based on a single-valued response variable. In this section, we extend the concept of decision trees in the case where we have a multiple-valued, or multivariate, response variable. This extension has a large potential for actuarial and insurance applications where we commonly encounter multivariate responses. See, for example, [8] and [7]. To fix ideas, we assume to have a dataset of N observations with p explanatory variables and q response variables. That is, for $i = 1, \ldots, N$, we have $(\mathbf{x}_i, \mathbf{y}_i)$ where $\mathbf{x}_i = (x_{i1}, x_{i2} \ldots, x_{ip})$ and $\mathbf{y}_i = (y_{i1}, y_{i2} \ldots, y_{iq})$. Multivariate decision trees are naturally extended from the univariate trees by expressing the loss function that is based on the multivariate nature of the responses in order to capture the possible association of the multiple responses. Here we discuss three possible extensions: multivariate regression trees (MRT), multivariate random forests, and multivariate tree boosting. The steps in Algorithms 1-3 are very similar to that of multivariate decision trees; the only difference lies in the splitting criteria which is based on multivariate loss function.

3.1 Multivariate regression trees

The idea of multivariate regression trees originated in ecology by [3] where the author analyzed relationships between multiple species and the environment. Similar to the univariate case, we produce regression tree by partitioning the space of the explanatory variables into M disjoint regions $R_1, R_2, \ldots, R_M$ and then assigning constant c_{m_k} for each region R_m, $m = 1, 2, \ldots, M$ in the k-th response for $k = 1, 2, \ldots, q$. Define the vector

of k constants as $\mathbf{c}_m = (c_{m_1}, c_{m_2}, \ldots, c_{m_q})$. Given the regression tree, the multivariate response variable for each observation is then modeled as

$$f(\mathbf{x}_i|\Theta) = \sum_{m=1}^{M} \mathbf{c}_m \cdot \mathbf{1}_{R_m}(\mathbf{x}_i), \tag{5}$$

where $\Theta = \{R_m, \mathbf{c}_m\}_{m=1}^{M}$ denotes the partition with the assigned constants. In the multivariate regression trees (MRT) developed by [3], one method to find the constants $\mathbf{c}_m$ is to minimize a multivariate sum of squared error loss function as:

$$L(\mathbf{y}_i, \widehat{\mathbf{y}}_i) = \sum_{i=1}^{N} (\mathbf{y}_i - \widehat{\mathbf{y}}_i)^T (\mathbf{y}_i - \widehat{\mathbf{y}}_i), \tag{6}$$

where $\widehat{\mathbf{y}}_i$ is the predicted value of the multivariate response based on equation (5). For the conventional sums of squared deviations, each terminal nodes can be represented by the multivariate mean of response variables, the number of observations at the terminal node, and the explanatory variable that defines the segmentation. This is similar to the univariate case where in the multivariate extension, we can demonstrate that $\hat{c}_{m_k}$ is the average of the k-th response y_{ik} in the region R_m.

Although [3] primarily focused on the sums of squared deviations, he discussed extensions of the concept of multivariate regression trees by using two other multivariate loss functions. One of these loss functions is based on the multivariate sums of absolute deviations about the median as defined below:

$$L(\mathbf{y}_i, \widehat{\mathbf{y}}_i) = \sum_{i=1}^{N} \sum_{k=1}^{q} |y_{ik} - \tilde{y}_k| \tag{7}$$

where $\tilde{y}_k$ is the median for the response variable k. This measure is generally more robust to outliers when compared to the conventional sums of squared deviations. The other extension is using a distance-based measure by considering the sums of squared dissimilarities within decision nodes. Therefore in this case, because we want regions to be as dissimilar as possible, the splitting criterion is to maximize the reduction of within-node sums of squared distances at each split.

As with univariate regression trees, this extension to multivariate regression trees also inherits some of its advantages such as improved predictions, especially when the data includes missing values, lack of balance, nonlinearity, and high order interactions. Furthermore, multivariate regression trees provide benefits of grouping effects on the response variables. In other words, we can consider this procedure to be characterized as a constrained clustering. There are a few ways that we can interpret this. First, we can visualize the contributions of the relative importance of each explanatory variable in the split of the tree model. Second, we can display the multivariate group means in the terminal nodes, as well as clustering information, through the use of tree biplots constructed with dimension reduction by principal components. This is further discussed in the section on data estimation. We use the R package `mvpart` for calibrating the multivariate regression trees to data.

In his survey paper, [21] provided a short discussion about similar work on regression trees for longitudinal and multiresponse variables. It is worth mentioning that the unbiased recursive partitioning framework for building trees proposed in [16] can be extended to multivariate regression trees.

3.2 Multivariate random forests

Recall that in the univariate case, we describe the concept of random forest where we generate an ensemble of regression trees using bootstrapping resampling. This technique has the advantage of producing better predictions by avoiding overfitting, an aspect of random forest regression that is often underappreciated. To similarly improve predictions with multiple responses, [31] extended the idea of multivariate random forests for applying regression trees when there may be anticipated dependencies among the several responses. First,

in this technique, recursive binary splitting for partitioning the space of explanatory variables by minimizing a 'covariance' weighted loss function as defined by

$$L(\mathbf{y}_i, \hat{\mathbf{y}}_i) = \sum_{i=1}^{N} (\mathbf{y}_i - \hat{\mathbf{y}}_i)^T \mathbf{V}^{-1} (\mathbf{y}_i - \hat{\mathbf{y}}_i), \tag{8}$$

where $\mathbf{V} = \text{Cov}(\mathbf{y})$ represents the covariance matrix of the multivariate response variable. The dependence structure is characterized by this covariance that can be described to capture different patterns (e.g., compound symmetry, unstructured, autoregressive, spatial power). Similar to univariate random forests, we generate an ensemble of such multivariate decision trees using bootstrap resampling.

In [40], multivariate random forests was applied in the area of genetics to examine and analyze "transcriptional regulation networks" that are said to be important for understanding the physiological processes in yeast under various stress conditions. In ecology, [31] applied multivariate random forests to identify various environmental characteristics and understand their effects on the habitat of several spider species. In particular, the results revealed a "dynamic relationship between environment and species". To be able to compare the results between multivariate regression trees and multivariate random forests, [31] used the same dataset as that used by [3]. The multivariate random forests did not produce reasonable results for our dataset and therefore, we did not present the results from this technique. However, this technique has advantages that may work for other datasets and therefore we provide this discussion.

3.3 Multivariate tree boosting

In order to find and interpret structure in datasets with multiple response variables and several explanatory variables (even possibly exceeding the sample size), [24] introduced the idea of extending the gradient boosted regression models to the multivariate case. This gives rise to the term multivariate tree boosting. In their abstract, [24] said that this extension "is a method for nonparametric regression that is useful for identifying important predictors, detecting predictors with nonlinear effects and interactions without specification of such effects" where the multivariate response variable represents several outcomes that are correlated.

The concept of multivariate tree boosting has similarities to the multivariate regression trees discussed in section 3.1. Regression trees are also constructed based on the multivariate response variable where each tree is modeled in terms of the optimal partitions of the space of explanatory variables into regions. The constants assigned in the regions are identified according to the solution of miminimizing the multivariate sum of squared error (SSE) loss function as defined in equation (7). However, unlike the multivariate regression trees, the multivariate tree boosting constructs several trees, as in the univariate tree boosting, done sequentially where at each iteration, a regression tree is grown through learning and using information from previously grown trees. Learning is accomplished by an examination of the residuals; those with smaller residuals (observations with good predictions) are assigned smaller weights than those with higher residuals (observations with bad predictions) in the next iteration. During the learning process, splits are limited for each iteration without having to produce a fully grown tree which may result in overfitting. In essence, the model performance is "boosted" by correcting bad predictions. Multivariate tree boosting can be implemented using the R package `mvtboost` that is based on [24].

In [24], multivariate tree boosting was applied in the area of psychology to understand the impact of demographic characteristics as well as physical and health states on the psychological well-being of aging adults. In the paper, psychological well-being is the multivariate response variable with six scales: "autonomy, environmental mastery, personal growth, positive relationships with others, purpose in life, and self-acceptance." Interestingly, [28] applied the concept of gradient multivariate tree boosting for longitudinal data that is based on understanding the health status of patients over time after lung transplantation.

To describe the algorithms for the multivariate extension, we use the multivariate response with the corresponding multivariate loss function in Algortihms 1, 2, and 3. To ease use of the `R` packages, we also summarize the hyperparameters that required tuning for the various decision tree methods in Table 6 in the appendix.

4 The LGPIF Data

For the empirical section of this paper, we use the dataset with information about the insurance coverage for buildings, vehicles, and equipments of local government units in Wisconsin. These units include, for example, cities, towns, villages, counties, school districts, fire departments, and other miscellaneous entities in the state. Funds to cover property and casualty insurance coverages for these local government units are established by the Wisconsin Local Government Property Insurance Fund (LGPIF). This dataset was drawn from a project of the actuarial research group at the University of Wisconsin and additional details about this project can be found at the website. This dataset has been extensively studied and analyzed in [6], [7], and [32]. We do not replicate the GLM approach used on this same dataset in earlier works. Instead, to analyze the multivariate structure of the data, these previous works used mainly parametric models; this paper emphasizes the benefits of using tree-based models without having to specify probability distributions. For our purpose, we used observations for years 2006-2010 as training set and year 2011 as validation set.

In the LGPIF data, there are six types of insurance coverage: Building and Contents (BC), Contractor's Equipment (IM), Comprehensive New (PN), Comprehensive Old (PO), Collision New (CN), and Collision Old (CO). BC provides insurance for buildings and their contents, IM provides insurance for equipments mainly belonging to contractors, and the rests provide comprehensive and collision coverages for moving vehicles. For more description of these types of coverages, please refer to Table 3 of [6]. For our purpose of fitting tree-based models, we examine a multivariate response variable with six different components, with each component associated to each of the six types of insurance coverages. We describe these six variables in Table 1. We note that these variables are transformed with a logarithmic scale where we added one to each average claim size, per year, to accommodate the zero claims. Zero claims indicate either there was no claims made or simply no coverage provided for the year. From hereon, we will simply describe these as the logarithm of the claim size.

Table 1: Description of the six components of the multivariate response variable.

Variable code	Description
yAvgBC	Log of the average building and contents claim size
yAvgIM	Log of the average contractor's equipment claim size
yAvgPN	Log of the average comprehensive new vehicles claim size
yAvgPO	Log of the average comprehensive old vehicles claim size
yAvgCN	Log of the average new vehicle collision claim size
yAvgCO	Log of the average old vehicle collision claim size

Table 2 provides summary statistics for the logarithm of the claim sizes for the training dataset, which consists of observations for the period years 2006-2010. The proportion of positive claims for BC is about 30%, the highest among all types of coverage; for all other types of coverage, the proportion of positive claims ranges between 4% and 6%. The means of the logarithm of the claim sizes for all types are in the range of 7.5 to 9.0, or in terms of the original dollar scale, this range is between 1800 and 6500, with BC giving the largest mean claim size. The table also indicates that the largest claim size comes from the BC type of coverage.

