Missing Data in Conditional Inference Trees

by

Danielle Manapat

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved March 2023 by the
Graduate Supervisory Committee:

Kevin Grimm, Co-Chair
Mike Edwards, Co-Chair
Samantha Anderson
Daniel McNeish

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

Decision trees is a machine learning technique that searches the predictor space for the variable and observed value that leads to the best prediction when the data are split into two nodes based on the variable and splitting value. Conditional Inference Trees (CTREEs) is a non-parametric class of decision trees that uses statistical theory in order to select variables for splitting. Missing data can be problematic in decision trees because of an inability to place an observation with a missing value into a node based on the chosen splitting variable. Moreover, missing data can alter the selection process because of its inability to place observations with missing values. Simple missing data approaches (e.g., deletion, majority rule, and surrogate split) have been implemented in decision tree algorithms; however, more sophisticated missing data techniques have not been thoroughly examined. In addition to these approaches, this dissertation proposed a modified multiple imputation approach to handling missing data in CTREEs. A simulation was conducted to compare this approach with simple missing data approaches as well as single imputation and a multiple imputation with prediction averaging. Results revealed that simple approaches (i.e., majority rule, treat missing as its own category, and listwise deletion) were effective in handling missing data in CTREEs. The modified multiple imputation approach did not perform very well against simple approaches in most conditions, but this approach did seem best suited for small sample sizes and extreme missingness situations.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The general purpose of psychological science is to explain and predict human behavior. Psychology explains human behavior by theorizing the mechanisms behind a mental process and uses predictions as a way to anticipate behaviors before they occur (Yarkoni & Westfall, 2017). Research conducted in this field aims to explain and predict behaviors simultaneously with the use of statistical models. However, Yarkoni and Westfall (2017) argue explanation and prediction are two separate goals that researchers ultimately have to choose between: researchers can either develop complex models that will accurately predict behaviors but fail to respect known psychological constraints, or develop simple models that are theoretically elegant but have limited capacity to make accurate predictions. Historically, social and behavioral fields have favored explanation as the primary goal of research with prediction as a secondary goal. But more recently, researchers have started to identify situations where prediction should be prioritized over explanation (Yarkoni & Westfall, 2017).

Exploratory methods, focused on prediction, are becoming increasingly popular in psychological research (Yarkoni & Westfall, 2017). Specifically, exploratory methods based on *machine learning theory* adopted from other fields (e.g., computer science) have recently been considered and combined with statistical methods common in psychological research (e.g., structural equation modeling). As more researchers engage in data exploration, methodologists are adapting and implementing machine learning techniques in psychological research (e.g., Brandmaier, von Oertzen, McArdle, & Lidenberger, 2013; Grimm, Mazza, & Davoudzadeh, 2017; Hajjem, Bellavance,

Larocque, 2011; Jacobucci, Grimm, & McArdle, 2016; Masyn, 2013; McNeish, 2015; Sela & Simonoff, 2012; Strobl, Malley & Tutz, 2009). Though machine learning methods have started to gain the attention of quantitative psychologists, there has been little research conducted on how these methods perform under data conditions commonly seen in the psychological and behavioral sciences. One common feature of behavioral science data is incompleteness or missingness.

Regardless of the nature of a study (exploratory or theory-driven), missing data are an inescapable problem in psychological research. The type of missing data we consider in this paper are observations that do not have a value for a given variable. Encountering missing data is inevitable because it is often due to situations beyond the researcher's control (e.g., participant unwillingness to divulge information, inadvertent skipping, fatigue, time considerations, etc.). Missing data are problematic because they can introduce *nonresponse bias* when there are systematic differences between nonresponding and responding participants. Nonresponse bias affects estimated model parameters and threatens the validity of conclusions drawn from a statistical model. Since researchers cannot completely prevent nonresponse bias, there has been an extensive amount of research conducted on the topic of missing data and various statistical approaches have been developed to handle missing data.

Though missing data has been extensively studied in theory-driven / confirmatory statistical frameworks in psychology (e.g., ANOVA, regression, latent variable modeling, psychometrics, etc.), the approaches for dealing with missing data in the machine learning / exploratory framework have been under-researched. There is existing literature on missing data in the fields that traditionally use machine learning methods like

computer science; however, understanding the classic missing data problem from an interdisciplinary standpoint, such as through the lens of psychological science, may offer a new perspective for how best to deal with missing data when employing machine learning techniques. For example, there is potential for an interdisciplinary perspective to offer new ideas in terms of the scope, research methods, and approaches for dealing with the missing data problem when conducting machine learning methods.

