Imbalanced learning for insurance using modified loss functions in tree-based models

Citation for published version:
https://doi.org/10.1016/j.insmatheco.2022.04.010

Digital Object Identifier (DOI):
10.1016/j.insmatheco.2022.04.010

Link:
Link to publication record in Heriot-Watt Research Portal

Document Version:
Peer reviewed version

Published In:
Insurance: Mathematics and Economics

General rights
Copyright for the publications made accessible via Heriot-Watt Research Portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
Heriot-Watt University has made every reasonable effort to ensure that the content in Heriot-Watt Research Portal complies with UK legislation. If you believe that the public display of this file breaches copyright please contact open.access@hw.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Imbalanced Learning for Insurance using Modified Loss Functions in Tree-Based Models

Changyue Hu† Zhiyu Quan† Wing Fung Chong‡

June 7, 2022

Abstract

Tree-based models have gained momentum in insurance claim loss modeling; however, the point mass at zero and the heavy tail of insurance loss distribution pose the challenge to apply conventional methods directly to claim loss modeling. With a simple illustrative dataset, we first demonstrate how the traditional tree-based algorithm’s splitting function fails to cope with a large proportion of data with zero responses. To address the imbalance issue presented in such loss modeling, this paper aims to modify the traditional splitting function of Classification and Regression Tree (CART). In particular, we propose two novel modified loss functions, namely, the weighted sum of squared error and the sum of squared Canberra error. These modified loss functions impose a significant penalty on grouping observations of non-zero response with those of zero response at the splitting procedure, and thus significantly enhance their separation. Finally, we examine and compare the predictive performance of such modified tree-based models to the traditional model on synthetic datasets that imitate insurance loss. The results show that such modification leads to substantially different tree structures and improved prediction performance.

Keywords: Predictive model of insurance claims, imbalanced learning, custom loss, Canberra distance, regression tree, tree-based algorithms.

*Department of Mathematics, University of Illinois at Urbana-Champaign. Email: ch47@illinois.edu.
†Department of Mathematics, University of Illinois at Urbana-Champaign. Email: zquan@illinois.edu.
‡Maxwell Institute for Mathematical Sciences and Department of Actuarial Mathematics and Statistics, Heriot-Watt University. Email: alfred.chong@hw.ac.uk.
1 Introduction

The modeling of insurance claims has always been a core subject in actuarial science. With the rapid development of data science, the literature has been applying existing techniques to build machine learning models on insurance loss data. For review, see Blier-Wong et al. (2021).

In this paper, we focus on modifying one particular machine learning model, the base learner for tree-based models, the Classification and Regression Tree (CART), which was first introduced in Breiman et al. (1984). CART builds a binary tree by recursively partitioning observations in the dataset. As a non-parametric model, CART does not make any distributional assumptions. Unlike linear regression models, CART can account for interactions in its binary tree structure since it does not require the assumption of linear additivity. Also, the model can be easily interpreted from the tree structure. These good properties of CART have contributed to its applications in actuarial science and insurance. For the details, see, for example, Quan (2019).

However, it is evident that insurance loss datasets usually contain a high percentage of zero claims, which poses significant challenges to directly applying any conventional regression models. Although imbalance is typically a concept within classification problems, we carry over this term to illustrate similar issues in actuarial loss modeling regression tasks. The predictive modeling in insurance claims presents an imbalanced learning problem due to the majority of the data points being zeros.

There is a growing concern about imbalanced learning across a variety of fields, including finance, biomedical engineering, etc. Except for claim prediction in the insurance industry, many other real-world classification problems have imbalanced datasets, including fraud detection, protein identification, and cancer diagnosis. In imbalanced learning, the fundamental issue is that the imbalanced data can significantly degrade the performance of most standard learning algorithms. As most standard algorithms assume balanced class distributions or universal misclassification costs, when faced with imbalanced data, these algorithms fail to depict data characteristics properly and therefore yield poor accuracy across classes; see He and Garcia (2009). To address imbalanced data issues, many approaches have been proposed, the majority of which are based on resampling techniques, ensemble methods, and cost-sensitive learning; see Guo et al. (2017).

Resampling techniques are employed to rebalance the sample space for an imbalanced dataset. It includes adding more observations/samples from the minority class (oversampling), removing observations/samples from the majority class (undersampling), or combining both. One popular over-sampling method is the Synthetic Minority Over-sampling Technique (SMOTE), introduced by Chawla et al. (2002). Unlike random oversampling, which naively replicates samples from the minority class, SMOTE creates synthetic samples by interpolation. The minority class is over-sampled by taking each sample from the minority, considering its k nearest neighbors (in feature space), and then populating synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. As a result, resampling techniques attempt to modify distribution by balancing class proportions artificially; however,
it may not be desirable in the insurance domain. It might be difficult for actuaries to accept or justify synthetic claims created by algorithms.

Another popular approach to handling imbalanced datasets is the ensemble method, which combines multiple base learners to improve the model performance; see López et al. (2013). Most ensemble methods can be divided into three categories: parallel-based ensembles, iterative-based ensembles, and a hybrid of both; see Guo et al. (2017). As the most common parallel ensemble method, bagging (or bootstrap aggregation) improves the generalization ability of the model by creating random sub-samples from the original dataset with replacement and then combining the predictions generated by each base learner trained on each bootstrap sub-sample. Boosting is the most common iterative ensemble method. The first applicable boosting algorithm, Ada-boost, developed by Freund and Schapire (1996), gives higher weights to samples that fail to be classified correctly, thus forces the future base learners to focus more on learning these misclassified samples. In recent years, the tree-based ensemble methods have achieved great success in predictive performance. See Guelman et al. (2012), Olbricht (2012), Guelman et al. (2015), Wüthrich (2018), Yang et al. (2018), Lopez et al. (2019), Quan et al. (2021), Henckaerts et al. (2021), and the references therein, for additional examples of actuarial applications. Despite the success achieved in prediction performance, limitations in interpretation remain to be overcome.

On the other hand, cost-sensitive learning targets the imbalanced learning problem by using different cost matrices that describe the costs for misclassifying any particular class example; see Elkan (2001). Classification algorithms typically assume that the costs of all misclassifications are similar; however, in the real world, these misclassifications often have different interpretations. Therefore, the cost of different misclassifications should vary according to actual applications. Cost-sensitive learning modifies the cost of misclassification by adding penalties to misclassified predictions related to the objects of interest. Inspired by the idea of cost-sensitive learning, in this paper, we pose a penalty by assigning different weights for prediction errors of observations with zero and non-zero responses by adopting novel loss functions; thereby, we inject the “classification” of zero and non-zero claims into our regression models.

The main contribution of this paper is that we propose to adopt modified loss functions in traditional tree-based models. Indeed, the above alluded issues stem from the fact that the traditional sum of squared error (SSE) loss tends to only partition the observations with small and large claims; consequently, the zero responses would not be standalone and are grouped together with small but non-zero claims. We put forward the weighted sum of squared error (WSSE) and the sum of squared Canberra error (SSCE) loss functions to naturally impose a penalty on prediction errors for observations with zero responses and remedy the imbalanced learning problem.

The rest of this paper is structured as follows. In Section 2, we review the CART algorithm under a regression framework and investigate the splitting performance of the regression tree using a simplified example. In particular, we illustrate that the traditional SSE loss function is unsuitable for insurance datasets containing a high proportion of observations with zero response. In Section 3, we propose the WSSE and the Canberra loss as the node impurity function for CART and introduce the WSSE tree model and the Canberra tree model. We implement the
two proposed regression tree models using R and compare the goodness of splitting performance with that of the regression tree using SSE. Section 4 provides a detailed simulation study and discussion of the performance of different regression tree models. Finally, concluding remarks are provided in Section 5.

2 Review and Potential Pitfall of Regression Tree

This section reviews CART algorithm, which is pioneered by Breiman et al. (1984), with the focus on the regression framework; that is, the response variable is continuous. A motivating example is provided to unveil a potential pitfall of the classical splitting criteria in CART; in particular, the fabricated dataset contains a high percentage of imbalanced observations with zero for their response variable, which is commonly observed from insurance claims.

2.1 Overview of Regression Tree

Consider a dataset \((X, y) = ((X_1, y_1), (X_2, y_2), \ldots, (X_N, y_N))^T\) with \(N\) observations. For the \(i\)-th observation, where \(i = 1, 2, \ldots, N\), \(X_i = (X_{i1}, X_{i2}, \ldots, X_{ip})\) is a vector of \(p\) explanatory variables sampled from a space \(X = X_1 \times X_2 \times \cdots \times X_p\), while \(y_i\) is a response variable sampled from a space \(Y\). For our purpose, in loss modeling context, \(Y\) is a subset of the non-negative real numbers set \(\mathbb{R}_+\).

In general, tree-based models aim to partition the space \(X\) of explanatory variables into \(M\) disjoint regions, which are denoted as \(R_1, R_2, \ldots, R_M\); for each region \(R_m\), where \(m = 1, 2, \ldots, M\), a constant \(c_m \in \mathbb{R}_+\) is assigned as a predicted value for observations falling into the region \(R_m\). Therefore, the regression tree is given by: for each \(i = 1, 2, \ldots, N\), the predicted value

\[
\hat{y}_i = T(X_i; \theta) = \sum_{m=1}^M c_m 1_{R_m}(X_i),
\]

where \(\theta = (R_1, R_2, \ldots, R_M, c_1, c_2, \ldots, c_M)\) is the vector of parameters for the regression tree, and \(1_{R_m}(X_i) = 1\), if \(X_i \in R_m \subseteq X\), while \(1_{R_m}(X_i) = 0\), if \(X_i \not\in R_m\). Denote \(\hat{\theta} = (\hat{R}_1, \hat{R}_2, \ldots, \hat{R}_M, \hat{c}_1, \hat{c}_2, \ldots, \hat{c}_M)\) as the optimal regions and the optimal assigned values to these regions. The classical loss function, to determine the optimal parameters for the regression tree, is given by the sum of squared errors (SSE). Therefore,

\[
\hat{\theta} = \arg \min_{\theta \in \Theta} L(y, \hat{y}) = \arg \min_{\theta \in \Theta} \sum_{i=1}^N (y_i - \hat{y}_i)^2,
\]

where the set of all feasible vectors of parameters

\[
\Theta = \{(R_1, R_2, \ldots, R_M, c_1, c_2, \ldots, c_M) \in \mathcal{P}(X) \times \mathcal{P}(X) \times \cdots \times \mathcal{P}(X) \times \mathcal{Y} \times \cdots \times \mathcal{Y} : \\
\cup_{m=1}^M R_m = X, \text{ and } R_{m_1} \cap R_{m_2} = \emptyset \text{ for } m_1 \neq m_2\},
\]
in which \( \mathcal{P}(\mathcal{X}) \) is the power set of \( \mathcal{X} \), i.e., the set of all subsets of \( \mathcal{X} \). Necessarily, for each \( m = 1, 2, \ldots, M \),

\[
\hat{c}_m = \frac{1}{N_m} \sum_{i : \mathbf{X}_i \in \mathcal{R}_m} y_i,
\]

where \( N_m = \left| \left\{ i = 1, 2, \ldots, N : \mathbf{X}_i \in \mathcal{R}_m \right\} \right| \) is the number of observations with explanatory variables falling into the optimal region \( \mathcal{R}_m \). It is easy to show the constant \( \hat{c}_m \) that minimizes the expected value of the SSE is the mean value of \( y_i \) in the region \( \mathcal{R}_m \).