Figure 3 provides density plots of the logarithm of the claim sizes where we excluded the zeroes and, for ease of comparison, we used the same scale for all types of coverage. The top portion gives the frequency by count while the bottom portion gives the frequency by proportion. We can deduce some interesting observations from these plots. First, as also indicated in Table 2, the BC coverage clearly shows most frequent positive claims and the top portion of the figure shows that this is also true for all ranges of claim sizes. Second, as shown in both the top and bottom portion, the BC coverage has the largest variability among all types and also the most positively skewed distribution. Finally, we also observe a variety of distribution shapes for

Table 2: Summary statistics of the six components of the multivariate response variable, 2006-2010 (training dataset).

	yAvgBC	yAvgIM	yAvgPN	yAvgPO	yAvgCN	yAvgCO
Percent of zeroes	70.34	95.84	94.42	95.35	93.45	93.55
Minimum	0.69	0.69	3.58	3.71	5.24	0.69
1st Quantile	7.79	7.24	7.19	7.08	7.35	7.56
Mean	8.75	8.42	7.64	7.71	8.12	8.22
Median	8.56	8.41	7.76	7.71	7.96	8.08
3st Quantile	9.55	9.46	8.18	8.37	8.85	8.85
Maximum	16.37	13.09	10.71	12.04	10.68	12.41

the different types of coverage. If we consider the large point mass at zero, this presents a challenging task of specifying marginal parametric distributions for these claims. This is one reason we use distribution-free tree-based models to fit these claim sizes.

We also provide Figure 4 which also shows how frequent claims are for the BC coverage because of the dominance of the shade for this coverage. This figure is a stacked density graph which includes the zero claim sizes. However, at zero claim size, BC does not dominate; as shown in Table 2, BC has the smallest proportion of zeroes.

Our primary interest is to build tree-based models that account for the dependence among the components of a multivariate response variable. Hence, we also need to examine evidence of the presence of correlation between the components. In this case, we present Figure 5 to show the strength of dependence between the components, with a darker shade and larger circle indicating stronger correlation. To illustrate, we find that coverage PN and PO has the largest correlation of 0.51. Interestingly we also note that all pairs of coverage have positive correlations.

In the Appendix, we define the explanatory variables we consider in constructing the regression trees. In general, the continuous explanatory variables are logarithms of the coverage amount for each type and notice that for coverage type BC and IM, we have deductible amounts. No deductible amounts are available for the other types. Similarly, for the categorical explanatory variables, we have indicators for entity type (e.g., city, town, school district) as well as indicators for "no claim credit" for each type of coverage. The latter indicators provide information about the presence or absence of claims in the prior year. See Appendix for more details including statistics.

As done in the two-part model of [7], we used log-transformed coverage amounts as explanatory variables. In other approaches especially for the severity component, coverage amounts may be alternatively used as exposures in the model. However, treating them as explanatory variables is the correct approach for us since we do not separate the frequency and severity components. The zero part of the claim will lose coverage information when treated as exposure.

5 Model calibration

In this section, we calibrate and build regression tree-based models to the LGPIF data described in Section 4. For the purpose of this section, we use the data for years 2006 to 2010 as our training sample. A cruical part of successfully training a tree-based model is to control model complexity in order to maintain a good balance between bias and variance that leads to high prediction accuracy. More specifically, a simple tree-based model may cause underfitting (low variance) with high bias; on the other hand, a complex tree-based model may cause overfitting (low bias) with high variance. See [15]. To address these issues, for training tree-based models, we use k-fold cross-validation on the training sample in tuning the parameters. In applying cross-validation to find the optimal set of tuning hyperparameters, the grid search is the most commonly

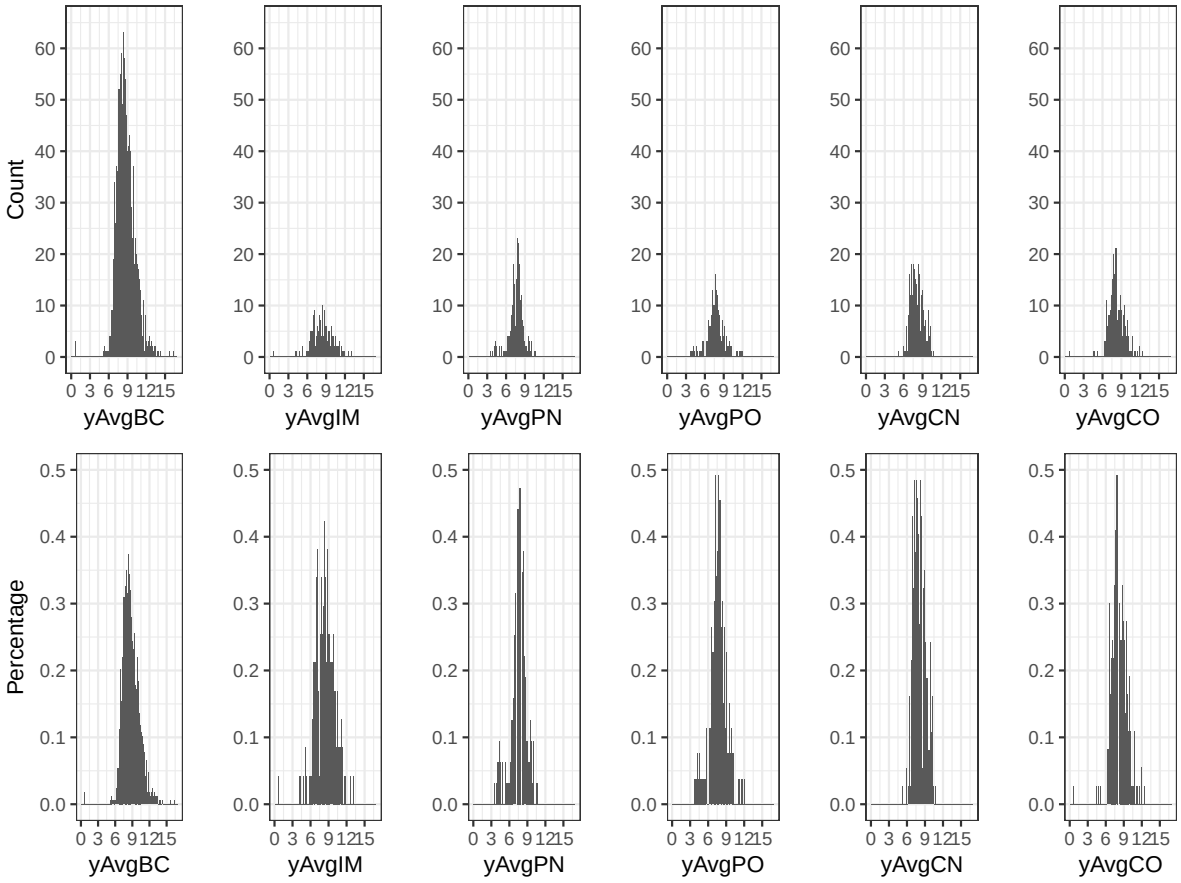

Figure 3: Density plots of the logarithm of the positive claim size by type of coverage, 2006-2010 (training dataset)

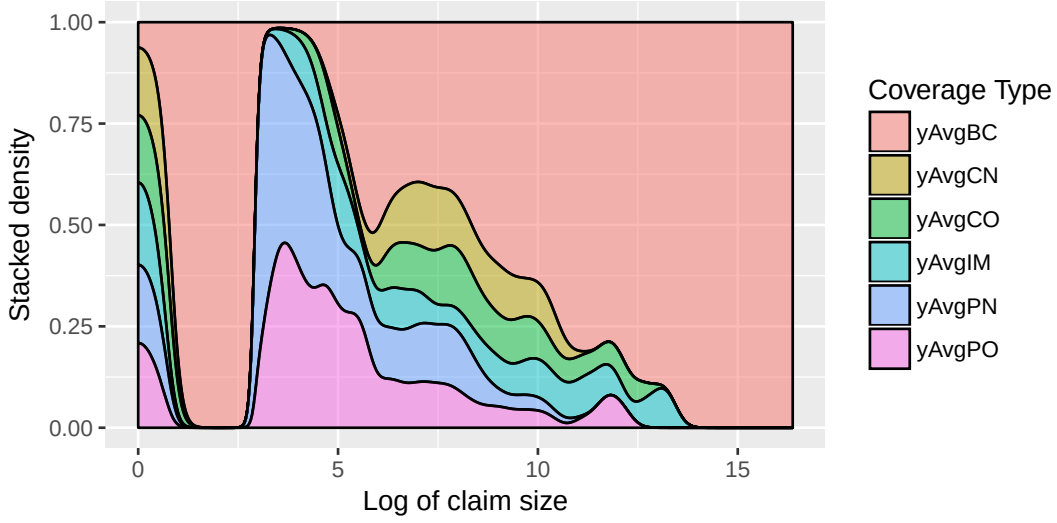

Figure 4: Stacked density plots of the logarithm of the claim sizes, including the zeroes

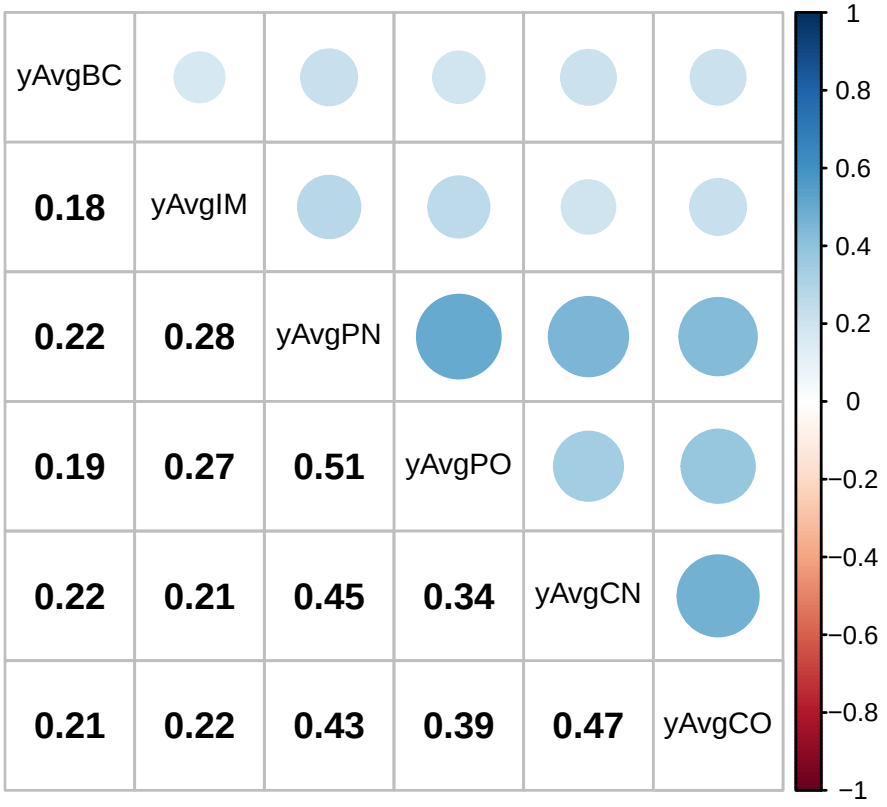

Figure 5: Correlations between components of the multivariate response variable

used algorithm that employs an exhaustive search process on specified subset of hyperparameter space. We choose the final model with the lowest cross-validation prediction error.

R codes can be readily provided upon request and at a future date, these codes will be posted on our research grant website.