This dissertation focuses on the topic of missing data in a specific machine learning method, *decision trees*. My objectives are to: (1) review current literature regarding missing data in decision tree algorithms, (2) propose a modified multiple imputation approach for handling missing data in *conditional inference trees*, and (3) conduct a simulation study to evaluate the proposed approach.

CHAPTER 2

AN OVERVIEW OF DECISION TREES

In the 1950s, scientists began to wonder if computers could "think" and started testing whether machines could perform intellectual tasks normally carried out by humans (Chollet & Allaire, 2018). This led to the development of artificial intelligence (AI), which is an interdisciplinary science that aims to design computer algorithms that can perform tasks that require human intelligence (Chollet & Allaire, 2018). AI has become increasingly popular in science, technology, media, and popular culture. There are numerous famous applications, such as computers that can play chess, self-driving cars, chatbots, tailored media streaming (e.g., Netflix and Spotify), targeted advertisement, and social media content. In the early stages of AI, computers were able to perform at the human level only when programmed using a large set of rules for manipulating information (Chollet & Allaire, 2018). These early stages of AI produced computers that were efficient within the limits of the pre-programmed rules. However, computers were not able to handle vague, novel problems until the development of machine learning algorithms (Chollet & Allaire, 2018).

Machine learning is a subfield of AI which was developed to mimic human decision making without relying on pre-programmed rules for each specific decision. Instead, the machine learning algorithms are programmed to develop its own rules for solving problems by *learning* from past experience. First, the algorithms need to understand the relationships between certain attributes and a desired outcome that has been observed. This information is given to an algorithm in the form of data which contain predictor and outcome variables. Once the algorithms are given data with known

outcome values as *training data*, they explore the predictor data space and finds natural patterns that it can use to make decisions. This process is sometimes referred to as *data mining* because the algorithm searches through large-scale data and "mines" out patterns within a vast data space. Once patterns are determined from the data in the form of a model, the same model is applied to a new data set to evaluate how well the algorithm generalizes. If the algorithm performs well at predicting values in new test data, the researchers are confident that the model will perform similarly when the outcome values are unknown.

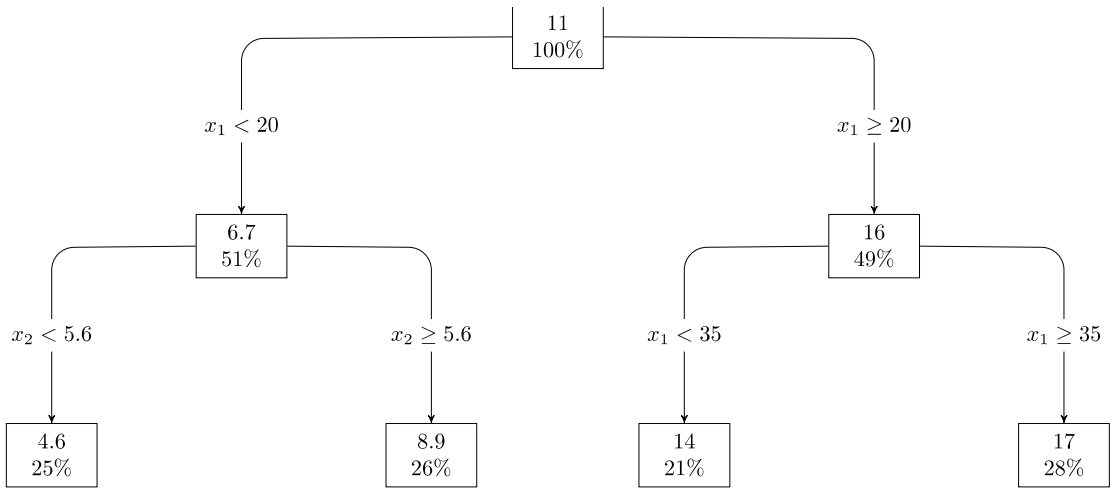
Machine learning is a large field comprised of numerous algorithms (see Carbonell, Michalski, & Mitchell, 1983 for an extensive overview). Decision trees have become one of the most popular machine learning methods (Berk, 2008). In this paper, I provide a broad overview of decision trees, with a specific focus on the Classification and Regression Tree (CART) algorithm (Breiman et al., 1984) and the Conditional Inference Tree (CTREE) algorithm (Hothorn et al., 2004).