The CART algorithm entails a more computationally efficient way to identify the optimal parameters for the regression tree via recursive binary splittings. In the following, fix a binary splitting step \( u = 1, 2, \ldots, \).

Denote \((\mathbf{X}^{(u)}, \mathbf{y}^{(u)}) = \left( \begin{pmatrix} \mathbf{X}^{(u)}_1, y^{(u)}_1 \\ \mathbf{X}^{(u)}_2, y^{(u)}_2 \\ \vdots \\ \mathbf{X}^{(u)}_{N^{(u)}}, y^{(u)}_{N^{(u)}} \end{pmatrix} \right)^T\) as the remaining dataset with \( N^{(u)} \) observations, which serves as a parent node and depends on the former splitting steps \( 1, 2, \ldots, u - 1 \). In particular, when \( u = 1 \), \((\mathbf{X}^{(1)}, \mathbf{y}^{(1)}) = (\mathbf{X}, \mathbf{y}) \) and \( N^{(1)} = N \); in the sequel, the superscript shall be dropped in this case. This dataset in the parent node is potentially split into two daughter nodes with respective datasets:

\[
(\mathbf{X}^{(u,l)}, \mathbf{y}^{(u,l)}) = \left( \begin{pmatrix} \mathbf{X}^{(u,l)}_1, y^{(u,l)}_1 \\ \mathbf{X}^{(u,l)}_2, y^{(u,l)}_2 \\ \vdots \\ \mathbf{X}^{(u,l)}_{N^{(u,l)}}, y^{(u,l)}_{N^{(u,l)}} \end{pmatrix} \right)^T, \quad \text{and} \\
(\mathbf{X}^{(u,r)}, \mathbf{y}^{(u,r)}) = \left( \begin{pmatrix} \mathbf{X}^{(u,r)}_1, y^{(u,r)}_1 \\ \mathbf{X}^{(u,r)}_2, y^{(u,r)}_2 \\ \vdots \\ \mathbf{X}^{(u,r)}_{N^{(u,r)}}, y^{(u,r)}_{N^{(u,r)}} \end{pmatrix} \right)^T,
\]

in which, there exist \( j^{(u)} = 1, 2, \ldots, p \) and \( s^{(u)} \in \mathcal{X}_{j^{(u)}}^{(u)} \) such that, for any \( i = 1, 2, \ldots, N^{(u,l)} \), \( X_{ij^{(u)}}^{(u,l)} \leq s^{(u)} \), while, for any \( i = 1, 2, \ldots, N^{(u,r)} \), \( X_{ij^{(u)}}^{(u,r)} > s^{(u)} \), if the explanatory variable \( X_{j^{(u)}} \) is continuous; if the explanatory variable \( X_{j^{(u)}} \) is categorical, there exists \( s^{(u)} \in \mathcal{P}(\mathcal{X}_{j^{(u)}}) \) such that, for any \( i = 1, 2, \ldots, N^{(u,l)} \), \( X_{ij^{(u)}}^{(u,l)} \in s^{(u)} \), while, for any \( i = 1, 2, \ldots, N^{(u,r)} \), \( X_{ij^{(u)}}^{(u,r)} \notin s^{(u)} \). Note that the datasets in the two daughter nodes depend on the choice of \( j^{(u)} \) and \( s^{(u)} \); for notational simplicity, such dependence do not spell out explicitly. If the explanatory variable \( X_{j^{(u)}} \) is continuous, define two regions of the space \( \mathcal{X}_{j^{(u)}} \) by

\[
\mathcal{R}_{j^{(u)}, s^{(u)}}^{(u,l)} = \left\{ x_{j^{(u)}} \in \mathcal{X}_{j^{(u)}} : x_{j^{(u)}} \leq s^{(u)} \right\}, \quad \text{and} \\
\mathcal{R}_{j^{(u)}, s^{(u)}}^{(u,r)} = \left\{ x_{j^{(u)}} \in \mathcal{X}_{j^{(u)}} : x_{j^{(u)}} > s^{(u)} \right\};
\]

if the explanatory variable \( X_{j^{(u)}} \) is categorical, define

\[
\mathcal{R}_{j^{(u)}, s^{(u)}}^{(u,l)} = \left\{ x_{j^{(u)}} \in \mathcal{X}_{j^{(u)}} : x_{j^{(u)}} \in s^{(u)} \right\} = s^{(u)}, \quad \text{and} \\
\mathcal{R}_{j^{(u)}, s^{(u)}}^{(u,r)} = \left\{ x_{j^{(u)}} \in \mathcal{X}_{j^{(u)}} : x_{j^{(u)}} \notin s^{(u)} \right\} = \left( s^{(u)} \right) \mathcal{C}.
\]

Similar to (1), for each \( i = 1, 2, \ldots, N^{(u)} \), the predicted value after the \( u \)-th split is

\[
\hat{y}_i^{(u)} = c^{(u,l)}(j^{(u)}, s^{(u)}) \mathbf{1}_{\mathcal{R}^{(u,l)}}(j^{(u)}, s^{(u)}) (X^{(u)}_{ij^{(u)}}) + c^{(u,r)}(j^{(u)}, s^{(u)}) \mathbf{1}_{\mathcal{R}^{(u,r)}}(j^{(u)}, s^{(u)}) (X^{(u)}_{ij^{(u)}}),
\]
where the last equality is due to (4). Necessarily, similar to (3), for some $c^{(u,l)}(j^{(u)}, s^{(u)})$, $c^{(u,r)}(j^{(u)}, s^{(u)}) \in \mathbb{R}_+$. Following the classical loss function for the tree-based models as in (2), in this binary splitting, the optimal parameters $\hat{j}^{(u)}$, $\hat{s}^{(u)}$, $\hat{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)})$, and $\hat{c}^{(u,r)}(\hat{j}^{(u)}, \hat{s}^{(u)})$ are given by minimizing the SSE; that is,

\[
\left(\hat{j}^{(u)}, \hat{s}^{(u)}, \hat{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)}), \hat{c}^{(u,r)}(\hat{j}^{(u)}, \hat{s}^{(u)})\right) = \arg\min_{j^{(u)}=1,2,\ldots; p; s^{(u)} \in X_{j^{(u)}} \text{ or } s^{(u)} \in \mathcal{P}(X_{j^{(u)})}} \quad L \left(\hat{y}^{(u)}, \hat{y}^{(u)}\right) = \arg\min_{j^{(u)}=1,2,\ldots; p; s^{(u)} \in X_{j^{(u)}} \text{ or } s^{(u)} \in \mathcal{P}(X_{j^{(u)})}} \quad \sum_{i=1}^{N^{(u)}} \left(y_i^{(u)} - \hat{y}_i^{(u)}\right)^2
\]

\[
= \arg\min_{j^{(u)}=1,2,\ldots; p; s^{(u)} \in X_{j^{(u)}} \text{ or } s^{(u)} \in \mathcal{P}(X_{j^{(u)})}} \quad \sum_{i=1}^{N^{(u,l)}} \left(y_i^{(u,l)} - c^{(u,l)}(j^{(u), s^{(u)})}\right)^2 + \sum_{i=1}^{N^{(u,r)}} \left(y_i^{(u,r)} - c^{(u,r)}(j^{(u), s^{(u)})}\right)^2,
\]

where the last equality is due to (4). Necessarily, similar to (3),

\[
\hat{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)}) = \frac{1}{N^{(u,l)}} \sum_{i: X_{j^{(u)}} \in R^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)})} y_i^{(u)} = \frac{1}{N^{(u,l)}} \sum_{i=1}^{N^{(u,l)}} y_i^{(u,l)},
\]

\[
\hat{c}^{(u,r)}(\hat{j}^{(u)}, \hat{s}^{(u)}) = \frac{1}{N^{(u,r)}} \sum_{i: X_{j^{(u)}} \in R^{(u,r)}(\hat{j}^{(u)}, \hat{s}^{(u)})} y_i^{(u)} = \frac{1}{N^{(u,r)}} \sum_{i=1}^{N^{(u,r)}} y_i^{(u,r)}.
\]

Hence, the optimal parameters $\hat{j}^{(u)}$ and $\hat{s}^{(u)}$ are given by

\[
\left(\hat{j}^{(u)}, \hat{s}^{(u)}\right) = \arg\min_{j^{(u)}=1,2,\ldots; p; s^{(u)} \in X_{j^{(u)}} \text{ or } s^{(u)} \in \mathcal{P}(X_{j^{(u)})}} \quad \sum_{i=1}^{N^{(u,l)}} \left(y_i^{(u,l)} - \hat{c}^{(u,l)}(j^{(u), s^{(u)})}\right)^2 + \sum_{i=1}^{N^{(u,r)}} \left(y_i^{(u,r)} - \hat{c}^{(u,r)}(j^{(u), s^{(u)})}\right)^2,
\]

which is also known as the ANOVA best split.

To summarize, the CART algorithm performs the above binary splittings recursively, starting from the $N$ observations in the original dataset. A parent node, at a binary splitting step $u = 1, 2, \ldots$, will be split into two homogeneous sub-regions or daughter nodes, if (i) the number of observations $N^{(u)}$ at the parent node is strictly larger than a hyperparameter $N_0$, which is
also known as the minsplit, i.e. \( N^{(u)} > N \), and (ii) the SSE is strictly reduced with the optimal parameters, i.e.

\[
\sum_{i=1}^{N^{(u)}} \left( y_i^{(u,l)} - \hat{\gamma}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2 + \sum_{i=1}^{N^{(u,r)}} \left( y_i^{(u,r)} - \hat{\gamma}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2 < \sum_{i=1}^{N^{(u)}} \left( y_i^{(u)} - \hat{y}_i^{(u-1)} \right)^2, \tag{8}
\]

where the predicted value, at the previous binary splitting step \( u - 1 \), is \( \hat{y}_i^{(u-1)} = \frac{1}{N^{(u)}} \sum_{k=1}^{N^{(u)}} y_k^{(u)} \), for all \( i = 1, 2, \ldots, N^{(u)} \); otherwise, there will be no further split for this parent node. As a result, a fully grown tree \( T_0 \) for (1) is constructed. To avoid over-fitting, the CART algorithm utilizes cost-complexity pruning to trim the fully grown tree \( T_0 \) while preserve its essential structure. For an overview of the algorithm for \texttt{rpart}, which is the \texttt{R} implementation of CART, and the pruning procedures, see Appendix A. For further reference, see, for example, Therneau and Atkinson (1997).

### 2.2 Motivating Example

Table 1 outlines a dataset to illustrate the potential pitfall of the classical ANOVA split in the CART. In this dataset, the response variable \( Y \) is continuous, while the explanatory variables \( X_1 \) and \( X_2 \) are respectively continuous and categorical. In particular, note that 70% of the observations output zero for their response variable.