5.1 Fitting univariate decision tree models

5.1.1 The univariate regression trees based on CART

In fitting the CART procedure to our training dataset, without loss of generality, we only consider the component BC. We first grow a full tree using recursive binary splitting starting with a minimum number of five observations in a region for a split to be attempted. The stopping criterion of a minimum number of five observations is the result of tuning this parameter; the default for the `rpart` package is 20. We next prune this full grown tree using cost-complexity pruning with 10-fold cross validation to determine the optimal number of splits, which in our case is 8. The result of the pruning process can be visualized in Figure 6 that shows the cost-complexity parameter, or equivalently the number of splits, in relation to the cross-validation relative error. This cross validation relative error is simply the percentage change in the mean squared prediction error. By choosing the cost-complexity parameter that corresponds to the smallest cross validation relative error, the figure shows that the optimal number of splits is 14. However, it is recommended to adjust the choice of this optimal number of splits by considering one standard error above this minimum relative error as indicated with a horizontal line. This adjustment is necessary to avoid overfitting. Hence, this gives parsimonious model to choose 8 as the optimal number of splits. See [2]. The red dot corresponds to the smallest cross validation relative error while the orange dot corresponds to the one standard error above this cross validation relative error.

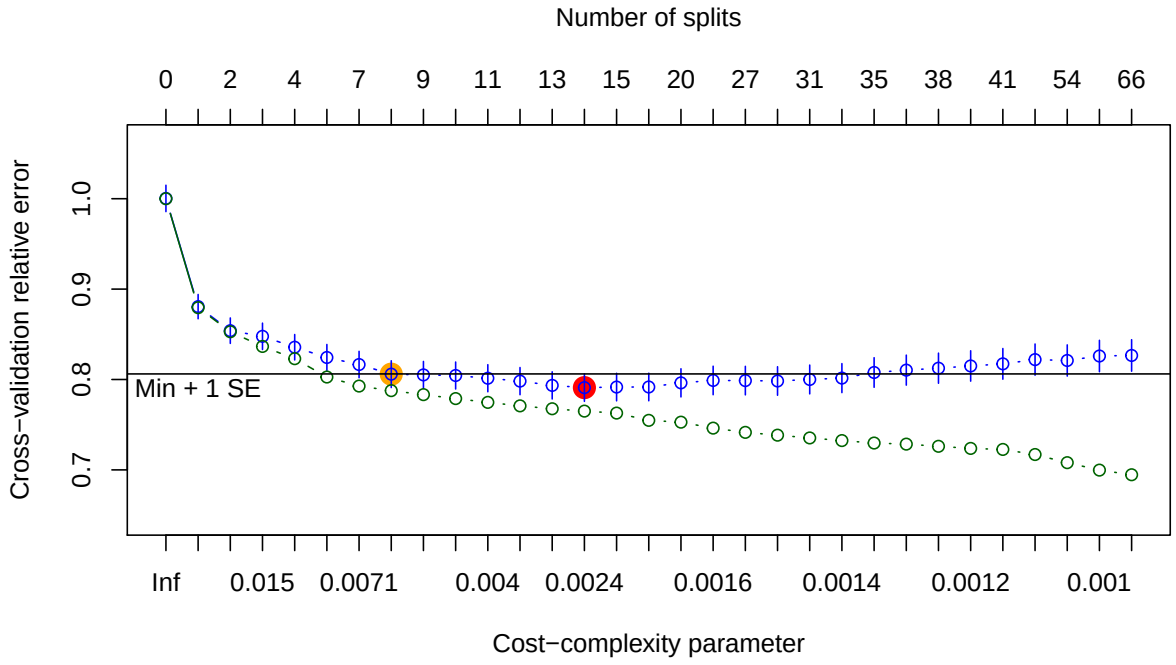

Figure 6: Choosing the optimal number of splits

In Figure 6, the green dots refer to the training relative error with each successive split reducing the mean squared prediction error.

After accounting for the optimal cost-complexity, we produce the optimal univariate regression tree in Figure 7 using [23]. The nodes at the bottom provide the information on prediction values. The figure shows the set of explanatory variables and split points used to form the nodes. The explanatory variable with the most significant effect is selected for the first split and each following split is conditioned on all previously selected explanatory variables. For example, the first split uses CoverageBC which has the most significant effect and divides into two child nodes. Hence, the decision tree model can capture interactions between explanatory variables.

5.1.2 Random forest regression

The recursive partitioning algorithm used in constructing decision trees often lead to the problems of overfitting. It is well established that such overfitting can lead to unreliable prediction of future or new observations. In other words, decision tree models can drastically change from sample to sample, which leads to low prediction accuracy. A remedy for overfitting is the use of random forests. Because of the multiplicy of regression trees necessary in random forests, we focus our discussion on two important aspect of fitting random forest regression.

An important phase is determining the number of trees to generate that will eventually be used for prediction. Figure 8 shows the relationship between the number of trees and the out-of-bag (OOB) error rate. The performance of the estimation is measured based on the out-of-bag samples that were not used during the learning stage. The prediction error computed from this estimation is called the OOB error. As to be expected, the fewer the trees, the larger this OOB error. However, due to large sample theory, the OOB error rate will eventually level off and deciding on the number of trees for building the random forests is based on this leveling off, as demonstrated in Figure 8. In deciding on the number of trees, we need to address the balance of using many trees in order to get more stable prediction and using fewer trees in order to achieve efficiency. Based on the figure, we decid that the optimal number of trees to use is 200.

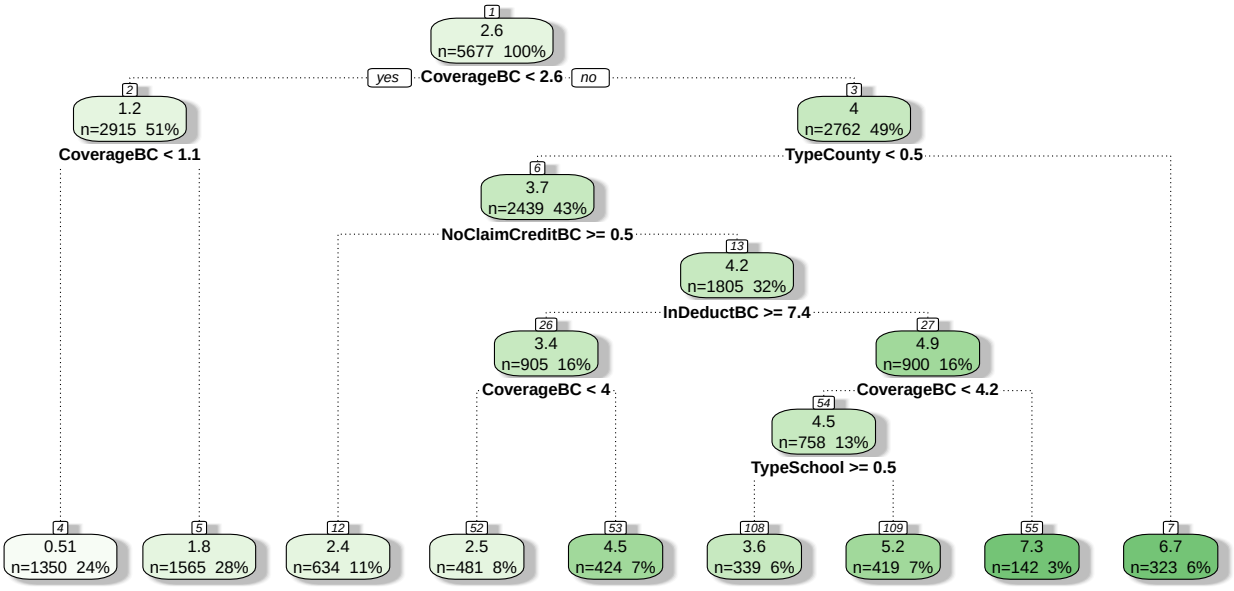

Figure 7: The optimal univariate regression tree

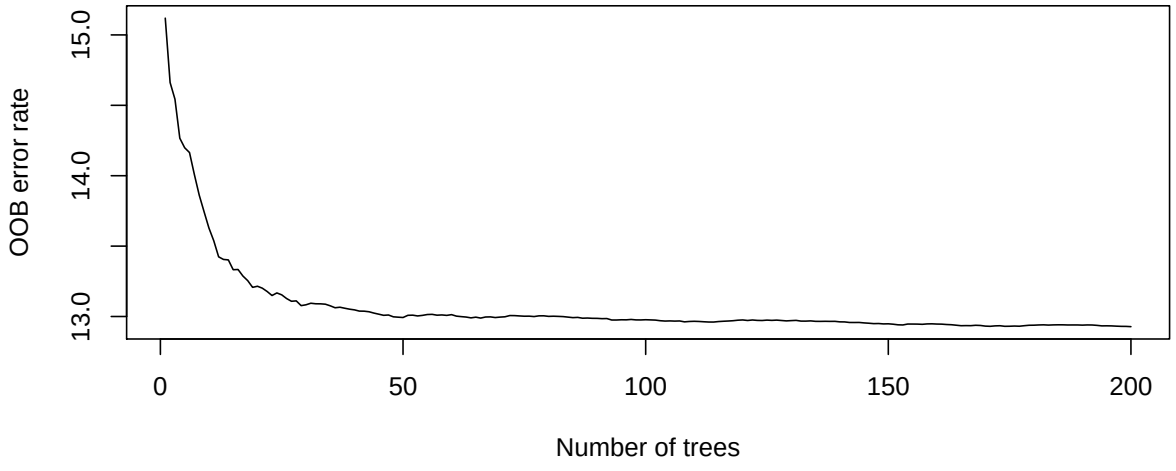

Figure 8: Random forests: the number of trees vs the out-of-bag (OOB) error rate

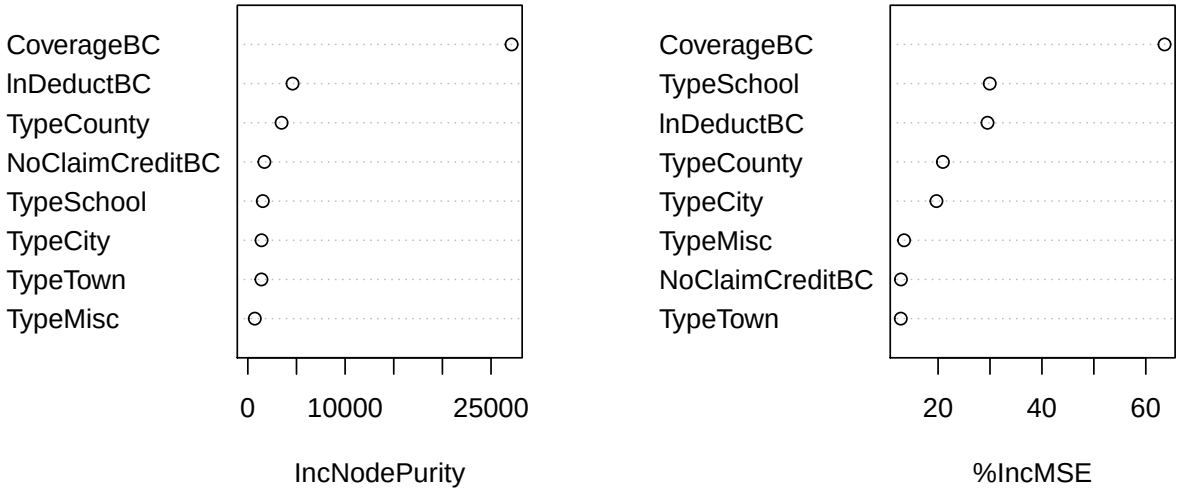

Figure 9: Variable importance in random forest regression

Another important phase is the measurement of the variable importance produced by the random forests using the selected optimal number of trees. In this case, there are two popular variable important measures used. The first one incorporates a weighted mean of the improvement of the individual trees based on the splitting criterion produced by each explanatory variable; [9]. The "IncNodePurity" measures the degree of impurities of a split at each node of a tree based on the loss function. This is accomplished by adding up all the decreases in the loss function for each explanatory variable over all the trees in the random forests. A higher value of the "IncNodePurity" represents a higher variable importance. Figure 9 shows CoverageBC has highest IncNodePurity with 28414 and followed by lnDeductBC with 4623.