Decision trees

Decision trees were developed to discriminate among classes (categories) of objects (outcome variables; Carbonell, Michalski, & Mitchell, 1983). The purpose of decision tree algorithms is to select object attributes and values of these attributes that identify sets of objects with identical classification (Carbonell, Michalski, & Mitchell, 1983). This process was developed to resemble human reasoning in a flowchart structured as an inverted tree (shown in Figure 1). The basic structure of the tree represents the decisions the algorithm makes to arrive at its prediction of the outcome.

Figure 1

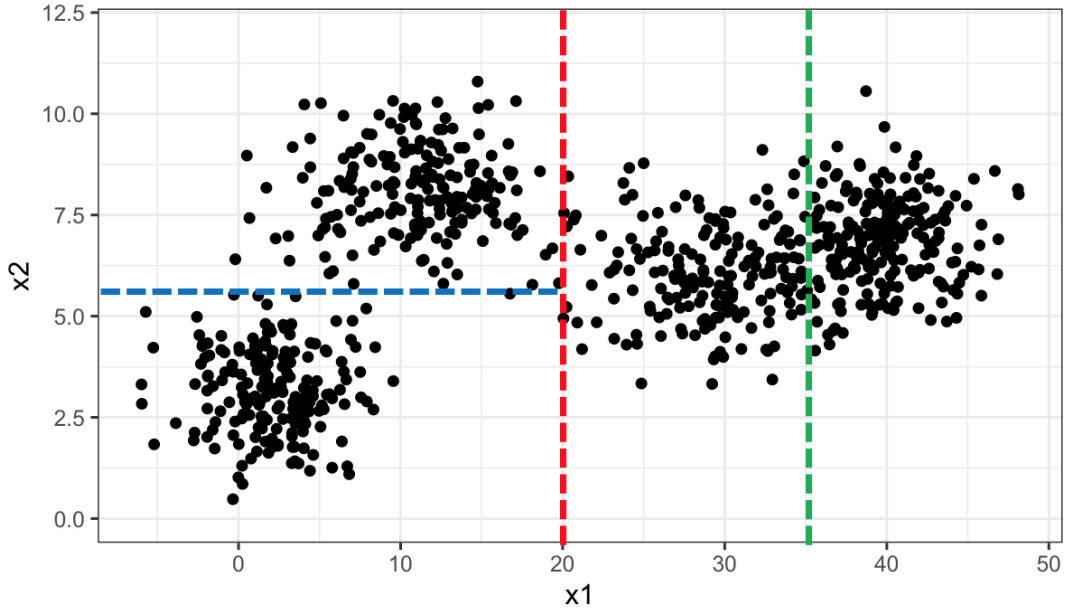
Decision Tree Example: Diagram



Decision trees typically use recursive partitioning, which is a process that involves repeatedly splitting a multivariate data space into two distinct sub spaces based on the relationship with the outcome. Consider the data space in Figure 2 with two predictors, x_1 and x_2. Splitting rules are used to identify the regions of a multivariate space that are most similar. For example, the first split (shown in red) partitions the data space into two distinct subspaces. The first region, R_1 includes any data point with a value on $x_1 < 20$ and the other region, R_2, includes any data point where $x_1 \geq 20$. The partitioning process is repeated until the distinct sub spaces are considered to have identical classification. Note that each of the regions from the first split (shown in red) were each split again to create four sub spaces with identical classifications. For example, R_1 was further split into two distinct regions in which data points located where $x_1 < 20$ and $x_2 < 5.6$ are grouped into one region and the other region contains data points located where $x_1 < 20$ and $x_2 \geq 5.6$. Similarly, R_2 was split into two sub regions based on a cutoff value of 35 on x_1.

Figure 2

Decision Tree Example: Partitioning Illustration



Note. This figure corresponds with the decision tree in Figure 1 and illustrates the partitioning of the data space. The first split is shown in red with a cutoff value of 20 on x_1. The second layer of splits is shown in blue with a cutoff value of 5.6 on x_2 and shown in green with a cutoff value of 35 on x_1.