<table>
<thead>
<tr>
<th>Obs.</th>
<th>( X_1 )</th>
<th>( X_2 )</th>
<th>( Y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>A</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>B</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>D</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>D</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1: Dataset of motivating example

Fix the first binary splitting step \( u = 1 \). If \( j = 1 \), there are 9 possible splitting points, namely \( s = 1, 2, \ldots, 9 \); for example, if \( s = 3 \), then \( R^{(l)} (1, 3) = \{1, 2, 3\} \) and \( R^{(r)} (1, 3) = \{4, 5, 6, 7, 8, 9, 10\} \). If \( j = 2 \), there are 14 possible subsets after splitting: \{A\},
{B,C,D}, {B}, {A,C,D}, {C}, {A,B,D}, {D}, {A,B,C}, {A,B}, {C,D}, {A,C}, {B,D}, {A,D}, {B,C}; for example, if \( s = \{A,C\} \), then \( R^l(2, \{A,C\}) = \{A,C\} \) and \( R^r(2, \{A,C\}) = \{B,D\} \). However, in this dataset, since the average values of the response variable among the observations in the same categorical level are ordered among the levels, i.e. \( \frac{1}{2} \sum_{i:X_{12} = A} y_i \leq \frac{1}{2} \sum_{i:X_{12} = B} y_i \leq \frac{1}{2} \sum_{i:X_{12} = D} y_i \), it suffices to consider only the following 3 splitting subsets to minimize the objective function in (7): \{A\}, \{A,B\}, \{A,B,C\}. Indeed, for instance, \( \sum_{i=1}^{7} \left( y_i^{l} - \hat{c}^{l}(2, \{A,B\}) \right)^2 + \sum_{i=1}^{3} \left( y_i^{r} - \hat{c}^{r}(2, \{A,B\}) \right)^2 = 2.00 < 12.99 = \sum_{i=1}^{4} \left( y_i^{l} - \hat{c}^{l}(2, \{A,C\}) \right)^2 + \sum_{i=1}^{6} \left( y_i^{r} - \hat{c}^{r}(2, \{A,C\}) \right)^2 \). With a slight abuse of terminology, these 3 splitting subsets shall be called as the splitting points A, B, and C respectively. In general, when there are \( K \) categorical levels for a categorical variable, there are \( 2^{K-1}-1 \) possible splitting subsets; by reordering the observations such that the average values of the response variable among the observations in the same categorical level are ordered among the levels, it suffices to consider only \( K-1 \) splitting subsets (or points, loosely speaking). For more details, see Therneau and Atkinson (1997).

As 70% of the observations output zero for their response variable, a regression tree model is expected to separate these observations from those remaining, even in the first split. More precisely, it is anticipated that, if \( \hat{j} = 1 \), then \( \hat{s} = 7 \), while, if \( \hat{j} = 2 \), then \( \hat{s} = B \), or more precisely, \( \hat{s} = \{A,B\} \).

To check this, define the goodness of split, for any \( j^{(u)} = 1, 2, \ldots, p \) and \( s^{(u)} \in X_{j^{(u)}} \) (resp. \( s^{(u)} \in \mathcal{P} (X_{j^{(u)}}) \)):

\[
G(j^{(u)}, s^{(u)}) = \frac{L(y^{(u)}, \hat{y}^{(u-1)}) - L(y^{(u)}, \hat{y}^{(u)})}{L(y^{(u)}, \hat{y}^{(u-1)})},
\]

(9)

where \( L(y^{(u)}, \hat{y}^{(u-1)}) \) is given by the right-hand-side of (8), and \( L(y^{(u)}, \hat{y}^{(u)}) \) is given by the objective function in the right-hand-side of (7). The goodness of split \( G(j^{(u)}, s^{(u)}) \in [0,1] \) measures the percentage decrease of loss by a potential split \( (j^{(u)}, s^{(u)}) \) at step \( u \). Therefore, the minimization of the SSE in (7) is equivalent to the maximization of the goodness of split in (9). In particular, if \( G(j^{(u)}, s^{(u)}) = 0 \), for all \( j^{(u)} = 1, 2, \ldots, p \) and \( s^{(u)} \in X_{j^{(u)}} \) (resp. \( s^{(u)} \in \mathcal{P} (X_{j^{(u)}}) \)), there will be no split for this parent node.

Figures 1(a) and 1(b) depict the goodness of split by the continuous explanatory variable \( X_1 \) and the categorical explanatory variable \( X_2 \) respectively, which show that, if \( \hat{j} = 1 \), then \( \hat{s} = 8 \), while, if \( \hat{j} = 2 \), then \( \hat{s} = C \), contradicting the expectation. The result may not be the way actuaries would expect since the model cannot clearly separate zero and non-zero claims. However, the results are indeed in line with the ANOVA best split; after all, the classical loss function SSE is designed to partition the observations by similar values of the response variable, but is not engineered to provide any segregated treatment for imbalanced observations with a high percentage of zero in the response variable. Therefore, it is an urgent need for custom-modified loss functions to tailor for insurance applications, which are more sensitive to zero values in the response variable.
3 Modifying Loss Functions in CART Algorithm

In this section, to address the pitfall of regression tree as demonstrated in the motivating example of Section 2.2, we propose two loss functions, namely, the weighted sum of squared errors (WSSE) and the sum of squared Canberra errors, to be implemented in the CART algorithm. The motivating example is also revisited to compare the implementation with the ANOVA best split.

3.1 Weighted Sum of Squared Errors

As mentioned in Section 2.2, the classical SSE loss function of the ANOVA best split does not treat imbalanced observations with zero in the response variable separately but only divides the observations with similar values of the response variable. Therefore, an ad-hoc modification to the classical SSE loss function is assigning different weights on the squared errors for observations with zero in the response variable and those with non-zero response observations. To this end, define the following loss function, which is given by the WSSE:

\[
L_W(y^{(u)}, \hat{y}^{(u)}) = w_0^{(u)} \sum_{i:y_i^{(u)}=0} (y_i^{(u)} - \hat{y}_i^{(u)})^2 + (1 - w_0^{(u)}) \sum_{i:y_i^{(u)} \neq 0} (y_i^{(u)} - \hat{y}_i^{(u)})^2
\]

\[
= w_0^{(u)} \sum_{i:y_i^{(u)}=0} (\hat{y}_i^{(u)})^2 + (1 - w_0^{(u)}) \sum_{i:y_i^{(u)} \neq 0} (y_i^{(u)} - \hat{y}_i^{(u)})^2,
\]

where \( u = 1, 2, \ldots \) is a binary splitting step, \( w_0^{(u)} \in [0, 1] \), and \( \hat{y}^{(u)} \) is given by (4). In particular, when \( w_0^{(u)} = 0.5 \), the WSSE loss function reduces to the classical SSE function.
While the weight $w_0^{(u)}$ is flexible, it should reflect the relative importance of the squared errors for observations with zero in the response variable. For example, it could be a universal hyperparameter of the tree-based model independent of the splitting step $u$; or, it could be the percentage of zero response observations in the original dataset $(X, y)$ before any splittings, that, for any $u = 1, 2, \ldots$,$$w_0^{(u)} = \frac{N_0}{N}$$universally, where $N_0 = |\{i = 1, 2, \ldots, N : y_i = 0\}|$; or, it could even be the percentage of zero response observations in the remaining dataset of the parent node $(X^{(u)}, y^{(u)})$ after the former splitting steps $1, 2, \ldots, u - 1$, that, for any $u = 1, 2, \ldots,$$w_0^{(u)} = \frac{N_0^{(u)}}{N^{(u)}},$$where $N_0^{(u)} = |\{i = 1, 2, \ldots, N^{(u)} : y_i^{(u)} = 0\}|$.

Regardless of the choice for the weight, the optimal parameters $\hat{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)})$ and $\hat{c}^{(u,r)}(\hat{j}^{(u)}, \hat{s}^{(u)})$ are necessarily given by the weighted version of (5) and (6):

$$\hat{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)}) = \frac{\left(1 - w_0^{(u)}\right) \sum_{i: y_i^{(u,l)} \neq 0} y_i^{(u,l)}}{w_0^{(u)} N_0^{(u,l)} + \left(1 - w_0^{(u)}\right) \left(N^{(u,l)} - N_0^{(u,l)}\right)},$$ (10)

$$\hat{c}^{(u,r)}(\hat{j}^{(u)}, \hat{s}^{(u)}) = \frac{\left(1 - w_0^{(u)}\right) \sum_{i: y_i^{(u,r)} \neq 0} y_i^{(u,r)}}{w_0^{(u)} N_0^{(u,r)} + \left(1 - w_0^{(u)}\right) \left(N^{(u,r)} - N_0^{(u,r)}\right)},$$ (11)

where $N_0^{(u,l)} = |\{i = 1, 2, \ldots, N^{(u,l)} : y_i^{(u,l)} = 0\}|$ and $N_0^{(u,r)} = |\{i = 1, 2, \ldots, N^{(u,r)} : y_i^{(u,r)} = 0\}|$. Hence, the optimal parameters $\hat{j}^{(u)}$ and $\hat{s}^{(u)}$ are necessarily given by

$$(\hat{j}^{(u)}, \hat{s}^{(u)}) = \arg \min_{\hat{j}^{(u)}=1,2,\ldots;p_{\hat{s}^{(u)}}} w_0^{(u)} \sum_{i: y_i^{(u,l)} = 0} \hat{c}^{(u,l)} \left(\hat{j}^{(u)}, \hat{s}^{(u)}\right)^2$$

$$+ \left(1 - w_0^{(u)}\right) \sum_{i: y_i^{(u,l)} \neq 0} \left(y_i^{(u,l)} - \hat{c}^{(u,l)} \left(\hat{j}^{(u)}, \hat{s}^{(u)}\right)\right)^2$$

$$+ \hat{c}^{(u,r)} \left(\hat{j}^{(u)}, \hat{s}^{(u)}\right)^2$$

$$+ \left(1 - w_0^{(u)}\right) \sum_{i: y_i^{(u,r)} \neq 0} \left(y_i^{(u,r)} - \hat{c}^{(u,r)} \left(\hat{j}^{(u)}, \hat{s}^{(u)}\right)\right)^2,$$ (12)

10
where $\tilde{c}^{(u,l)}(\tilde{j}^{(u)}, \tilde{s}^{(u)})$ and $\tilde{c}^{(u,r)}(\tilde{j}^{(u)}, \tilde{s}^{(u)})$ are given by (10) and (11).

These optimal parameters in (10), (11), and (12) for the binary splitting step $u$ to minimize the WSSE loss function in (12) intuitively explain why this binary splitting tends to segregate all observations with zero in the response variable from non-zero response observations, when the remaining dataset contains a high percentage of zero response observations, and if the weight $w_0^{(u)}$ is relatively large.