The second commonly used measurement is the permutation accuracy importance. "%IncMSE" calculates the deterioration of the prediction accuracy of the random forests when permuting the values of each explanatory variable of the test set in order to break the association with the response variable. For random forest regression, it is the average increase in squared residuals of the test set when the explanatory variable is permuted. A higher %IncMSE value represents a higher variable importance. Figure 9 shows CoverageBC has highest %IncMSE with 7.95%, followed by TypeSchool with 1.26% and lnDeductBC with 1.21%.

It is typically understood that %IncMSE provides the more accurate measure for variable selection because the IncNodePurity may have a bias in the variable selection inherited by the regression tree. To illustrate such a bias, potential explanatory variables may vary in their scale of measurement or their number of categories. This explains the variable importance may vary significantly between the two measures for some datasets, however, our dataset does not present this issue. Random forests variable importance measures may still be a sensible means for variable selection in many applications, but may not be too reliable in other situations. See [33].

5.1.3 Gradient boosting

Both the random forests and the gradient boosting are major improvements to the CART algorithm in terms of prediction accuracy. While both create ensemble of trees from weak learners, gradient boosting is performed iteratively to develop a strong learner. For details of the procedure for gradient boosting, please see section 2.3. In fitting gradient boosted trees to our dataset, we mainly used the `gbm` R package. The procedure includs tuning for the number of trees to grow and the size of the weak learner tree. Based on our dataset, we use 10-fold cross validation to determine these tuning parameters and we find that the optimal number of trees

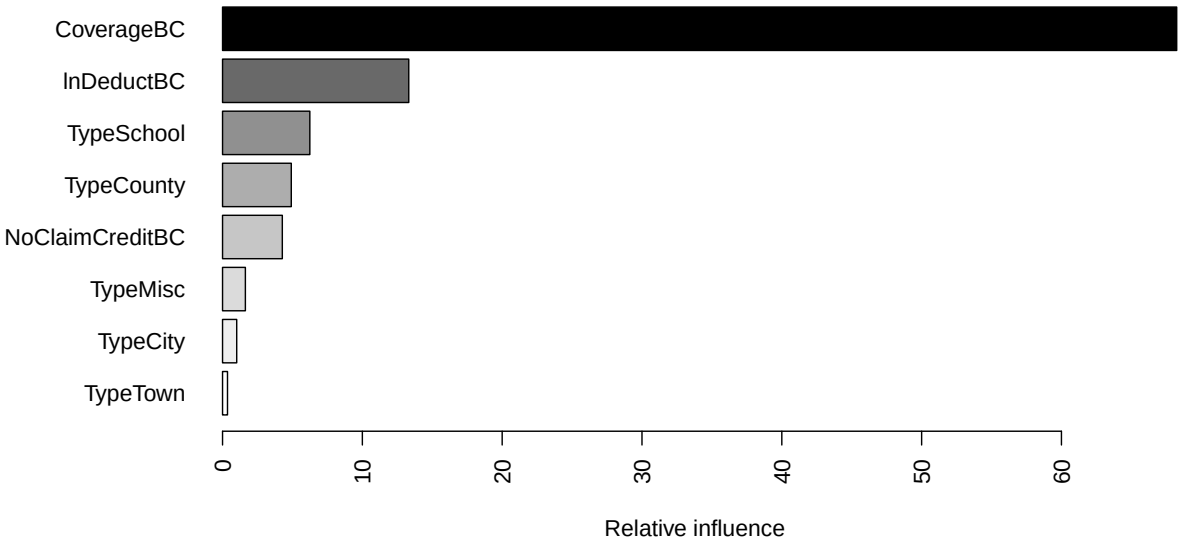

Figure 10: Relative influence of the explanatory variables in gradient boosting

is 1000, the interaction depth is 5, the minimum number of observations to terminate the splitting process is 5, and the shrinkage parameter is 0.005.

The result of creating gradient boosted regression trees is best summarized in terms of the relative influence of the explanatory variables. In this case, the relative influence is calculated by the number of times an explanatory variable is selected for splitting, weighted by the improvement in the sum of squared errors to the model as a result of each split, and then, averaged over all the trees. See [11]. The relative influence of each variable is further standardized so that the sum adds up to 100%, with the larger value indicating a stronger influence on the response variable. Figure 10 displays the relative influence of the explanatory variables with the highest impact in predicting yAvgBC. According to this figure, it is not surprising to find that CoverageBC is the most dominant explanatory variable with 60.28%; this is followed by lnDeductBC with 13.56%.

5.2 Fitting multivariate decision tree models

5.2.1 Multivariate regression trees

In this and the subsequent subsection, we discuss the results of fitting multivariate tree-based models. Similar to the univariate case, we select the complexity parameter to determine the optimal size of the tree and we can accomplish this by examining Figure 11, which displays the relationship between the complexity parameter and the cross-validation relative error of the prediction. It can be inferred from this figure to select 0.0064 as the cost-complexity parameter. See Figure 6 for additional details.

After the cross-validation, we produce the final multivariate regression trees using our training dataset and this is displayed in Figure 12. This is the smallest tree within one standard error of the minimum relative error. This tree has eight terminal nodes with an estimated predictive error that explains (1-0.739)*100% of total variance. During each recursive binary splitting, it also shows the cyclical shadings across the bar plots that indicate the six response variables differentiated from left to right. At each terminal node, the height of each bar gives the mean of the respective response variables and n indicates the number of observations within that node. Recall that the split is determined according to the multivariate squared error loss function that captures the dependence structure of the multivariate response. Colored circles help to identify each node.

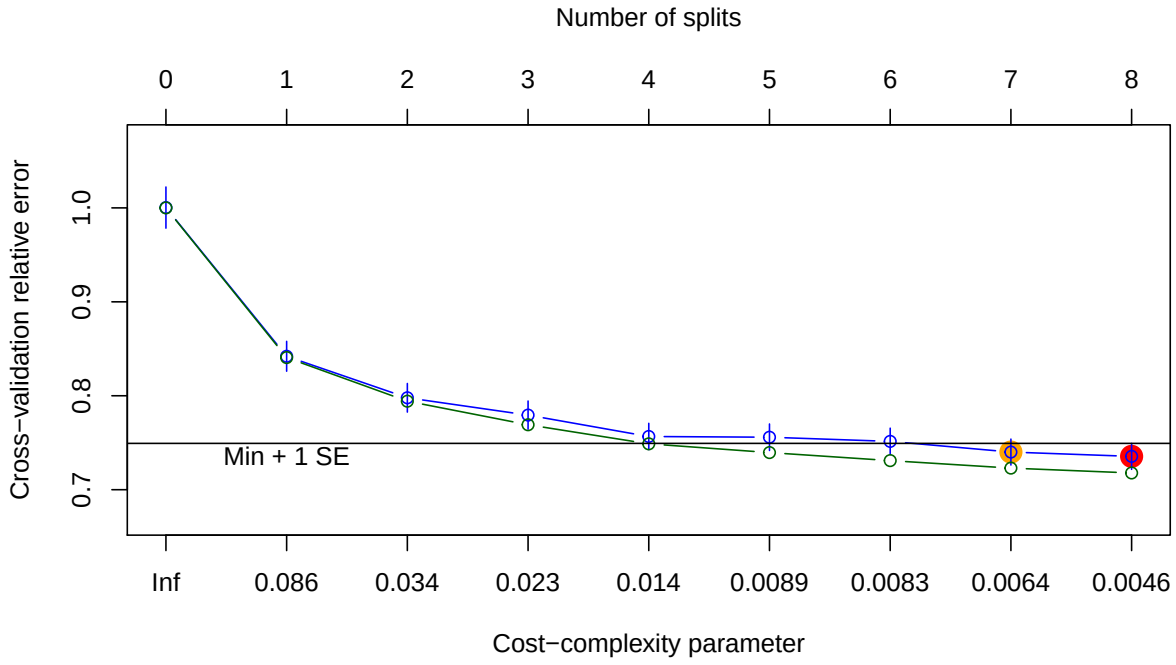

Figure 11: Choosing the optimal number of splits for the multivariate regression trees

The length of the branch is strictly proportional to the reduced amount of the variance (SSE) by the explanatory variable used in the split. The length of each branch is proportional to the percentage of explained variance and the splits are in descending order of importance. Note from Figure 11 that TypeCounty variable explains most of the variance; in particular, immediately following the first split, the final tree explains (1-0.841)% of the variance, while the rest of the splits together account for (0.841-0.739)% of the variance. This is in contrast from Figure 7 where CoverageBC was chosen as the first split.

It is difficult to compare the differences of the level of claims for each component of the response vector: BC, IM, PN, PO, CN, and CO. This is because the mean response is multivariate for each terminal node. However, multivariate regression trees can be viewed as constrained clustering. The terminal nodes are similar to clusters with respect to a measure of dissimilarity, e.g., squared error loss function, with each cluster defined by the region $\{R\}_{m=1}^{M}$. See [14]. Still this makes it difficult to visualize but we reduced the dimensionality using the first two principal components in a Principal Component Analysis (PCA); the details of this dimension reduction are not necessary for our purpose. We refer the reader to [18].

In Figure 13, we show the biplots representing the tree structure in Figure 12. See [12] and [35]. In the tree biplot, the large colored circles, consistent with the terminal nodes, represent the response vector means in the terminal nodes. The small colored points, again consistent with each node, represent PCA-projected individual observations. The label for each response variable in the figure is indeed the corresponding weighted mean calculated from the means of all the terminal nodes. From the figure, we deduce that the overall average claim for BC is very far apart from those of the lines of coverage. We can also see that the overall average claims for CN, CO, PN, and PO are relatively close; these lines of coverage all refer to moving vehicles. In addition, this helps explain the relatively weak dependence between BC and other lines; see also [7].