The splits in Figure 2 are often represented in tree like structure shown in Figure 1. The *root* of the tree is the very first node in the diagram which corresponds to the entire data space in Figure 2 (prior to making any splits). Each node specifies the mean of the outcome within a data space that contains a certain percentage of the data. For example, the root node indicates that the outcome has a mean of 11 in the data space which contains 100% of the data points. The *branches* are the arrows that specify which variable and values on the variable were used to split the data. For example, branches from the root node indicate that any case with a value less than 20 on x_1 would be placed in one node and any value greater than or equal to 20 would be place in the other node (this corresponds to the first split shown in red in Figure 2). The two nodes created from

the root node are referred to as *daughter* or *child* nodes with the root node referred to as the *parent* node. *Parent* nodes represent the data that was used for a split into two separate groups that resulted into two *child* nodes. The nodes that have branches (i.e., two arrows) that point beneath are *nonterminal* nodes (Breiman et al., 1984). For example, the first three nodes in the tree would be considered *nonterminal*. The very last nodes without any branches extending out beneath are considered *leaves* or *terminal* nodes.

The process of determining how partitions of the data are made and evaluated differ across decision tree algorithms. I discuss two different algorithms: CART (Breiman et al, 1984) and CTREE (Hothorn et al., 2004). First, I provide a broad overview of classification and regression trees. Then I will describe one of the most popular algorithms, CART, followed by an alternative algorithm, CTREE, which is the focus of this study.

The Classification and Regression Tree (CART) Algorithm

The CART algorithm was developed by Breiman et al. (1984) for conducting both classification and regression trees. CART is a greedy decision tree algorithm that recursively partitions data and fits a simple prediction model within each partition (James, Witten, & Hastie, 2013; Loh, 2011). The term *greedy* indicates that the CART algorithm searches for the best possible outcome without considering any previous or future splits (Berk, 2008). Three critical aspects of the CART algorithm are *variable splitting*, *stopping criteria*, and *outcome prediction*.

Variable Splitting

For variable splitting, the CART algorithm selects the variable and partitioning value that splits the data into two groups where the outcome is maximally homogenous

within each group (Breiman et al., 1984). The two resulting groups are often referred to as *child* nodes (with the node that was split referred to as the *parent* node). All values of the predictors are considered decision points to partition the data into two child nodes. For a regression tree (numeric outcome), the predictor variable and splitting value that minimizes the residual sums of squares (RSS) is used to split the node (Berk, 2008). For a classification tree (categorical outcome), the predictor variable and splitting value that best minimizes the Gini index (entropy can also be used) is used to partition the node.

A regression tree is grown using RSS as the criterion for variable splitting. James et al., (2013) describes building regression trees by considering a predictor space that is made up of values from predictors $X_1, X_2, X_3, \ldots, X_P$ that can be partitioned into J distinct regions $R_1, R_2, R_3, \ldots, R_J$. Observations that fall within each R_j region are given predicted values equal to the mean of the observations that fall within R_j. Considering all possible predictors, the variable X_J and splitting value s are chosen to split the predictor space into two distinct sub spaces $\{X|X_j < s\}$ and $\{X|X_j \geq s\}$ based on which value minimizes the RSS (James et al., 2013). First, the algorithm considers split value s and predictor j for splitting the predictor space into two nodes or regions, such that

$$R_1(j,s) = \{X|X_j < s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\}$$

and computes the RSS as

$$\sum_{i:\, x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2 \qquad (1)$$

where $\hat{y}_{R_1}$ is the mean for the observations within region $R_1(s,j)$ and $\hat{y}_{R_2}$ is the mean for the observations within region $R_2(s,j)$ (James et al., 2013). This process is continued for all splitting values and predictors, and the predictor and split value that minimizes the

RSS is chosen to partition the data as long as the improvement in RSS meets some threshold (i.e., stopping criterion). Once the split occurs, splitting is considered again within each of the resulting child nodes and the same process is repeated. Within the predictor space region determined by a previous split, values of s and j are again considered to split that node into two child nodes that minimizes the RSS. This process continues by repeatedly splitting the predictor space to grow the regression tree until a stopping criterion is reached.