- First, when all zero response observations are classified into either one of the daughter nodes, say “l” without loss of generality, only the last term in (12) is not zero, which is the SSE, of non-zero response observations which are all in the daughter node “r”, being scaled down with the relatively small weight $1 - w_0^{(u)}$. Note that, in this case, $\tilde{c}^{(u,l)}(\tilde{j}^{(u)}, \tilde{s}^{(u)})$ in (10) is zero, while $\tilde{c}^{(u,r)}(\tilde{j}^{(u)}, \tilde{s}^{(u)})$ in (11) is simply the average of non-zero responses. This is the ideal binary splitting.

- Consider then the following small twist from the ideal scenario, that a non-zero, yet closest to zero, response observation is classified into the daughter node “l”, which contained only and all zero response observations, instead of the daughter node “r”. In this case, only the third term in (12) remains zero. In particular, while the second term is moderately increased and the last term is slightly reduced, because of the relatively small weight $1 - w_0^{(u)}$, the first term is tremendously gotten larger, due to the relatively large weight $w_0^{(u)}$ and the big number $N_0^{(u,l)}$ of zero response observations in the daughter node “l”, and since $\tilde{c}^{(u,l)}(\tilde{j}^{(u)}, \tilde{s}^{(u)})$ in (10) is no longer zero, which is impurified by the non-zero response observation. Therefore, the right-hand-side of (12) is most likely not minimized.

- Consider another small twist from the ideal scenario, that a zero response observation is classified into the daughter node “r”, which contained only and all non-zero response observations, instead of the daughter node “l”. In this case, the right-hand-side of (12) is also most likely not minimized, since, while the first and the second terms in (12) remain zero, the last term could be moderately increased or decreased because of the relatively small weight $1 - w_0^{(u)}$, but the third term is remarkably enlarged because of the relatively large weight $w_0^{(u)}$. In this case, $\tilde{c}^{(u,r)}(\tilde{j}^{(u)}, \tilde{s}^{(u)})$ in (11) is reduced from the average of non-zero responses, which is impurified by the zero response observation.

### 3.2 Sum of Squared Canberra Errors

While the above discussions shed light on the novelty of weighing the observations with zero response and non-zero response differently to segregate the imbalanced observations with zero response, the treatment via the WSSE loss function is not necessarily robust among datasets. Indeed, the choice for the weight is too flexible, and if it is not sufficiently large, one of the above two twisted scenarios could still be yielded from the optimal parameters in the binary
splitting that either one of the daughter nodes is impurified. This leads us to further modify the loss function such that the weight depends endogenously on the response and prediction values themselves for each observation. The *Canberra distance* is a perfect fit.

Being introduced in Lance and Williams (1966) and being refined in Lance and Williams (1967) for similarity analyses, the Canberra distance is defined by, for any two real numbers \( p \) and \( q \),

\[
d_{\text{CAN}}(p, q) = \begin{cases} 
0 & \text{if } p = q = 0, \\
\frac{|p-q|}{|p|+|q|} & \text{otherwise.}
\end{cases}
\]

It is essentially the Euclidean distance being normalized by the magnitude of the two real numbers in the denominator. By the triangle inequality, the Canberra distance \( d_{\text{CAN}}(\cdot, \cdot) \) is bounded above by 1, which is attained when either \( p \) or \( q \), but not both, is zero; that is, in this attaining case, \( p \) and \( q \) are the most far apart from each other in terms of the Canberra distance. In this sense, the Canberra distance naturally entails an amplifying for the deviation of zero and non-zero values. As a metric, the Canberra distance is often used for data scattered around an origin, as it is a biased measure around the origin and is very sensitive to values close to zero. For example, we observe \( d_{\text{CAN}}(0, 1) = 1 \) and \( d_{\text{CAN}}(100, 101) \approx 0.005 \), however, under the SSE, these two cases should have the same distance 1. The discrepancy between these two distance can be significant in the context of insurance loss modeling. Therefore, this kind of asymmetric distance can shed light on finding the solution to the imbalanced learning problem in the insurance domain, as we mentioned in Section 1. To be in line with the order of errors in the SSE and the WSSE, which is of squared, also define the *squared* Canberra distance by, for any two real numbers \( p \) and \( q \),

\[
d_{\text{SCAN}}(p, q) = \begin{cases} 
0 & \text{if } p = q = 0, \\
\frac{(p-q)^2}{p^2+q^2} & \text{otherwise.}
\end{cases}
\]

The above properties of the Canberra distance also hold for the squared Canberra distance as long as the two real numbers \( p \) and \( q \) are either both non-negative or both non-positive.

Define the following loss function which is given by the sum of squared Canberra errors (SSCE):

\[
L_C(\mathbf{y}^{(u)}, \mathbf{\hat{y}}^{(u)}) = \sum_{i=1}^{N^{(u)}} d_{\text{SCAN}}(y_i^{(u)}, \mathbf{\hat{y}}_i^{(u)}),
\]

where \( u = 1, 2, \ldots \) is a binary splitting step and \( \mathbf{\hat{y}}^{(u)} \) is given by (4). In this binary splitting, the optimal parameters \( \mathbf{\hat{y}}^{(u)}, \mathbf{\hat{s}}^{(u)}, \mathbf{\hat{c}}^{(u,l)} \left( \mathbf{\hat{j}}^{(u)}, \mathbf{\hat{s}}^{(u)} \right), \) and \( \mathbf{\hat{c}}^{(u,r)} \left( \mathbf{\hat{j}}^{(u)}, \mathbf{\hat{s}}^{(u)} \right) \) are given by minimizing
the SSCE; that is,

\[
\left( \hat{j}^{(u)}, \hat{s}^{(u)}, \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right), \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)
\]

\[
= \arg \min_{j^{(u)}=1,2,\ldots,|P|; s^{(u)} \in X_j^{(u)} \text{ or } s^{(u)} \in P\left( X_j^{(u)} \right)} \sum_{i=1}^{N^{(u,l)}} d_{\text{SCAN}} \left( y_i^{(u,l)}, c^{(u,l)} \left( j^{(u)}, s(u) \right) \right) + \sum_{i=1}^{N^{(u,r)}} d_{\text{SCAN}} \left( y_i^{(u,r)}, c^{(u,r)} \left( j^{(u)}, s(u) \right) \right).
\]

(13)

Although the predicted values \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) and \( \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) cannot be explicitly solved from (13), their existences can be proved. We first have the following lemma.

**Lemma 3.1.** If \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) and \( \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) in (13) exist, then

\[
y_i^{(u,l)} \leq \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \leq y_i^{(N(u,l))},
\]

\[
y_i^{(u,r)} \leq \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \leq y_i^{(N(u,r))},
\]

where \( y_i^{(u,l)} \) and \( y_i^{(u,r)} \) are the smallest response values in the respective daughter nodes, while \( y_i^{(N(u,l))} \) and \( y_i^{(N(u,r))} \) are the largest response values in the respective daughter nodes.

**Proof.** By symmetry, it suffices to show for one of the daughter nodes, say “l” without loss of generality. Assume, to the contrary, that \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) < y_i^{(u,l)} \) or \( y_i^{(N(u,l))} < \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \).

Suppose that \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) < y_i^{(u,l)} \). If \( y_i^{(u,l)} = 0 \), that is impossible as \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \in \mathbb{R}_+ \). If \( y_i^{(u,l)} > 0 \), then \( y_i^{(u,l)} > 0 \), for any \( i = 1, 2, \ldots, N^{(u,l)} \). Moreover, for any \( i = 1, 2, \ldots, N^{(u,l)} \),

\[
\left( y_i^{(u,l)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2 > \left( y_i^{(u,l)} - y_i^{(u,l)} \right)^2 \quad \text{and} \quad \left( y_i^{(u,l)} \right)^2 + \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right)^2 < \left( y_i^{(u,l)} \right)^2 + \left( y_i^{(u,l)} \right)^2.
\]

Hence,

\[
\sum_{i=1}^{N^{(u,l)}} d_{\text{SCAN}} \left( y_i^{(u,l)}, \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right) = \sum_{i=1}^{N^{(u,l)}} \frac{\left( y_i^{(u,l)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2}{\left( y_i^{(u,l)} \right)^2 + \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right)^2}
\]

\[
> \sum_{i=1}^{N^{(u,l)}} \frac{\left( y_i^{(u,l)} - y_i^{(u,l)} \right)^2}{\left( y_i^{(u,l)} \right)^2 + \left( y_i^{(u,l)} \right)^2} = \sum_{i=1}^{N^{(u,l)}} d_{\text{SCAN}} \left( y_i^{(u,l)}, y_i^{(u,l)} \right),
\]

which contradicts that \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) solves (13).
Suppose that \( y^{(u,l)}_{(N(u,l))} < \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)}) \). If \( y^{(u,l)}_{(N(u,l))} = 0 \), then \( y^{(u,l)}_i = 0 \), for any \( i = 1, 2, \ldots, N(u,l) \), and hence

\[
\sum_{i=1}^{N(u,l)} d_{\text{SCAN}}(y^{(u,l)}_i, \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)})) = N(u,l) > 0 = \sum_{i=1}^{N(u,l)} d_{\text{SCAN}}(y^{(u,l)}_i, y^{(u,l)}_{(N(u,l))}),
\]

which contradicts that \( \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)}) \) solves (13). If \( y^{(u,l)}_{(N(u,l))} > 0 \), for any \( q \in \left[ y^{(u,l)}_{(N(u,l))}, \infty \right) \),

\[
\frac{\partial}{\partial q} \left( \sum_{i=1}^{N(u,l)} d_{\text{SCAN}}(y^{(u,l)}_i, q) \right) = \sum_{i=1}^{N(u,l)} \frac{2 \left( q^2 - (y^{(u,l)}_i)^2 \right) y^{(u,l)}_i}{\left( (y^{(u,l)}_i)^2 + q^2 \right)^2} > 0,
\]

and thus \( \sum_{i=1}^{N(u,l)} d_{\text{SCAN}}(y^{(u,l)}_i, \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)})) > \sum_{i=1}^{N(u,l)} d_{\text{SCAN}}(y^{(u,l)}_i, y^{(u,l)}_{(N(u,l))}) \), which, again, contradicts that \( \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)}) \) solves (13).

\[
\square
\]

**Proposition 3.1.** The predicted values \( \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)}) \) and \( \tilde{c}^{(u,r)}(j^{(u)}, s^{(u)}) \) in (13) exist.