5.2.2 Multivariate tree boosting

In this subsection, we improve the multivariate regression trees with gradient boosting where we sequentially grow trees by learning the information from previously grown trees. The first important step is to determine the optimal size of the tree as well as other tuning parameters. Figure 14 displays the mean squared error in

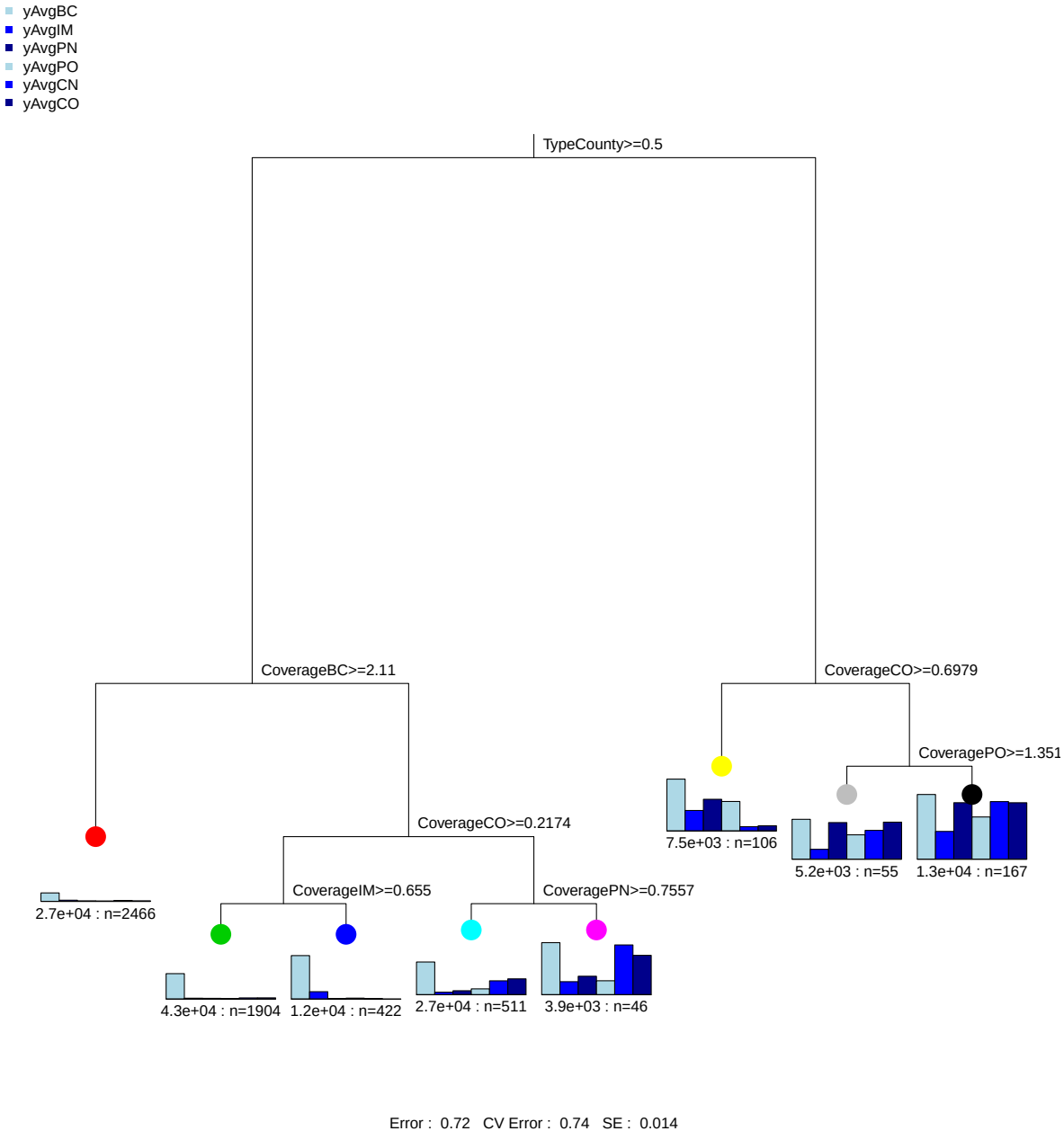

Figure 12: The optimal multivariate regression trees

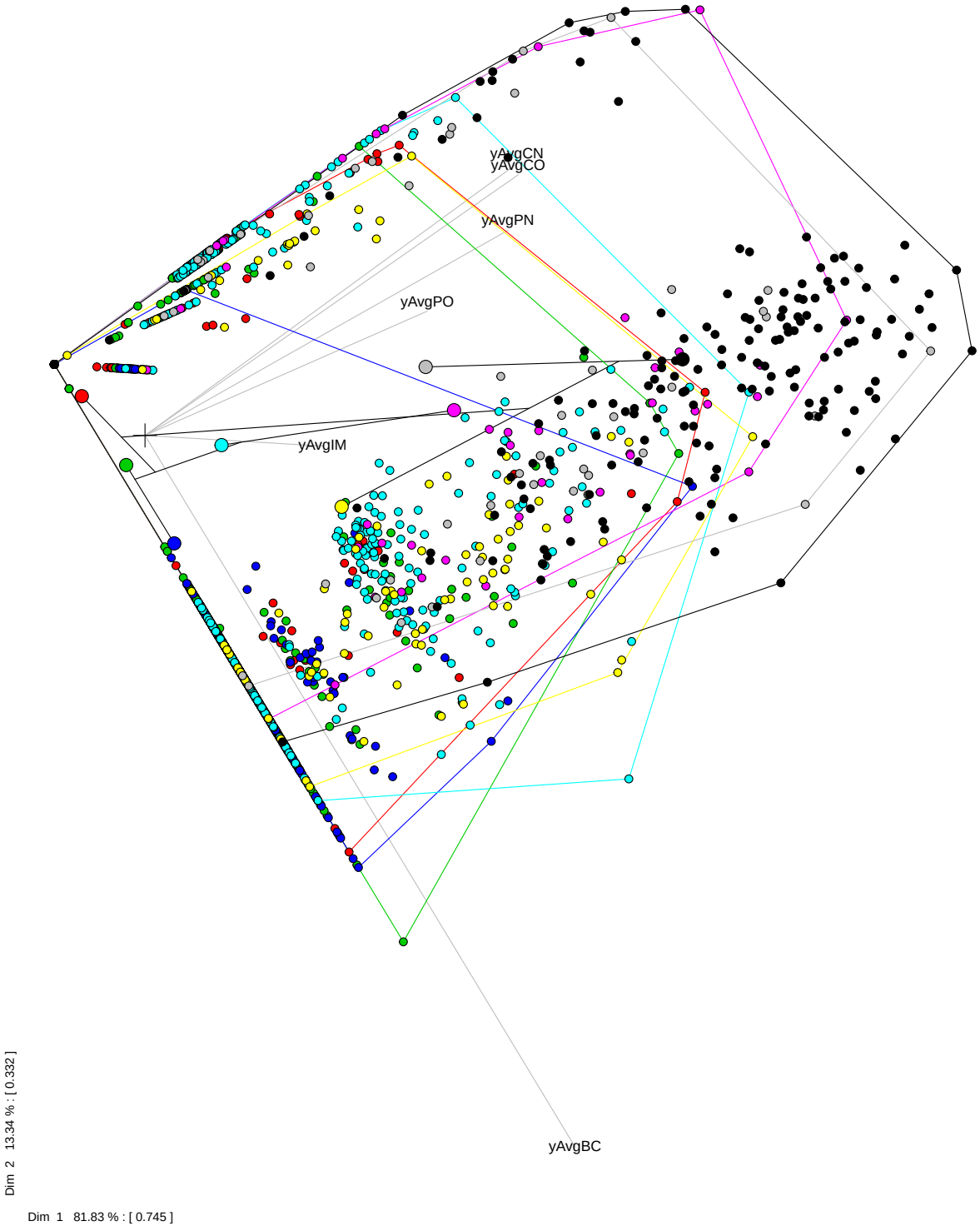

Figure 13: Multivariate regression tree biplots using PCA to reduce into two dimensions

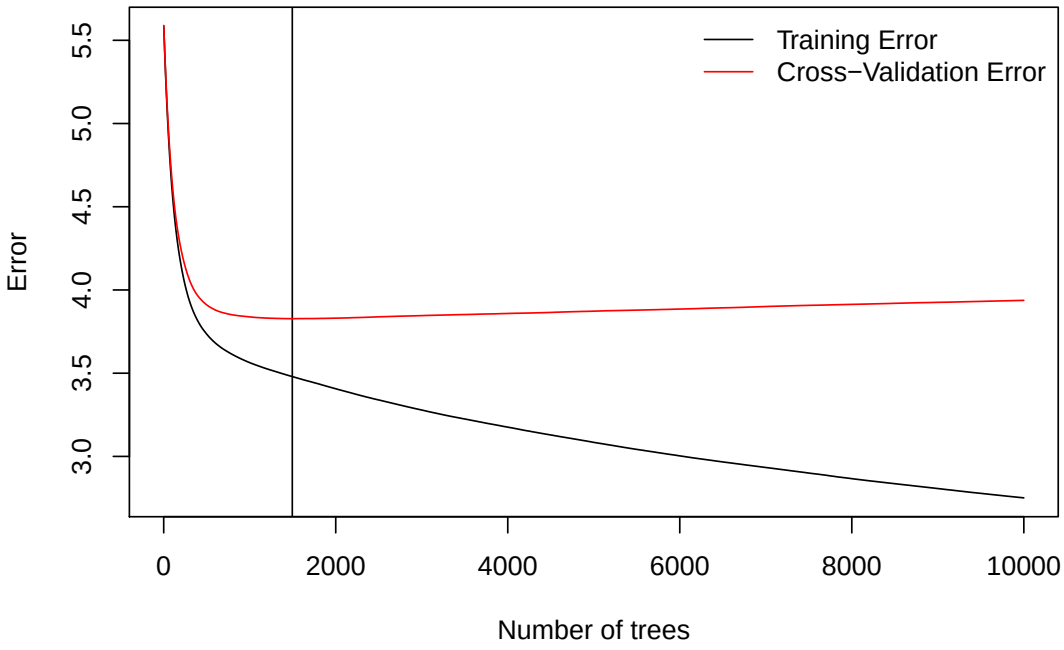

Figure 14: Determining the optimal size of the tree for the multivariate tree boosting

relation to the size of the tree. Here we show the mean squared error based on the training dataset as well as the result based on 10-fold cross-validation. From this figure, we conclude that the optimal size of the tree should be 1495. We note that this figure was based on a shrinkage of 0.005; this shrinkage parameter was determined on a trial-and-error basis through several iterations.

Similar to the univariate gradient tree boosting, it is important to determine the relative influence of each explanatory variable because this reveals their degrees of variable importance. Excluding some irrelevant explanatory variables from the models is favorable for prediction accuracy. To understand the variable importance of all the explanatory variables with respect to all response variables, we display this in the form of a heat map Figure 15 which lists the explanatory variables on the x-axis and the six response variables on the y-axis. To illustrate, consider the BC coverage where we find that these three explanatory variables have the following relative variable importance: CoverageBC with 59.2%, CoverageIM with 19.0%, and lnDeductBC with 10.7%. Furthermore, it is apparent from the figure that the variable TypeCounty is important across the different coverage. We recall that the explanatroy variable TypeCounty is used at the first split in the MRT, see the figure 12. For each coverage in the figure, we can examine the relative variable importance horizontally. In the figure, the numbers displayed, as well as the darkness of the shade, provide a sense of the degree of variable importance. The variable importance here account for the dependence structure between the variables.