A classification tree is grown much like a regression tree. Instead of predicting the mean of all the training set data that within a predictor space region, classification trees predict which class k each observation belongs to by determining the modal class for the observations within a region (James et al., 2013). For classification trees, the splitting criterion is node purity as measured by the Gini index or entropy. The Gini index is a measure of uncertainty and used to assess whether a node contains observations mostly from a single class. The Gini index is calculated as

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{2}$$

where $\hat{p}_{mk}$ is the proportion of the observations in region m from class k. An alternative measure to node purity is entropy, which is a measure of information. The entropy is calculated as

$$D = -\sum_{k=1}^{K} \hat{p}_{mk}\log(\hat{p}_{mk}) \tag{3}$$

where $\log()$ is the natural logarithm. Whether the Gini Index or entropy is used to determine quality of a potential split, small values indicate greater node purity (i.e., that a

node contains observations mostly from a single class). Splits that lead to the smallest Gini Index or entropy value (indicating greatest node purity) are retained when partitioning. This process in repeated on each child node to grow the classification tree until a stopping criterion is reached.

Stopping Criteria

Stopping criteria include tree depth, sample size required to partition a node, or a minimum improvement in prediction accuracy. The tree depth criterion is specified as the number of layers, such that the decision tree is grown until the desired level of splits is reached. Another stopping criterion is related to sample size. That is, the tree is grown until it reaches a specified minimum number of cases in one or more of the terminal nodes. The minimum improvement in prediction accuracy criterion indicates a tree will be grown by adding a layer of splits only if the new layer improves how well the predicted values match the actual outcome values by a certain threshold. One commonly used method is *cost complexity pruning*, which can be considered a *pre-pruning* method (Therneau & Atkinson, 2019). Cost complexity pruning indicates which split does not improve prediction accuracy beyond a pre-specified threshold (e.g., minimum reduction of RSS in regression trees). A threshold is often referred to as the *complexity parameter*, or *cp* value, and computed using the following formula:

$$R_{cp}(T) = R(T) + cp \cdot |T| \cdot R(T_1) \quad (4)$$

where R is the risk, $|T|$ is the number of splits for a given tree, and T_1 is the tree with no splits (Therneau & Atkinson, 2019). The prespecified cp value indicates at which point a tree stops splitting based on its prediction accuracy. A common value to set the cp parameter is .01 (Therneau & Atkinson, 2019). In a regression tree, *cp* is a measure of the

overall increase in R^2. Thus, if $cp = .01$, then a split of the node is not retained if the R^2 doesn't increase by more than 1% and no further splits are considered. Stopping criteria based on minimum improvement in prediction accuracy reduces tree size so that tree pruning via cross-validation will only need to remove a few branches to obtain optimal tree depth.

Pruning

Once a stopping criterion is reached, the decision tree can be pruned, or reduced in size, based on *k*-fold cross-validation. The purpose of cross validation is to avoid overfitting a tree to the observations in hand (i.e., training data) by estimating its error rate for a new sample (i.e., test data). Since an actual test data set should only be considered once when the final tree structure is determined, cross-validation allows for pruning a tree based on an estimate of the test error rates. Cross-validated estimates are obtained using only the training data set, which is the data used to construct the tree. The *k*-fold cross-validation approach randomly divides the training data set into *k* approximately equally sized subsets that are referred to as folds. One of the folds is considered a *holdout* sample that will act as a test data set to obtain a test error estimate, while a tree is fit to the rest of the $k - 1$ folds. For a regression tree, the Mean Square Error (MSE) is calculated when predicting values in the holdout fold. This process is repeated k times so that every fold will be the holdout sample exactly once. This process will produce k MSEs such as $MSE_1, MSE_2, \ldots, MSE_k$, and the cross-validated estimate (CV) is computed by averaging these MSE values (James et al., 2013). For a classification tree, the same process is conducted for *k*-fold cross-validation. However, instead of using MSE, the number of misclassified observations is evaluated k times from

each hold out fold. The CV for classification trees is computed by averaging the number of misclassified observations, Err_i (James et al., 2013). Both regression and classification trees are pruned to the tree size that produces the smallest CV estimate. **Conditional Inference Tree (CTREE) Algorithm**

Popular decision tree algorithms like CART use an exhaustive approach to recursive partitioning. At each partition, the algorithm searches across every possible split to maximize the homogeneity in each node and selects the best possible split. Exhaustive approaches have been known to lead to overfitted trees and biased variable selection by favoring variables with many possible splits (Hothorn et al., 2004). While overfit trees can be pruned back via cross validation, the problem with biased variable selection is not easily remedied.