*Proof.* Again, by symmetry, it suffices to show for one of the daughter nodes, say “l” without loss of generality. Due to Lemma 3.1, the existence of \( \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)}) \) should be established in

\[
\left[ y^{(u,l)}_{(1)}, y^{(u,l)}_{(N(u,l))} \right].
\]

Suppose that \( y^{(u,l)}_{(1)} = y^{(u,l)}_{(N(u,l))} \). Then, \( y^{(u,l)}_i = y^{(u,l)}_{(1)} = y^{(u,l)}_{(N(u,l))} \), for any \( i = 1, 2, \ldots, N(u,l) \); it is thus clear that \( \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)}) = y^{(u,l)}_{(1)} = y^{(u,l)}_{(N(u,l))} \) such that

\[
\sum_{i=1}^{N(u,l)} d_{\text{SCAN}}(y^{(u,l)}_i, \tilde{c}^{(u,l)}(j^{(u)}, s^{(u)})) = 0.
\]

Suppose that \( y^{(u,l)}_{(1)} < y^{(u,l)}_{(N(u,l))} \). For any \( q \in \left[ y^{(u,l)}_{(1)}, y^{(u,l)}_{(N(u,l))} \right] \),

\[
\frac{\partial}{\partial q} \left( \sum_{i=1}^{N(u,l)} d_{\text{SCAN}}(y^{(u,l)}_i, q) \right) = \sum_{i:y^{(u,l)}_i \neq 0} \frac{2 \left( q^2 - (y^{(u,l)}_i)^2 \right) y^{(u,l)}_i}{\left( (y^{(u,l)}_i)^2 + q^2 \right)^2},
\]

which is a continuous function in \( q \in \left[ y^{(u,l)}_{(1)}, y^{(u,l)}_{(N(u,l))} \right] \). Moreover,
\[
\frac{\partial}{\partial q} \left( \sum_{i=1}^{N(u,l)} d_{\text{SCAN}} \left( y_i^{(u,l)}, q \right) \right) \bigg|_{q=y_i^{(u,l)}} = \sum_{i:y_i^{(u,l)} \neq 0} 2 \left( \left( y_i^{(u,l)} \right)^2 - \left( y_i^{(u,l)} \right)^2 \right) y_i^{(u,l)} < 0,
\]

\[
\frac{\partial}{\partial q} \left( \sum_{i=1}^{N(u,l)} d_{\text{SCAN}} \left( y_i^{(u,l)}, q \right) \right) \bigg|_{q=y_i^{(N(u,l))}} = \sum_{i:y_i^{(N(u,l))} \neq 0} 2 \left( \left( y_i^{(u,l)} \right)^2 - \left( y_i^{(u,l)} \right)^2 \right) y_i^{(u,l)} > 0.
\]

Therefore, by the Intermediate Value Theorem, there exists an \( \widehat{c}^{(u,l)} \left( \widehat{j}^{(u)}, \widehat{s}^{(u)} \right) \in \left( \left( y_i^{(u,l)} \right)^{(1)}, y_i^{(N(u,l))} \right) \) such that

\[
\frac{\partial}{\partial q} \left( \sum_{i=1}^{N(u,l)} d_{\text{SCAN}} \left( y_i^{(u,l)}, \widehat{c}^{(u,l)} \left( \widehat{j}^{(u)}, \widehat{s}^{(u)} \right) \right) \right) = 0,
\]

\[
\frac{\partial}{\partial q} \left( \sum_{i=1}^{N(u,l)} d_{\text{SCAN}} \left( y_i^{(u,l)}, \widehat{c}^{(u,l)} \left( \widehat{j}^{(u)}, \widehat{s}^{(u)} \right) + \right) \right) > 0,
\]

and so solves (13).

\[ \Box \]

### 3.3 Practical Implementation

From now on, we shall call the tree-based models, by the WSSE loss function as the WSSE tree model, and by the SSCE loss function as the Canberra tree model. This section discusses the practical implementation aspect for the WSSE tree and Canberra tree models; in particular, to modify the classical CART algorithms with the two proposed loss functions, we employ the \texttt{rpart} function from R package \texttt{rpart}, see Therneau and Atkinson (1997). The package provides a user splits option, see Therneau (2019), which allows an easier way to extend \texttt{rpart} and validate new methodologies than additions to the C base.

#### 3.3.1 WSSE Tree Model

To construct the WSSE tree model, one needs to search for the optimal parameters \( \widehat{j}^{(u)} \) and \( \widehat{s}^{(u)} \) in (12) for each binary splitting step \( u = 1, 2, \ldots \). This is computationally expensive by (12), as one shall need to, for each possible split, identify whether each observation has a zero or non-zero response in both daughter nodes.

However, first, note that

\[
\sum_{i:y_i^{(u,l)} \neq 0} \left( y_i^{(u,l)} - \widehat{c}^{(u,l)} \left( \widehat{j}^{(u)}, \widehat{s}^{(u)} \right) \right)^2 = \sum_{i=1}^{N(u)} \left( y_i^{(u)} - \widehat{c}^{(u,l)} \left( \widehat{j}^{(u)}, \widehat{s}^{(u)} \right) \right)^2 \mathbb{1} \{ y_i^{(u)} \neq 0, X_i^{(u)} \leq \widehat{s}^{(u)} \}.
\]
where the right-hand-side contains a slight abuse of notation that, if \( X_{ij}^{(u)} \) is categorical, \( X_{ij}^{(u)} \leq \hat{s}^{(u)} \) in fact means that \( X_{ij}^{(u)} \in \hat{s}^{(u)} \), as long as the response variable among the observations in the same categorical level are ordered among the levels.

As discussed in Section 2.2, in \texttt{rpart}, before any potential split, the observations are ordered for a continuous explanatory variable. For a categorical explanatory variable, the observations are reshuffled such that observations with the same level are grouped together, and the response variable among the observations with the same level is ordered among the levels. Let \( \tilde{s} = 1, 2, \ldots, N(u) - 1 \) be the ordered position for the explanatory variable \( X_{ij}^{(u)} \) of the binary split, then

\[
\sum_{i: y_i^{(u,l)} \neq 0} \left( y_i^{(u,l)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2 = \sum_{i=1}^{N(u)} \left( y_i^{(u)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2 \mathbb{1}_{\left\{ y_i^{(u)} \neq 0, i \leq \tilde{s} \right\}}.
\]

Define, for \( i = 1, 2, \ldots, N(u) \) and \( \tilde{s} = 1, 2, \ldots, N(u) - 1 \), \( d_{is}^{(u,l)} = \mathbb{1}_{\left\{ y_i^{(u)} \neq 0, i \leq \tilde{s} \right\}} \). Therefore,

\[
\sum_{i: y_i^{(u,l)} \neq 0} \left( y_i^{(u,l)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2 = \sum_{i=1}^{N(u)} \left( y_i^{(u)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right)^2 d_{is}^{(u,l)}
\]

\[
= \left\| \left( y^{(u)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right) \cdot d^{(u,l)}_{s} \right\|_2^2,
\]

where \( \cdot \) represents the dot product herein, \( \| \cdot \|_2 \) is the \( l^2 \)-norm, and \( d_{s}^{(u,l)} = \left( d_{1s}^{(u,l)}, d_{2s}^{(u,l)}, \ldots, d_{N(u)s}^{(u,l)} \right)^T \), for \( \tilde{s} = 1, 2, \ldots, N(u) - 1 \).

Similarly, we have

\[
\sum_{i: y_i^{(u,r)} \neq 0} \left( y_i^{(u,r)} - \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}(u) \right) \right)^2 = \left\| \left( y^{(u)} - \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}(u) \right) \right) \cdot e_{s}^{(u,r)} \right\|_2^2,
\]

\[
\sum_{i: y_i^{(u,l)} = 0} \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}(u) \right)^2 = \left\| \left( y^{(u)} - \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}(u) \right) \right) \cdot e_{s}^{(u,l)} \right\|_2^2,
\]

\[
\sum_{i: y_i^{(u,r)} = 0} \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}(u) \right)^2 = \left\| \left( y^{(u)} - \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}(u) \right) \right) \cdot e_{s}^{(u,r)} \right\|_2^2,
\]

where, for \( i = 1, 2, \ldots, N(u) \) and \( \tilde{s} = 1, 2, \ldots, N(u) - 1 \), \( d_{is}^{(u,l)} = \mathbb{1}_{\left\{ y_i^{(u)} \neq 0, i \leq \tilde{s} \right\}} \), \( e_{is}^{(u,l)} = \mathbb{1}_{\left\{ y_i^{(u)} = 0, i \leq \tilde{s} \right\}} \).
\( e_{i\tilde{s}}^{(u,r)} = 1 \{ y_{i}^{(u)} = y_{i,0}^{(u)} > \tilde{s} \} \). Hence, (12) can be rewritten as

\[
\left( \tilde{\gamma}(u), \tilde{s}(u) \right) = \arg \min_{\gamma(u) = 1,2, \ldots, p; \ s(u) \in X_{j(u)}, \ s(u) \in P \left( X_{j(u)} \right)} w_{0}^{(u)} \left\| \left( y^{(u)} - \tilde{c}(u,l) \left( \tilde{\gamma}(u), \tilde{s}(u) \right) \right) \cdot e_{\tilde{s}}^{(u,l)} \right\|_{2}^{2} + \left( 1 - w_{0}^{(u)} \right) \left\| \left( y^{(u)} - \tilde{c}(u,l) \left( \tilde{\gamma}(u), \tilde{s}(u) \right) \right) \cdot d_{\tilde{s}}^{(u,l)} \right\|_{2}^{2} (14)
\]

\[
\left( 1 - w_{0}^{(u)} \right) \left\| \left( y^{(u)} - \tilde{c}(u,r) \left( \tilde{\gamma}(u), \tilde{s}(u) \right) \right) \cdot e_{\tilde{s}}^{(u,r)} \right\|_{2}^{2} + \left( 1 - w_{0}^{(u)} \right) \left\| \left( y^{(u)} - \tilde{c}(u,r) \left( \tilde{\gamma}(u), \tilde{s}(u) \right) \right) \cdot d_{\tilde{s}}^{(u,r)} \right\|_{2}^{2},
\]

in which the vectors \( d_{\tilde{s}}^{(u,l)}, d_{\tilde{s}}^{(u,r)}, e_{\tilde{s}}^{(u,l)}, e_{\tilde{s}}^{(u,r)} \), for \( \tilde{s} = 1, 2, \ldots, N^{(u)} - 1 \), can be obtained easily without going through the ordered positions of the binary split one-by-one, and thus are computationally less expensive.

Indeed, from the parent node, we have the vector \( d^{(u)} = \left( \mathbb{1}_{\{ y_{1}^{(u)} \neq 0 \}}, \mathbb{1}_{\{ y_{2}^{(u)} \neq 0 \}}, \ldots, \mathbb{1}_{\{ y_{N^{(u)}} \neq 0 \}} \right)^{T} \); we then construct the following auxiliary matrix \( D^{(u)} = \begin{bmatrix} d^{(u)} & d^{(u)} & \ldots & d^{(u)} \end{bmatrix} \), which is of size \( N^{(u)} \times (N^{(u)} - 1) \); finally, define \( D^{(u,l)} \) and \( D^{(u,r)} \) as the upper and lower trapezoidal versions of \( D^{(u)} \) respectively, which are in fact \( D^{(u,l)} = \begin{bmatrix} d_{1}^{(u,l)} & d_{2}^{(u,l)} & \ldots & d_{N^{(u)} - 1}^{(u,l)} \end{bmatrix} \) and \( D^{(u,r)} = \begin{bmatrix} d_{1}^{(u,r)} & d_{2}^{(u,r)} & \ldots & d_{N^{(u)} - 1}^{(u,r)} \end{bmatrix} \). For example, in the motivating example of Section 2.2, if \( \tilde{j} = 1 \), then

\[
D = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix},
\]
For obtaining the vectors \( e^{(u,l)}_s \) and \( e^{(u,r)}_s \), for \( s = 1, 2, \ldots, N(u) - 1 \), simply begin with \( e^{(u)} = \left( \mathbb{1}_{y_1^{(u)} = 0}, \mathbb{1}_{y_2^{(u)} = 0}, \ldots, \mathbb{1}_{y_{N(u)}^{(u)} = 0} \right) \) and the remaining steps follow the same.