Another way to explain the effect of each explanatory variable to the dependence structure is to measure the contribution of each variable to the changes in the covariance. [24] defined the covariance discrepancy, D, to measure the difference between sample covariance matrices, $\hat{\Sigma}$, of multiple response variables at each gradient descent step. At each gradient descent step b, the explanatory variable chosen by the algorithm, explains the covariance discrepancy, $D_{b,k}$, after the k-th response variable fitted where

$$D_{b,k} = \|\hat{\Sigma}_{(b-1)} - \hat{\Sigma}_{(b,k)}\|.$$

Summing overall the covariance discrepancies at each step measures the contribution to the covariance explained by each specific explanatory variable. Figure 16 shows a heatmap of the covariance discrepancies for any pair of response variables (y-axis) as explained by each explanatory variable (x-axis). We can deduce from the figure that TypeCounty, CoverageCN, and CoverageCO explain covariance discrepancies across a wide range of pairwise response variables. In essence, these explanatory variables are useful for detecting

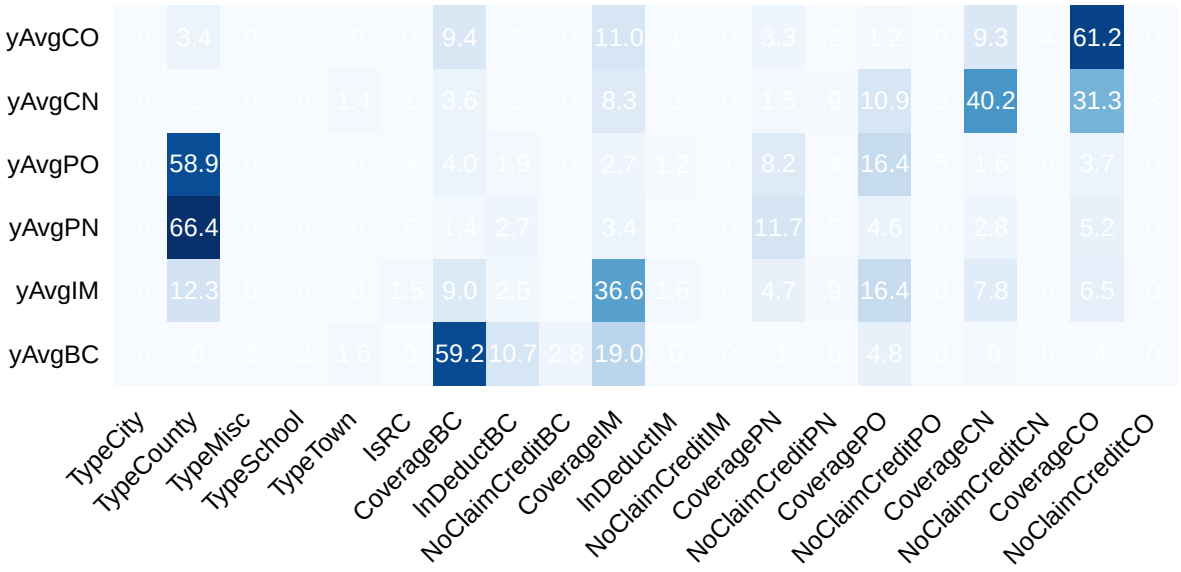

Figure 15: A heatmap of variable importance using multivariate tree boosting

the presence of dependency for the multivariate response. On the other hand, coverageBC explains most of the covariance discrepancies only for pairs that contain BC, and coverageIM explains most of the covariance discrepancies only for pairs that contain BC or IM.

Additional information that provides the effect of each explanatory variable for each response variable is provided in the appendix.

6 Model validation and comparison

This section provides details about the result of comparing the performance of our five different models. For this purpose, we use the calibrated models discussed in the previous sections to make predictions based on the validation (or test) dataset. In addition, we compare the various models only for BC; we extracted the prediction for yAvgBC from the marginals for multivariate tree-based models. This is done in order not to overwhelm the reader, and at the same time, BC coverage has the most number of observations. Similar comparisons can be made for other coverages. For summary statistics for this test dataset, we refer the reader to the appendix.

There are several prediction accuracy measures but there is no unique perfect measure that can be used to judge prediction accuracy under all circumstances. Each measure has its own focus, which also leads to its shortcomings. To make a fair comparison between different models, we utilize a few popular measures. These measures are: coefficient of determination R^2, Gini index, mean error (ME), mean percentage error (MPE), mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). These validation measures are each defined in the appendix.

The results of comparing the performance measures of the five tree-based models we calibrated are summarized in Table 3. The comparative values from these tables are straightforward to interpret, however, it is better to make model comparisons based on these values graphically. Figure 17 provides a heatmap comparing the performance of the various models according to the validation measures. For ease of comparison, this heatmap has been organized by rescaling all the measures so that for each measure, a value of 100 is the best and a value of 0 is the worst. For R^2 and the Gini index, we know that the higher the better; for these measures, we find the highest value in each column and scale it to 100. For all the other measures, since the smaller the better, we multiplicatively invert (take the reciprocal) the original value and then apply the rescale. The figure

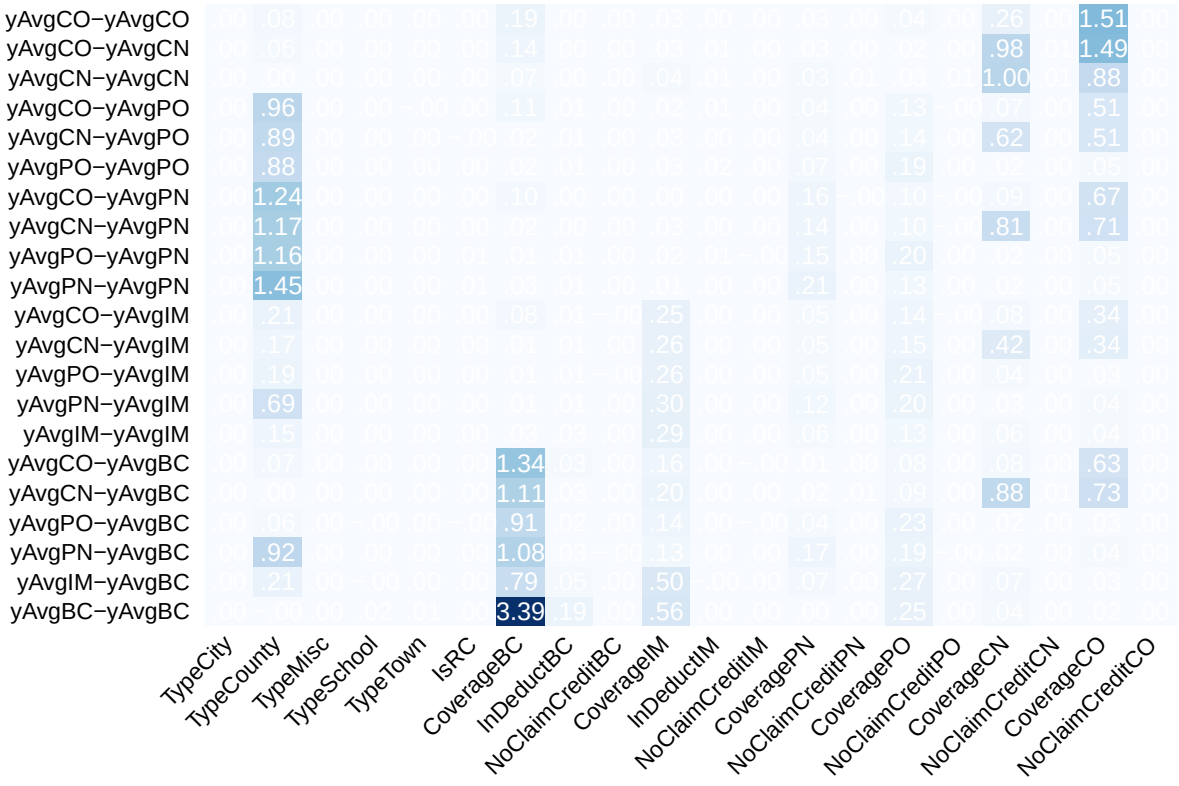

Figure 16: A heatmap of covariance discrepancies for pairs of response variables

is also color coded so that dark blue represents the best and the dark red represents the worst. Anything that performs in between is relatively measured according to their degree of closeness to each of these colors.

Overall, we can say that multivariate tree-based models generally outperform univariate tree-based models. From figure 17, although multivariate tree boosting is not always the best predictive model, it is clear that, overall, multivariate tree boosting generally outperforms all the other predictive models. Gradient boosting and random forests, on the other hand, provide values of validation measures that are fairly close to those from multivariate tree boosting. Multivariate tree boosting has the unique additional feature that captures the dependency structure of the response variables. This helps the prediction accuracy only slightly better because we do not have very strong presence of dependencies of BC with the other coverages. Not surprisingly, both the univariate and the multivariate regression trees underperformed in comparison to the ensemble models. This is because as already pointed out, ensemble models, such as gradient boosting and random forsts, make weak learners into strong learners.

Table 3: Comparison of model validation measures.

Model	R2	Gini	ME	MPE	MSE	MAE	MAPE
Regression tree (CART)	0.177	0.346	0.065	54.300	14.572	3.026	58.381
Random forests	0.220	0.406	0.089	50.308	13.805	2.806	54.916
Gradient boosting	0.226	0.410	0.033	51.001	13.701	2.893	55.585
Multivariate regression trees	0.204	0.376	0.047	51.907	14.097	2.974	58.229
Multivariate tree boosting	0.229	0.414	0.048	50.920	13.651	2.883	55.823

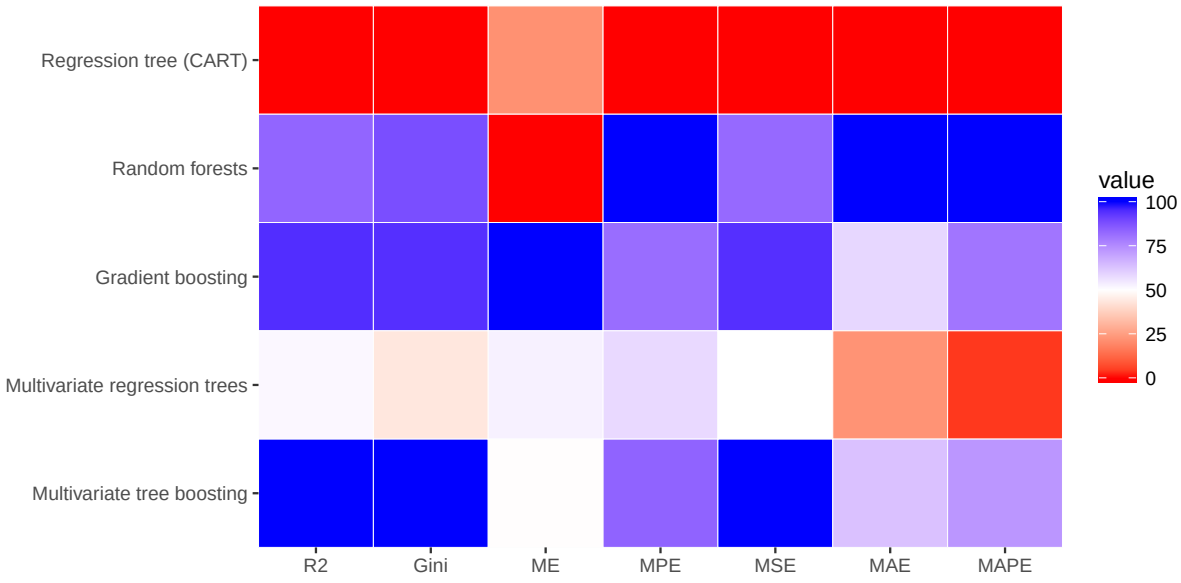

Figure 17: A heatmap of model performance according to the various validation measures

7 Concluding remarks

In this paper, we explore decision trees and its extensions as predictive models for insurance claims. Decsion tree, or tree-based, models have increased in popularity in various disciplines (e.g., psychology, ecology, and biology) and in recent years because of its advantages as an alternative predictive tool. In particular, we extend the usefulness of tree-based predictive models to the case where we have a multivariate response vector. For the empirical part of our investigation, we analyze the LGPIF data that contains claims information about insurance coverage for properties owned by local government units in Wisconsin. We compare the predictive performance of various univariate tree-based models (regression trees, random forests, boosting) against multivariate tree-based models (multivariate regression trees and multivariate tree boosting). Broadly speaking, the multivariate tree-based models generally outperform the univariate tree-based models using a set of different validation measures. In particular, multivariate tree boosting provides the best model according to several validation measures. These results explain the importance of building predictive models that account for dependencies. In actuarial science, insurance, and finance, we have witnessed in recent years the importance of capturing the dependency structure of a multivariate response when developing tools for prediction. For future work, we plan to explore the use of multivariate decision trees where the loss function is different from the sum of squares; an even more interesting extension is the use of a multivariate loss function tailored for a zero-inflated claim structure. Another interesting theoretical and practical problem is related to dealing with zero claims in decision trees.

Appendix A. Summary statistics for both the training and validation datasets

As stated in Section 4, we use the observations for years 2006-2010 as training dataset. In this appendix, we provide useful summary statistics for the explanatory variables used in our analysis: eight (8) are continuous variables and thirteen (13) are categorical variables. In addition, we also provide summary statistics of the response variables as well as the explanatory variables for year 2011, observations of which were used as validation dataset.

Table A.1: Summary statistics of the explanatory variables, 2006-2010 (training dataset).

Continuous variables	Description	Minimum	1st Quantile	Mean	Median	3st Quantile	Maximum
CoverageBC	Log of the coverage amount of building and contents	0	1.15	2.47	2.5	3.63	7.8
CoverageIM	Log of the coverage amount of contractor's equipment	0	0.03	0.37	0.16	0.43	3.87
CoveragePN	Log of the coverage amount of comprehensive new vehicles	0	0	0.09	0	0.03	3.28
CoveragePO	Log of the coverage amount of comprehensive old vehicles	0	0	0.21	0	0.12	4.82
CoverageCN	Log of the coverage amount of building and contents	0	0	0.07	0	0.02	1.64
CoverageCO	Log of the coverage amount of building and contents	0	0	0.16	0	0.1	3.06
lnDeductBC	Log of the deductible amount of building and contents	0	6.21	7.14	6.91	7.82	11.51
lnDeductIM	Log of the deductible amount of contractor's equipment	0	6.21	5.34	6.21	6.21	9.21

Categorical variables		Proportions
IsRC	Indicator for replacement cost (for motor vehicles)	23.48 %
TypeCity	Indicator for city entity	14.00 %
TypeCounty	Indicator for county entity	5.78 %
TypeMisc	Indicator for miscellaneous entity	11.04 %
TypeSchool	Indicator for school entity	28.17 %
TypeTown	Indicator for town entity	17.28 %
TypeVillage	Indicator for village entity	23.73 %
NoClaimCreditBC	Indicator for no building and contents claims in prior year	32.83 %
NoClaimCreditIM	Indicator for no inland marine claims in prior year	42.10 %
NoClaimCreditPN	Indicator for no comprehensive new vehicles (PN) claims in prior year	10.96 %
NoClaimCreditPO	Indicator for no comprehensive old vehicles claims in prior year	17.02 %
NoClaimCreditCN	Indicator for no new vehicle collision claims in prior year	8.97 %
NoClaimCreditCO	Indicator for no old vehicle collision claims in prior year	14.02 %

Table A.2: Summary statistics of the variables in the 2011 validation dataset.

Response variables	Description	Percent of zeroes	Minimum	1st Quantile	Mean	Median	3st Quantile	Maximum
yAvgBC	Log of the average building and contents claim size	70.13	0.69	8.18	9	8.84	9.78	13.23
yAvgIM	Log of the average contractor's equipment claim size	95.72	0.69	7.27	7.98	8.1	8.86	11.35
yAvgPN	Log of the average comprehensive new vehicles claim size	93.72	4.19	7.18	7.73	7.8	8.13	11.38
yAvgPO	Log of the average comprehensive old vehicles claim size	94.72	3.71	6.79	7.4	7.53	8.08	11.17
yAvgCN	Log of the average new vehicle collision claim size	94.08	6.22	7.14	7.81	7.61	8.32	10.18
yAvgCO	Log of the average old vehicle collision claim size	94.44	5.42	7.6	8.26	8.31	8.84	11.23
Continuous variables	**Description**		**Minimum**	**1st Quantile**	**Mean**	**Median**	**3st Quantile**	**Maximum**
CoverageBC	Log of the coverage amount of building and contents		0	1.24	2.58	2.62	3.76	7.78
CoverageIM	Log of the coverage amount of contractor's equipment		0	0.03	0.4	0.18	0.46	4.05
CoveragePN	Log of the coverage amount of comprehensive new vehicles		0	0	0.09	0	0.02	3.36
CoveragePO	Log of the coverage amount of comprehensive old vehicles		0	0	0.22	0	0.11	4.58
CoverageCN	Log of the coverage amount of building and contents		0	0	0.07	0	0	1.65
CoverageCO	Log of the coverage amount of building and contents		0	0	0.17	0	0.09	2.94
lnDeductBC	Log of the deductible amount of building and contents		0	6.21	7.22	6.91	7.82	11.51
lnDeductIM	Log of the deductible amount of contractor's equipment		0	6.21	5.43	6.21	6.21	8.52
Categorical variables					**Proportions**			
IsRC	Indicator for replacement cost (for motor vehicles)				23.41 %			
TypeCity	Indicator for city entity				14.03 %			
TypeCounty	Indicator for county entity				6.47 %			
TypeMisc	Indicator for miscellaneous entity				11.66 %			
TypeSchool	Indicator for school entity				27.60 %			
TypeTown	Indicator for town entity				16.48 %			
TypeVillage	Indicator for village entity				23.73 %			
NoClaimCreditBC	Indicator for no building and contents claims in prior year				50.73 %			
NoClaimCreditIM	Indicator for no inland marine claims in prior year				0.00 %			
NoClaimCreditPN	Indicator for no comprehensive new vehicles (PN) claims in prior year				0.00 %			
NoClaimCreditPO	Indicator for no comprehensive old vehicles claims in prior year				0.00 %			
NoClaimCreditCN	Indicator for no new vehicle collision claims in prior year				0.00 %			
NoClaimCreditCO	Indicator for no old vehicle collision claims in prior year				0.00 %			

Appendix B. The impact of each explanatory variable using multivariate tree boosting

While boosted regression tree models are considered black-box, we can visualize the effect of the explanatory variables on the response variable using partial dependence plots. These plots provide a visual effect after accounting and holding the average effects of all other explanatory variables. It helps to detect explanatory variables with non-linear effects or interactions. However, these plots cannot be similarly interpreted to the effect of coefficients in an ordinary regression framework because of possible biases present when there are interactions between explanatory variables. See, for example, [5].

Figure 18 shows partial dependence plots for each explanatory variables that are directly associated with yAvgBC. This figure also includes variable importance information as shown in figure 15 to see how the important explanatory variables drive the response variable when holding the average effects of the others. From the figure, we can see the non-linear effects of CoverageBC and lnDeductBC on yAvgBC.

Figure 19 shows partial dependence plots for each explanatory variables that are from other coverages and that are not directly associated with yAvgBC. This figure also demonstrates the importance of accounting for the association of the BC coverage with the other coverages. From the figure, we can see that CoverageIM has a 19% relative importance but also non-linear effects on yAvgBC. The figure also shows the non-linear effects of several other explanatory variables, e.g. CoverageCO and CoveragePN, but these same variables also have very low relative importance.

Appendix C. R packages for decision trees

Table 6: R packages for decision trees with tuning hyperparameters.

R package	Description
rpart	Classification and regression tree (CART)
cp	complexity parameter
minsplit	minimum number of observations in a node in order to be considered for splitting
maxdepth	maximum depth of any node of the final tree
mvpart	Multivariate regression trees
minauto	automatic choice for minsplit based on number of observations
xv	choices for the size of tree based on cross-validation: “1se” one standard error rule “min” smallest cross-validation error “pick” interactively pick size of tree “none” no cross-validation
gbm	Gradient boosting
mvtboost	Multivariate tree boosting
n.trees	number of additive trees (iterations)
interaction.depth	maximum depth of variable interactions: 1 implies an additive model 2 means a model with up to 2-way interactions
n.minobsinnode	minimum number of observations in the region
shrinkage	shrinkage parameter (the learning rate)

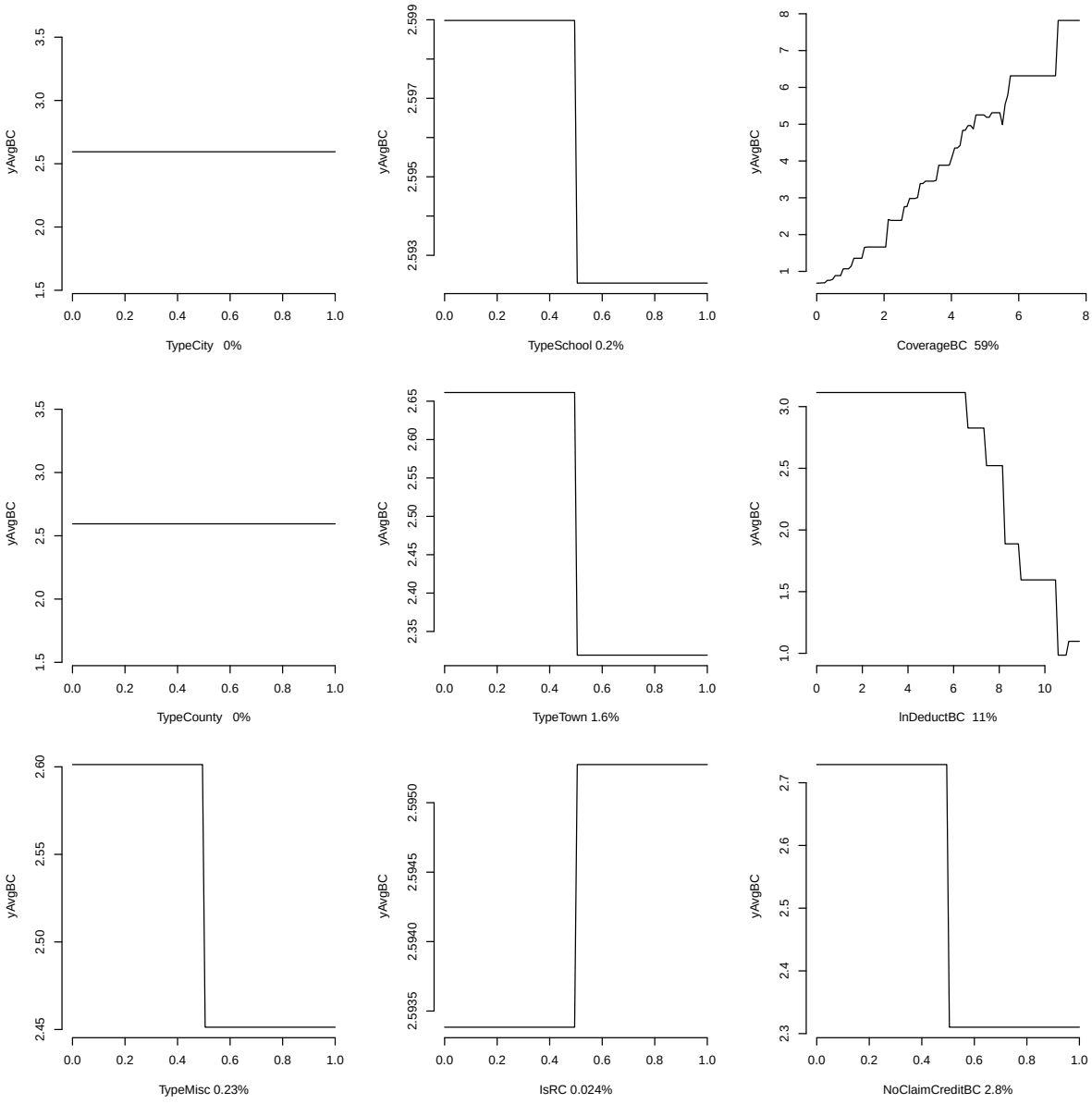

Figure 18: Partial dependence plots of yAvgBC with explanatory variables that are directly associated with the BC coverage