Making use of (14), which is more computationally efficient, Algorithm 1 summarizes the details of implementing the WSSE tree model based on the \texttt{R} package \texttt{rpart}. Note that we only highlight key distinctions in splitting function comparing with the classical ANOVA best split. The complete CART algorithm with parameter tuning via cross validation is detailed in Algorithm 3.

### 3.3.2 Canberra Tree Model

Unlike the WSSE tree model implementation, in which the cost of computation arises from the identification of a zero or non-zero response for an observation in (12), which has been addressed in the last section via (14), a computational burden for the implementation of the Canberra tree model lies in the predicted values \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) and \( \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) in (13). Although their existences have been shown in Proposition 3.1, they are still implicit, and thus are solved via numerical optimization.

To reduce such computational cost, we make use of the following proposition, which states that if the number of observations with zero response in a daughter node is more than half of the number of observations in the node, then the predicted value has to be 0, and thus there is no need to solve it by numerical optimization in this case. Algorithm 2 summarizes the details of implementing the Canberra tree model based on the \texttt{R} package \texttt{rpart}.

**Proposition 3.2.** If \( N_0^{(u,l)} > \frac{1}{2} N^{(u,l)} \) (resp. \( N_0^{(u,r)} > \frac{1}{2} N^{(u,r)} \)), then \( \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \) (resp. \( \hat{c}^{(u,r)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \)) is unique and must be 0; moreover, \( \sum_{i=1}^{N^{(u,l)}} d_{SCAN} \left( y_i^{(u,l)}, \hat{c}^{(u,l)} \left( \hat{j}^{(u)}, \hat{s}^{(u)} \right) \right) = \)
Recall that, by Lemma 3.1,

\[
\text{Proof.}
\]

Compare the goodness of all the splits and choose the pair

Calculate the goodness of the each possible split,

Get the non-zero and zero indicator matrices at the daughter nodes

Get the non-zero and zero indicator matrices at the daughter nodes

Calculate the weight

Get the predicted value and the WSSE at the parent node from step \( u - 1 \);

for \( j^{(u)} = 1, 2, \ldots, p \) do

Calculate the sum of WSSE at the left node and the right node from the objective function of (14);

end

Calculate the goodness of the each possible split, \( G ( j^{(u)}, s^{(u)} ) \) from (9);

Compare the goodness of all the splits and choose the pair \( ( \hat{j}^{(u)}, \hat{s}^{(u)} ) \) with the largest goodness;

return \( ( \hat{j}^{(u)}, \hat{s}^{(u)} ) \)

\[
N^{(u,l)} - N_0^{(u,l)} \quad \text{(resp. } \sum_{i=1}^{N^{(u,r)}} d_{\text{SCAN}} ( y_i^{(u,r)}), \hat{c}^{(u,r)} ( \hat{j}^{(u)}, \hat{s}^{(u)} ) ) = N^{(u,r)} - N_0^{(u,r)} \).
\]

**Proof.** Without loss of generality, we, again, only prove the result for the “l” daughter node. Recall that, by Lemma 3.1, \( \hat{c}^{(u,l)} ( \hat{j}^{(u)}, \hat{s}^{(u)} ) \in \left[ y^{(u,l)} (1), y^{(u,l)} ( N^{(u,l)} ) \right] \). Note also that, since \( N_0^{(u,l)} > \frac{1}{2} N^{(u,l)} > 0 \), \( y^{(u,l)} (1) = 0 \).

Suppose that \( y^{(u,l)} (1) = y^{(u,l)} ( N^{(u,l)} ) \). Then, \( y^{(u,l)} (1) = y^{(u,l)} ( N^{(u,l)} ) = 0 \), and thus \( \hat{c}^{(u,l)} ( \hat{j}^{(u)}, \hat{s}^{(u)} ) = 0 \).

Obviously, \( \sum_{i=1}^{N^{(u,l)}} d_{\text{SCAN}} ( y_i^{(u,l)}, \hat{c}^{(u,l)} ( \hat{j}^{(u)}, \hat{s}^{(u)} ) ) = 0 = N^{(u,l)} - N_0^{(u,l)} \).

Suppose that \( y^{(u,l)} (1) < y^{(u,l)} ( N^{(u,l)} ) \). Then, for any \( q \in \left[ 0, y^{(u,l)} ( N^{(u,l)} ) \right] \),

\[
\sum_{i=1}^{N^{(u,l)}} d_{\text{SCAN}} ( y_i^{(u,l)}, q ) = \sum_{i:y_i^{(u,l)} = 0} d_{\text{SCAN}} ( y_i^{(u,l)}, q ) + \sum_{i:y_i^{(u,l)} \neq 0} d_{\text{SCAN}} ( y_i^{(u,l)}, q )
\]
\[
\begin{aligned}
  &N^{(u,l)} - N_0^{(u,l)} \\
  = &\begin{cases}
    N^{(u,l)} - N_0^{(u,l)} & \text{if } q = 0, \\
    N_0^{(u,l)} + \sum_{i:y_i^{(u,l)} \neq 0} \frac{(y_i^{(u,l)} - q)^2}{(y_i^{(u,l)})^2 + q^2} & \text{otherwise}.
  \end{cases}
\end{aligned}
\]

Since \(N_0^{(u,l)} > \frac{1}{2} N^{(u,l)}\), we have \(N^{(u,l)} - N_0^{(u,l)} < N_0^{(u,l)} \leq N_0^{(u,l)} + \sum_{i:y_i^{(u,l)} \neq 0} \frac{(y_i^{(u,l)} - q)^2}{(y_i^{(u,l)})^2 + q^2}\), and thus \(\bar{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)})\) is unique and must be 0, and hence \(\sum_{i=1}^{N^{(u,l)}} d_{\text{SCAN}} \left( y_i^{(u,l)}, \bar{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)}) \right) = N^{(u,l)} - N_0^{(u,l)}\).

\[\square\]

**Algorithm 2:** Implementation of the Canberra tree model using \texttt{rpart}: Splitting procedure at step \(u\)

\begin{algorithm}
\textbf{Input} : The remaining dataset \((X^{(u)}, y^{(u)})\) with \(N^{(u)}\) observations
\textbf{Output}: The optimal split \((\hat{j}^{(u)}, \hat{s}^{(u)}, \bar{c}^{(u,l)}(\hat{j}^{(u)}, \hat{s}^{(u)}), \bar{c}^{(u,r)}(\hat{j}^{(u)}, \hat{s}^{(u)}))\)
\begin{enumerate}
  \item Get the predicted value and the SSCE at the parent node from step \(u - 1\);
  \item for \(j^{(u)} = 1, 2, \ldots, p\) do
    \item for \(s^{(u)} \in X_j^{(u)}\) (resp. \(s^{(u)} \in P(X_j^{(u)})\)) do
      \item if \(N_0^{(u,l)} > \frac{1}{2} N^{(u,l)}\) then
        \begin{itemize}
          \item Assign a predicted value of 0 to all observations in the “l” daughter node;
          \item Assign the number of observations with non-zero response, \(N^{(u,l)} - N_0^{(u,l)}\),
                to the SSCE to the “l” daughter node;
        \end{itemize}
      \item else
        \begin{itemize}
          \item Obtain the SSCE and the corresponding predicted value \(\bar{c}^{(u,l)}(j^{(u)}, s^{(u)})\)
                by utilizing the \texttt{optimize} function;
        \end{itemize}
    \item end
  \item if \(N_0^{(u,r)} > \frac{1}{2} N^{(u,r)}\) then
    \begin{itemize}
      \item Assign a predicted value of 0 to all observations in the “r” daughter node;
      \item Assign the number of observations with non-zero response, \(N^{(u,r)} - N_0^{(u,r)}\),
            to the SSCE to the “r” daughter node;
    \end{itemize}
  \item else
    \begin{itemize}
      \item Obtain the SSCE and the corresponding predicted value \(\bar{c}^{(u,r)}(j^{(u)}, s^{(u)})\)
            by utilizing the \texttt{optimize} function;
    \end{itemize}
  \item end
  \item Calculate the goodness of the split, \(G(j^{(u)}, s^{(u)})\) from (9);
\end{enumerate}
\end{algorithm}
3.4 Motivating Example Revisited

Armed with the basics and the practical consideration for the WSSE tree and Canberra tree models, let us revisit the motivating example in Section 2.2, in which the fabricated dataset (see Table 1) contains 70% of the observations with zero response. In particular, we shall compare the splitting performance among the classical ANOVA best split, the WSSE tree model, and the Canberra tree model.

Figure 2(a) and Figure 2(b) describe the goodness of split by the three tree-based models via the continuous explanatory variable $X_1$ and the categorical variable $X_2$ respectively. It is evident that the optimal split by the Canberra tree model differs from those by the other two. If $\hat{j} = 1$, using the Canberra tree model, $\hat{s} = 7$ which is the ideal splitting; but, using the ANOVA best split or the WSSE tree model, $\hat{s} = 8$. If $\hat{j} = 2$, using the Canberra tree model, $\hat{s} = \{A, B\}$ which is the ideal splitting again; but, using the ANOVA best split or the WSSE tree model, $\hat{s} = \{A, B, C\}$.

Two additional features can be observed from Figure 2. First, using the Canberra tree model, there is not any SSCE improvement with respect to the parent node, when $s \leq 4$ for $j = 1$, or when $s = \{A\}$ for $j = 2$; but, there is at least some SSE or WSSE improvement under these cases. These are because the SSCE is the most sensitive to observations with zero response impurifying the daughter node containing all observations with non-zero response. Second, although the optimal splits by the ANOVA best split and the WSSE tree model are the same, the WSSE tree model demonstrates a greater preference, at $s = 7$ for $j = 1$, and at $s = \{A, B\}$ for $j = 2$, than the ANOVA best split; moreover, by examining the slope, the goodness of split increment of the WSSE tree model, from $s = 6$ to $s = 7$ for $j = 1$, and from $s = \{A\}$ to $s = \{A, B\}$ for $j = 2$, is remarkably larger than that of the ANOVA best split.

To ensure that the proposed WSSE tree and Canberra tree models can be applied not only to the fabricated dataset in the motivating example, we shall also enrich the results in two more analyses. First, in Appendix B, another experimental dataset, in which 95% of the observations...
output zero response and the values of the non-zero responses range from 1 to 1000, and which is more in tune with real-life insurance claims, is studied via the three tree-based models; same conclusions are drawn as in the motivating example. Second, the overall predictive performance of the three tree-based models in a simulation study is provided, which do not only investigate the first splitting step as in the motivating and experimental examples; this shall be discussed in the following section.