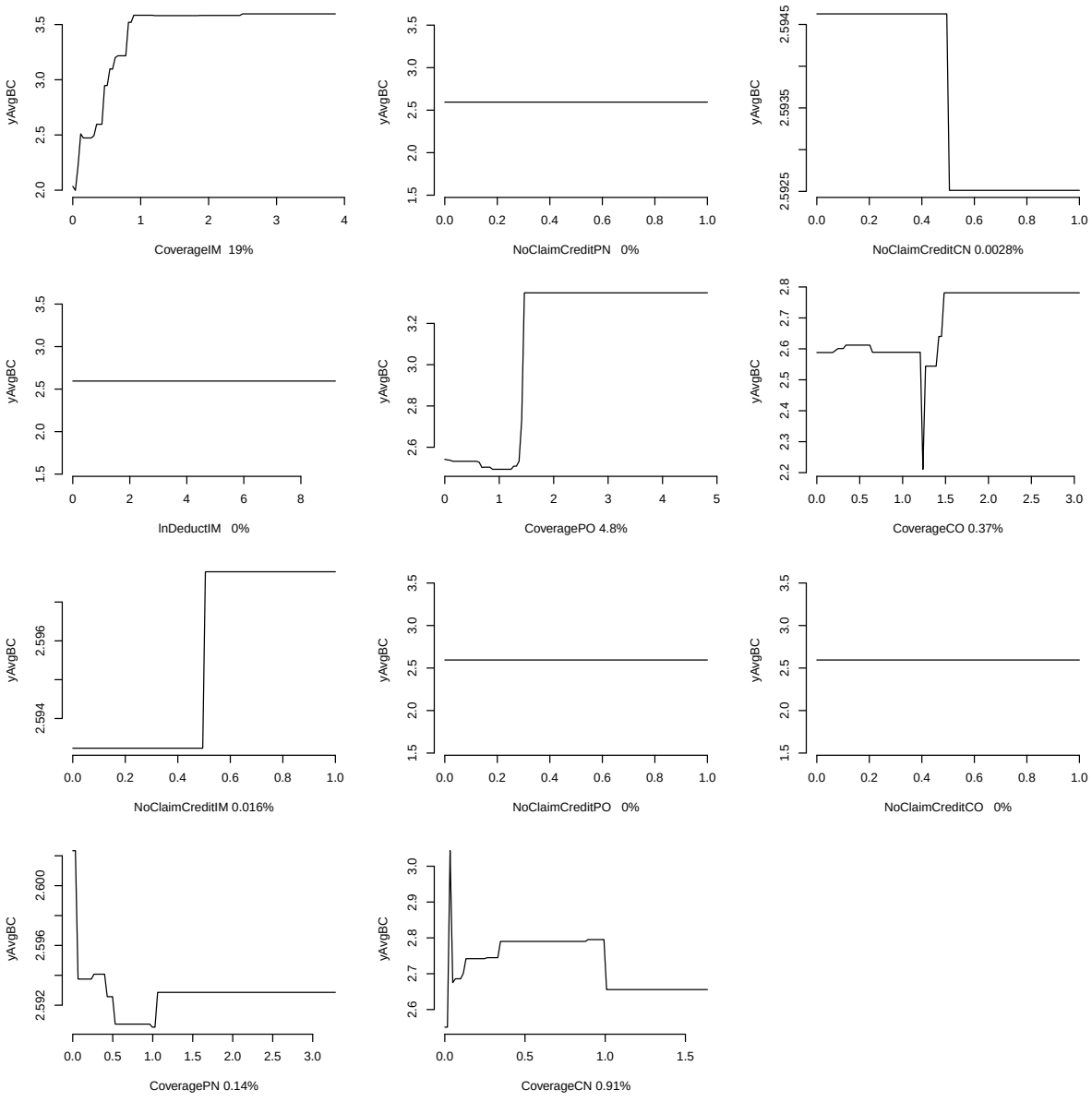

Figure 19: Partial dependence plots of yAvgBC with explanatory variables that are not directly associated with the BC coverage

Appendix D. Validation measures

For purposes of comparing the models in this paper, we use the following validation measures:

- Coefficient of Determination: $R^2 = 1 - \frac{\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2}{\sum_{i=1}^{N}\left(y_i - \frac{1}{n}\sum_{i=1}^{n} y_i\right)^2}$
- Gini Index: $Gini = 1 - \frac{2}{N-1}\left(N - \frac{\sum_{i=1}^{N} i\tilde{y}_i}{\sum_{i=1}^{N}\tilde{y}_i}\right)$,
  where $\tilde{y}$ is the corresponding observed value y after ranking, from lowest to highest, the corresponding predicted values $\widehat{y}$. To accommodate ties in the ranking, uniform random sampling is used.
- Mean Error: $ME = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)$
- Mean Percentage Error: $MPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\widehat{y}_i - y_i}{y_i}$
- Mean Squared Error: $MSE = \frac{1}{N}\sum_{i=1}^{N}(\widehat{y}_i - y_i)^2$
- Mean Absolute Error: $MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\widehat{y}_i - y_i\right|$
- Mean Absolute Percentage Error: $MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\widehat{y}_i - y_i}{y_i}\right|$

We generally make the following conclusions regarding these measures:

- Higher R^2 is better;
- Higher Gini index is better;
- Lower ME is better;
- Lower MPE is better;
- Lower MSE is better;
- Lower MAE is better; and
- Lower MAPE is better;

Acknowledgment: We would like to thank the Society of Actuaries for the funding support of this research project through our Centers of Actuarial Excellence (CAE) grant on data mining. The data used in this paper was provided by Gee Lee and Edward W. (Jed) Frees of the University of Wisconsin in Madison; we extend our appreciation to them for allowing us to use the data. We would also like to thank the participants of the 10th Conference in Actuarial Science and Finance on Samos for the feedback. Zhiyu would like to acknowledge the doctoral student travel award provided by the University of Connecticut (UConn) Graduate School .

References

[1] Breiman, L. (2001). Random forests. *Mach. Learn. 45*(1), 5–32.
[2] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Taylor & Francis, Boca Raton FL.
[3] De'ath, G. (2002). Multivariate regression trees: A new technique for modeling species-environmental relationships. *Ecology 83*(4), 1105–1117.
[4] Deprez, P., Shevchenko, P. V., and Wüthrich, M. V. (2017). Machine learning techniques for mortality modeling. *Eur. Actuar. J. 7*(2), 337–352.
[5] Elith, J., Leathwick, J. R., and Hastie, T. (2008). A working guide to boosted regression trees. *J. Anim. Ecol. 77*(4), 802–813.
[6] Frees, E. W. and Lee, G. (2015). Rating endorsements using generalized linear models. *Variance 10*(1), 51–74.
[7] Frees, E. W., Lee, G., and Yang, L. (2016). Multivariate frequency-severity regression models in insurance. *Risks 4*(4), 36.
[8] Frees, E. W. and Valdez, E. A. (2008). Hierarchical insurance claims modeling. *J. Amer. Statist. Assoc. 103*(484), 1457–1469.
[9] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist. 29*(5), 1189–1232.
[10] Friedman, J. H. (2002). Stochastic gradient boosting. *Comput. Statist. Data Anal. 38*(4), 367–378.
[11] Friedman, J. H. and Meulman, J. J. (2003). Multiple additive regression trees with application in epidemiology. *Statist. Med. 22*(9), 1365–1381.

[12] Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika 58*(3), 453–467.
[13] Guelman, L. (2012). Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Syst. Appl. 39*(3), 3659–3667.
[14] Hartigan, J. A. (1975). *Clustering Algorithms*. Wiley, New York.
[15] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York.
[16] Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *J. Comput. Graph. Statist. 15*(3), 651–674.
[17] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer, New York.
[18] Jolliffe, I. T. (1986). *Principal Component Analysis and Factor Analysis*. Springer, New York.
[19] Lee, S. C. and Lin, S. (2018). Delta boosting machine with application to general insurance. *N. Am. Actuar. J. 22*(3), 405–425.
[20] Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News 2/3*, 18–22.
[21] Loh, W.-Y. (2014). Fifty years of classification and regression trees. *Int. Stat. Rev. 82*(3), 329–348.
[22] Maroco, J., Silva, D., Rodrigues, A., Guerreiro, M., Santana, I., and de Mendonça, A. (2011). Data mining methods in the prediction of dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. *BMC Res. Notes 4*(299), 14.
[23] Milborrow, S. (2016). Plotting rpart trees with the rpart.plot package. Available at http://www.milbo.org/rpart-plot/prp.pdf.
[24] Miller, P. J., Lubke, G. H., McArtor, D. B., and Bergeman, C. (2016). Finding structure in data using multivariate tree boosting. *Psychol. Meth. 21*(4), 583–602.
[25] Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *J. Amer. Stat. Assoc. 58*(302), 415–434.
[26] Muñoz, J. and Felicísimo, Á. M. (2004). Comparison of statistical methods commonly used in predictive modelling. *J. Veget. Sci. 15*(2), 285–292.
[27] Olbricht, W. (2012). Tree-based methods: a useful tool for life insurance. *Eur. Actuar. J. 2*(1), 129–147.
[28] Pande, A., Li, L., Rajeswaran, J., Ehrlinger, J., Kogalur, U. B., Blackstone, E. H., and Ishwaran, H. (2017). Boosted multivariate trees for longitudinal data. *Mach. Learn. 106*(2), 277–305.
[29] Ridgeway, G. (2018). *gbm: Generalized Boosted Regression Models*. R package version 2.1.4. Available on CRAN.
[30] Ridgeway, G. (2007b). *Generalized Boosted Models: A guide to the gbm package*. Available at https://cran.r-project.org/web/packages/gbm/vignettes/gbm.pdf.
[31] Segal, M. and Xiao, Y. (2011). Multivariate random forests. *Data Min. Knowl. Discov. 1*(1), 80–87.
[32] Shi, P. and Yang, L. (2018). Pair copula constructions for insurance experience rating. *J. Amer. Stat. Assoc. 113*(521), 122–133.
[33] Strobl, C., Boulesteix, A.-L., Zeileis, A., and Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics 8*, 25.
[34] Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*. Pearson Education Limited, Harlow.
[35] Ter Braak, C. J. (1986). Canonical correspondence analysis: a new eigenvector technique for multivariate direct gradient analysis. *Ecology 67*(5), 1167–1179.
[36] Therneau, T., Atkinson, B., and Ripley, B. (2018). rpart: Recursive partitioning and regression trees. R package version 4.1–13. Available on CRAN.
[37] Thuiller, W., Araújo, M. B., and Lavorel, S. (2003). Generalized models vs. classification tree analysis: predicting spatial distributions of plant species at different scales. *J. Veget. Sci. 14*(5), 669–680.
[38] Wüthrich, M. V. (2018). Machine learning in individual claims reserving. *Scand. Actuar. J. 2018*(6), 465–480.
[39] Wüthrich, M. V. and Buser, C. (2018). Data analytics for non-life insurance pricing. Available at https://dx.doi.org/10.2139/ssrn.2870308.
[40] Xiao, Y. and Segal, M. R. (2009). Identification of yeast transcriptional regulation networks using multivariate random forests. *PLoS Comput. Biol. 5*(6), e1000414.