4 Simulation Study

In this section, we evaluate the splitting performance of the proposed WSSE tree and Canberra tree models via a simulation experiment. Under the dataset with a high percentage of observations with zero response, they are compared with the ANOVA best split. We observe the improved prediction performance of the tree by accommodating the imbalanced learning problem. The synthetic datasets are designed to imitate real-life insurance datasets, which contain a mixture of zeros and non-negative insurance claims along with the correlated explanatory variables.

4.1 Design of Synthetic Dataset

In order to mimic the real-life insurance datasets, we create the synthetic training and test datasets that consist of both continuous and categorical explanatory variables \( \mathbf{X} = (\mathbf{X}_{\text{categorical}}, \mathbf{X}_{\text{continuous}}) \), and a response variable following the Tweedie distribution. The continuous explanatory variables are simulated by using the \texttt{mvrnorm} function in the \texttt{R} package \texttt{MASS}. They are sampled from a \( p \)-dimensional multivariate normal distribution with mean 0 and the variance-covariance matrix \( \Sigma \); that is,

\[
\mathbf{X}_{\text{continuous}} \sim \mathcal{N}_p(0, \Sigma),
\]

where the covariance between continuous variable \( \Sigma_{ij} = \text{Cov}(\mathbf{X}_i, \mathbf{X}_j) = (0.8)^{|i-j|} \). In other words, continuous explanatory variables that are located closer to one another have a more significant correlation. Each categorical explanatory variable of \( \mathbf{X}_{\text{categorical}} \) is generated by random sampling from the set of integers \((-3, -2, 1, 4)\), with respective probabilities of \((0.1, 0.2, 0.2, 0.5)\). Note that there is no correlation between these categorical variables.

Each of our synthetic training and test datasets contains 100 observations\(^1\), with 53% of the observations in the training dataset and 63% in the test dataset have a zero response. For both the training and test datasets, we consider 10 explanatory variables, including 5 continuous explanatory variables and 5 categorical variables respectively. In real-life datasets, not all explanatory variables are related to the response variable. Therefore, assigning a different size of linear coefficients is assumed to generate explanatory variables with a strong signal, explanatory variables with a weak signal, and noisy non-sense explanatory variables. Specifically, the

\(^1\)We also study on another larger synthetic dataset, with details being relegated to Appendix C.
synthetic dataset consists of 2 continuous explanatory variables and 2 categorical explanatory variables with relatively larger coefficient, 2 continuous explanatory variables and 2 categorical explanatory variables with relatively smaller coefficient, 1 continuous explanatory variable, and 1 categorical explanatory variable with coefficient 0. Below is the vector of regression coefficients $\beta$ used:

$$
\beta = (-0.1, 1.0, 1.0, 0.5, 0.5, 0.0, 1.0, 1.0, 0.5, 0.5, 0.0)^T.
$$

The response variable $Y = (y_1, y_2, \ldots, y_i, \ldots, y_{100})$ is generated from a Tweedie GLM framework, with a log link function using the `rtweedie` function in the R package `tweedie`. Therefore, the response variable $y_i$ of the $i$th observation should follow a Tweedie distribution,

$$
y_i \sim \text{Tweedie}(\mu_i, \phi, \xi),
$$

where $\mu_i = E(y_i)$ is the expectation of the $i$th response with the log link function $g(\mu_i) = \log(\mu_i) = X_i\beta$, the dispersion parameter $\phi = 2$, and the variance power parameter $\xi = 1.7$. The variance of the response variable $\text{Var}(Y) = \phi\mu^\xi$.

### 4.2 Splitting and Predictive Performances

We fit the ANOVA, WSSE, and Canberra tree models to the synthetic training dataset. These tree models are computed using `rpart` as demonstrated in Section 3.3 and Appendix A. The hyperparameter tuning on minsplit $N$ (from 2 to 5), maxdepth (from 5 to 15), and cost-complexity $\alpha$ ($1 \times 10^{-4}$) are performed for these tree models with 10-fold cross-validation. The structures of these obtained tree models are shown in Figure 3, Figure 4, and Figure 5, respectively. In these plotted trees, we list the predicted value, the number of observations, and the percentage of observations in each node. In addition, at the bottom of each tree, the percentage of zeros on each terminal node is presented.

We first note that the overall structures of the WSSE tree and Canberra tree models are quite different from that of the ANOVA tree model. Both the WSSE tree and Canberra tree models reveal apparent structural changes at node 1 (the root node), which is the most critical split for tree-based models. In the Canberra tree model, the first split is initiated by the variable $V_6$, whereas in the ANOVA tree model, the root nodes are split by the variable $V_8$, and in the WSSE tree model, the root nodes are split by the variable $V_5$. Recall the regression coefficients shown in equation (15); $V_6$ is strongly correlated with the response variable, while $V_5$ and $V_8$ are noisy variable or weakly correlated with the response variable. Hence, the Canberra tree model is more effective in finding the correct explanatory variable to split under the imbalanced problem presented in the dataset. It is also noteworthy that the first few splits in the ANOVA tree model are determined mainly by the continuous variables, while categorical variables, such as $V_1$ and $V_2$, are taken into account in the WSSE tree and Canberra tree models. For instance, the node 2 in the WSSE tree model is divided by the categorical variable $V_2$, and the node 3 in the Canberra tree model is divided by the categorical variable $V_1$. This phenomenon illustrates the
Figure 3: Fitted ANOVA tree model on synthetic data

Type of explanatory variable, more specifically the number of splitting points, which can affect choosing splitting variable in the tree-based algorithms. This issue, bias in variable selection, is exaggerated in the imbalanced dataset. In addition, as we can see from the percentages of zeros on the terminal nodes, the Canberra tree model is more likely to have 0% or 100% zero claims than the other two tree models, which indicates the Canberra tree model outperforms the other two models at separating the zero and non-zero claims.

Several prediction accuracy measures are considered in this simulation study to allow a fair comparison among different models. The measures utilized include Gini index, coefficient of determination R², concordance correlation coefficient (CCC), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), mean percentage error (MPE), and mean Tweedie deviance (TweedieDev). The definitions and interpretations of these validation measures are detailed in Table 2.

In order to make for an easier comparison of various prediction accuracy measures among different models, we adopt the heatmap for validation introduced in Quan and Valdez (2018). Figure 6 and Figure 7 provide heatmaps comparing the performance of the models based on the training and test dataset respectively. The heatmaps have been organized by rescaling and color-coding all the validation measures. Dark blue indicates the best model, while dark red classifies the worst model. For any color in between, the performance of the corresponding model is determined by how close the color is to dark blue and dark red. Note that the numbers displayed on the heatmap are the original values for validation measures instead of the rescaled
Figure 4: Fitted WSSE tree model on synthetic data

From Figure 6 and Figure 7, we can observe that the Canberra tree model consistently has the best overall performance on both the training dataset and the test dataset. It is worth mentioning that the Canberra tree model has much better performance in MAPE and MPE compared to the two other tree models. Referring to Table 2, both MAPE and MPE include the observed response as a denominator in their calculations.

In this analysis, the MAPE and MPE are calculated regardless of the observations with zero response, which implies the resulting MAPE and MPE only reveal the prediction performance based on the observations with non-zero response. The small values of MAPE and MPE indicate that the Canberra tree model has a good fit for the observations with the non-zero response. Hence, the Canberra tree model is not only able to separate the zero and non-zero claims but also good at fitting positive claims.

Figure 8 presents the density plots of the predicted and observed values of the response variable. Note that a log transformation has been performed on the response values to make the plots more informative. As we can observe from Figure 8, the Canberra tree model outperforms ones.
the other two models on the prediction of the zero responses. It is worth mentioning that response values predicted by the ANOVA tree model are centered on a relatively small positive value; on the other hand, the Canberra tree model is able to identify zero claims precisely. This result is consistent with our discussion in Section 2.2 that the zeros are grouped together with small positive values under the SSE loss function. Overall, the Canberra tree model provides the best prediction performance on the synthetic dataset.

4.3 Robustness of Simulation Study

In Section 4.2, we show the splitting and predictive performances on one synthetic dataset, which is generated based on a random seed. To ensure their robustness, we re-examine these performances to each of 100 synthetic datasets, which are still based on the same design in Section 4.1, but are generated by different random seeds. In particular, for each of these datasets, under each validation measure, we record which model, among the ANOVA, WSSE, and Canberra tree models, performs the best. The radar plot in Figure 9 summarizes these records by
<table>
<thead>
<tr>
<th>Validation measure</th>
<th>Description</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gini Index</td>
<td>[ Gini = 1 - \frac{2}{N-1} \left( N - \frac{\sum_{i=1}^{N} \tilde{y}<em>i}{\sum</em>{i=1}^{N} y_i} \right) ]</td>
<td>Higher Gini is better.</td>
</tr>
<tr>
<td></td>
<td>where ( \tilde{y} ) is the corresponding to ( y ) after ranking the corresponding predicted values ( \tilde{y} ).</td>
<td></td>
</tr>
<tr>
<td>Coefficient of Determination</td>
<td>[ R^2 = 1 - \frac{\sum_{i=1}^{N} (\tilde{y}<em>i - y_i)^2}{\sum</em>{i=1}^{N} \left( y_i - \frac{1}{N} \sum_{i=1}^{N} y_i \right)^2} ]</td>
<td>Higher ( R^2 ) is better.</td>
</tr>
<tr>
<td></td>
<td>where ( \tilde{y} ) is predicted values.</td>
<td></td>
</tr>
<tr>
<td>Concordance Correlation</td>
<td>[ CCC = \frac{2\sigma_\tilde{y} \sigma_y}{\sigma_\tilde{y}^2 + \sigma_y^2 + (\mu_{\tilde{y}} - \mu_y)^2} ]</td>
<td>Higher CCC is better.</td>
</tr>
<tr>
<td></td>
<td>where ( \mu_{\tilde{y}} ) and ( \mu_y ) are the means</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \sigma_\tilde{y}^2 ) and ( \sigma_y^2 ) are the variances</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( \rho ) is the correlation coefficient.</td>
<td></td>
</tr>
<tr>
<td>Root Mean Squared Error</td>
<td>[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\tilde{y}_i - y_i)^2} ]</td>
<td>Lower ( RMSE ) is better.</td>
</tr>
<tr>
<td>Mean Absolute Error</td>
<td>[ MAE = \frac{1}{N} \sum_{i=1}^{N}</td>
<td>\tilde{y}_i - y_i</td>
</tr>
<tr>
<td>Mean Absolute Percentage Error</td>
<td>[ MAPE = \frac{1}{N} \sum_{i=1}^{N} \left</td>
<td>\frac{\tilde{y}_i - y_i}{y_i} \right</td>
</tr>
<tr>
<td>Mean Percentage Error</td>
<td>[ MPE = \frac{1}{N} \sum_{i=1}^{N} \frac{\tilde{y}_i - y_i}{y_i} ]</td>
<td>Lower (</td>
</tr>
<tr>
<td>Mean Tweedie Deviance</td>
<td>[ TweedieDev = \frac{1}{N} \sum_{i=1}^{N} 2 \left( \frac{\max(y,0)^{\frac{1}{\xi}}}{{(1-\xi)}^2} - \frac{y^{\frac{1}{\xi}}}{{(1-\xi)}^2} + \frac{\mu_{\tilde{y}}^{\xi}}{\xi} + \frac{\mu_y^{\xi}}{\xi} \right) ]</td>
<td>Lower TweedieDev is better.</td>
</tr>
</tbody>
</table>

Table 2: List of validation measures

displaying the number of datasets in which each model performs the best under each measure. First, the WSSE tree model is slightly superior to the ANOVA tree model, in terms of the number of synthetic datasets that a model performs the best on, under all validation measures. Second, except under the Gini index being exceeded by the ANOVA and WSSE tree models, the Canberra tree model substantially outmatches these two tree models under all measures.

5 Concluding Remarks and Future Directions

In an effort to tailor the regression tree model to insurance datasets, we propose two modified loss functions, which are the weighted sum of squared error and the sum of squared Canberra error loss functions, as the node impurity function under the CART framework. An experiment on a simplified example has demonstrated that the WSSE tree and Canberra tree models are more effective at separating zero response from non-zero response observations than the default ANOVA method. We examine the prediction performance of our proposed models using a synthetic dataset that contains a high percentage of zeros. The result suggests that the Canberra tree model offers the best overall prediction performance, especially when it comes to the
Further modeling work needs to be conducted to validate the performance of the proposed models on real-world datasets. In addition, there are a few avenues for future research. As discussed in Section 1, tree-based ensemble methods can be utilized when the model prediction performance is the core objective. We can offer the WSSE tree and Canberra tree models as
base learners for ensemble methods. Also, extending our findings to a claims frequency with a tailored loss would be a valuable contribution to the literature.

**Acknowledgment**

The authors are grateful to anonymous reviewers for their careful reading and insightful comments. Zhiyu Quan would like to thank the Campus Research Board for the funding support of this research project through the Arnold O. Beckman Research Award.
Figure 9: Radar plot of number of synthetic datasets performing the best by each tree model under each validation measure.
References


Appendix A  CART Algorithm and Pruning Procedures

Let $T$ be a subtree of the fully grown tree $T_0$, which is based on the recursive binary splittings in Section 2.1. Denote $|T|$ as the number of terminal nodes in the subtree, which is also the number of disjoint regions $M$ of the explanatory variables’ space $X$; that is, $M = |T|$. Recall that, by (1), (2), and (3), the total SSE of the subtree $T$ is given by

$$
\sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \sum_{m=1}^{|T|} \sum_{i : \mathbf{x}_i \in \hat{R}_m} (y_i - \hat{c}_m)^2 = \sum_{m=1}^{|T|} L_m(T),
$$

where $L_m(T) = \sum_{i : \mathbf{x}_i \in \hat{R}_m} (y_i - \hat{c}_m)^2$ is the SSE of the subtree $T$ for the region $\hat{R}_m$.

The pruning criterion in the CART algorithm is defined by adding a weighted cost term to the total SSE of a subtree $T$:

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

where the tuning parameter $\alpha \geq 0$ measures the degree of penalization to a subtree $T$ for its number of terminal nodes. The CART algorithm then, for each $\alpha \geq 0$, finds an optimal subtree $T_\alpha$ among all subtrees $T$ to minimize the criterion $C_\alpha(\cdot)$, and thus prunes the fully grown tree $T_0$ to be the subtree $T_\alpha$. Obviously, when $\alpha = 0$, $T_0$ is the fully grown tree, which also explains its notation. A large $\alpha$ leads to a small subtree $T_\alpha$, and vice versa. Therefore, the tuning parameter $\alpha$ balances the trade-off between the cost, in terms of the number of terminal nodes $|T_\alpha|$, and the complexity of the tree-model, and hence the name of cost-complexity pruning criterion for $C_\alpha(\cdot)$. Finally, the CART algorithm estimates the tuning parameter $\alpha$ by a $K$-fold cross-validation; see Algorithm 3 below for details.

**Algorithm 3:** R package `rpart` of CART

**Input:** Training dataset $(\mathbf{X}, \mathbf{y})$, parameter $K$ for $K$-fold cross-validation, msplit $N$

**Output:** Optimal subtree $T_\alpha$

1. Grow a full tree $T_0$ on a training dataset using recursive binary splittings as in Section 2.1;
2. Prune the full tree $T_0$ to subtrees $T_\alpha$ using cost-complexity pruning criterion $C_\alpha(\cdot)$;
3. Divide the training dataset into $K$ folds to determine the optimal tuning parameter $\alpha$;
4. for $k = 1, 2, \ldots, K$ do
   5. Repeat steps 1 and 2 on all folds except the $k$-th one;
   6. Compute the mean squared prediction error on the hold out $k$-th fold using $T_\alpha$;
7. end
8. Average the results for each value of $\alpha$ and pick the $\alpha$ which minimizes the average prediction error;
9. return $T_\alpha$
Appendix B  Experimental Example

To be more in line with the real-life insurance claims, we extend the motivating example to a practical example dataset; see Table 3.

<table>
<thead>
<tr>
<th>Obs</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>46</th>
<th>47</th>
<th>48</th>
<th>49</th>
<th>...</th>
<th>94</th>
<th>95</th>
<th>96</th>
<th>97</th>
<th>98</th>
<th>99</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>1</td>
<td>2</td>
<td>...</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>...</td>
<td>94</td>
<td>95</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>$X_2$</td>
<td>A</td>
<td>A</td>
<td>...</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>B</td>
<td>...</td>
<td>B</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>D</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>$Y$</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>100</td>
<td>200</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 3: Dataset of experimental example

In this dataset, 95% of the observations with zero for response variable. For observations with non-zero response, two small size claims, two medium size claims, and one large size claim are considered. To be more specific, for the $i$th observation, $i = 1, 2, 3, ... 100$,

$$X_{1i} = i, \quad X_{2j} = \begin{cases} A & 1 \leq i < 48, \\
B & 48 \leq i < 96, \\
C & 96 \leq i < 98, \\
D & 98 \leq i < 100, \\
E & i = 100, \end{cases} \quad \text{and} \quad Y_i = \begin{cases} 0 & 1 \leq i < 96, \\
1 & i = 96, \\
2 & i = 97, \\
100 & i = 98, \\
200 & i = 99, \\
1000 & i = 100, \end{cases}$$

As in the motivating example of Sections 2.2 and 3.4, we only consider the first splitting step $u = 1$ to illustrate the comparison among the three tree-based models. If $j = 1$, there are 99 possible splitting points, namely $s = 1, 2, \ldots, 99$. If $j = 2$, there are 4 splitting point, namely $s = A, B, C, D$, corresponding to the splitting subsets $\{A\}, \{A,B\}, \{A,B,C\}, \{A,B,C,D\}$. Figure 10 and Figure 11 compares the goodness of split among the tree-based models by the continuous explanatory variable $X_1$ and the categorical variable $X_2$ respectively. Observe that the optimal split determined by the Canberra tree model is different from those by the other two models. Based on the Canberra tree model, if $\hat{j} = 1$, then $\hat{s} = 95$, separating zero responses from non-zero responses as anticipated. It is also noteworthy that, when $s > 95$ for $j = 1$, or when $s > B$ for $j = 2$, the goodness of split determined by the Canberra tree model declines, while the goodness of split determined by the classical ANOVA best split and the WSSE tree model are still increasing. This is consistent with our previous observation in the motivating example that the traditional SSE loss function tends to only partition the observations with small and large claims without separating zero from non-zero. Additionally, it is remarkable that the splitting performance of the WSSE tree is similar to that of the Canberra tree model when $s \leq 95$ for $j = 1$, but almost identical to that of the ANOVA tree when $s > 95$ for $j = 1$. This is because this split, $s \leq 95$ for $j = 1$, separates zero from non-zero by putting all observations with non-zero response into the right daughter node and zero into the left node. Thus, the WSSE applied
on the daughter node is approximately equal to the SSE. The same conclusion is again drawn from a more complicated setting that mimics a real-life insurance claim dataset.

Figure 10: Comparison of splitting performance by continuous variable

Figure 11: Comparison of splitting performance by categorical variable
Appendix C  Scalability of Simulation Study

Due to computational constraints on investigating the robustness of the performances in Section 4.3, we considered datasets with 100 observations in Section 4. In the context of insurance, datasets are usually larger, say at least 50,000 observations. Thus, in this appendix, we revisit the splitting and predictive performances of a new synthetic dataset, that has 50,000 observations among which 90.03% of them have a zero response.

We consider 4 continuous and 4 categorical explanatory variables. The continuous explanatory variables are sampled from a 4-dimensional multivariate normal distribution with a mean 0 and the variance-covariance matrix \( \Sigma \), where \( \Sigma_{ij} = (0.3)^{i-j} \). The categorical explanatory variables, with no correlation among them, are generated by random sampling from the set of integers \((-0.07, -0.06, -0.03, -0.02, 0.01, 0.04, 0.05, 0.08)\), with equal probabilities of 0.125. The vector of regression coefficients used is

\[
\beta = (-5, \begin{array}{c} 3 \\ 1 \text{ categorical} \\ -1 \\ 1 \text{ categorical} \\ -0.05 \\ 1 \text{ categorical} \\ 0 \\ 1 \text{ continuous} \\ -3 \\ 1 \text{ categorical} \\ -1 \\ 1 \text{ continuous} \\ -0.05 \\ 1 \text{ continuous} \\ 0 \\ 1 \text{ continuous} \end{array})^T.
\]

The response variable is generated from a Tweedie GLM framework, with the dispersion parameter \( \phi = 50 \) and the variance power parameter \( \xi = 1.7 \).

We fit the ANOVA, WSSE, and Canberra tree models to this new synthetic dataset, with hyperparameters setting as minsplit \( N = 2 \), maxdepth = 5, and cost-complexity \( \alpha = 1 \times 10^{-7} \). Table 4 presents the mean Tweedie deviance for fitted values of the response variable provided by the three models. The Canberra tree model has the lowest Tweedie deviance, followed by the WSSE tree model, while the ANOVA tree model exhibits the highest value. Thus, the Canberra tree model offers the best fit on the new synthetic dataset. Note that the WSSE tree model and the Canberra tree model require more computing time than the built-in ANOVA tree model, due to the fact that we implement the WSSE tree and Canberra tree models by adding user-defined splitting functions for \texttt{rpart}. In \texttt{rpart}, the user-defined methods are expected to be much slower than the built-in ones, see Therneau (2019). To further utilize the WSSE tree and Canberra tree models as base learners for ensemble methods, additional modifications to the \texttt{C} base of \texttt{rpart} or implementation with \texttt{Python} are necessary.

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean Tweedie Deviance</th>
<th>Computing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANOVA Tree Model</td>
<td>71.0234</td>
<td>0.3519</td>
</tr>
<tr>
<td>WSSE Tree Model</td>
<td>15.7885</td>
<td>341.3640</td>
</tr>
<tr>
<td>Canberra Tree Model</td>
<td>11.0089</td>
<td>15.3625</td>
</tr>
</tbody>
</table>

Table 4: ANOVA, WSSE, and Canberra model performances on large synthetic dataset with computing time (in seconds)