To address these problems, Hothorn et al. (2004) developed conditional inference trees (CTREE) which uses a statistical approach to recursive partitioning. Unlike the exhaustive approaches, the statistical approach splits the data based on the association between predictors and outcome which allows for unbiased variable selection among variables with different scales. The algorithm tests the global null hypothesis of independence for every predictor and the outcome. If the hypothesis is rejected, the variable with strongest association to the outcome is selected to split. If the hypothesis is not rejected, then the algorithm stops splitting. This statistically motivated stopping criteria has been shown to produce trees with predictive performance as good as optimally pruned trees (Hothorn et al., 2004).

The general method for recursive partitioning by conditional inference is completed in the following three steps (Hothorn et al., 2004): First, test the global null

hypothesis of independence between each of the m predictors and the outcome Y. If the null is not rejected, the algorithm stops splitting. If the null hypothesis is rejected, the algorithm finds the predictor X_{j*} which has the strongest association with Y. Second, split X_{j*} into two distinct groups where the case weights w_L and w_R determine the subgroups. Last, the first two steps are recursively repeated with modified case weights.

Variable Selection and Stopping Criterion

The first step described above involves determining the splitting variable and stopping criterion. First, it is assumed that the conditional distribution $D(Y|X)$ of Y given the predictors X depends on a function f of the predictors:

$$D(Y|X) = D(Y|X_1, \dots X_m) = D(Y|f(X_1, \dots X_m)) \quad (5)$$

Within each node, the global test of independence is formulated in terms of m partial hypotheses $H_0^j: D(Y|X_j) = D(Y)$ with the null hypothesis $H_0: \cup_{i=1}^{m} H_0^j$ (Hothorn et al., 2004). Both variable selection and stopping criterion are determined by testing H_0. If we fail to reject H_0 at a pre-specified α, the algorithm stops partitioning. If we reject H_0, the association between Y and each of the predictors $X_j, j = 1, \dots, m$ is measured by *p*-values that indicate a deviation from the partial hypothesis H_0^j (Hothorn et al., 2004). The predictor with the strongest association with Y is the selected as the splitting variable.

The following formula is used to compare the associations between each of the predictors $X_j, j = 1, \dots, m$ and Y (Hothorn et al., 2004):

$$T_j(L_n, \boldsymbol{w}) = \text{vec}\left(\sum_{i=1}^{n} w_i g_i(X_{ij}) h(Y_i, (Y_1 \dots Y_n))^T\right) \in \mathbb{R}^{p_j q} \quad (6)$$

where L_n is the learning sample, w_i represents case weights (assumed here to be zero or one for convenience), $g_i: X_j \rightarrow \mathbb{R}^{p_j}$ is a non-random transformation of the predictor X_j, and $h_i: Y \times Y^n \rightarrow \mathbb{R}^q$ is considered the influence function which depends on the responses $(Y_1 \ldots Y_n)$ in a permutation symmetric way (Hothorn et al., 2004). The distribution of $T_j(L_n, \boldsymbol{w})$ under the partial hypothesis depends on the joint distribution of X_j and Y which is almost always unknown. Therefore, Hothorn et al., (2004) use permutation tests to rid the dependency by fixing the predictors and conditioning on all possible permutations of the responses. The permutation test procedures were originally developed by Strasser and Weber (1999), and the derivation can be found in Hothorn et al. (2004). The permutations allow for the calculation of the conditional expectation μ and covariance Σ, which in turn allows for the standardization of Equation 7. The result is a test statistic c that is used to compare the predictors. If the predictors have different scales of measurement, the test statistics $c(t_j, \mu_j, \Sigma_j), j = 1, \ldots, m$ will likely bias the variable selection. To unbias variable selection, the algorithm switches to the *p*-value scale and the *p*-values for the condition distribution of the test statistic $c(T_j(L_n, w), \mu_j, \Sigma_j)$ are used to compare predictors with different scales (Hothorn et al., 2004). The purpose is to identify the predictor with the minimum *p*-value P_j:

$$P_j = \mathbb{P}_{H_0^j}(c(T_j(L_n, w), \mu_j, \Sigma_j) \geq c(t_j, \mu_j, \Sigma_j) | S(L_n, w)) \quad (7)$$

where $S(L_n, w)$ is the symmetric group of all permutations of the elements $(1, \ldots, n)$ with case weights that are equal to 1 (Hothorn et al., 2004). The predictor with the smallest significant *p*-value is selected as the splitting variable and then evaluated to determine its optimal split point.

Splitting Criteria

Once a splitting variable is determined, the second step in the general partitioning algorithm involves determining the optimal split point. The goodness of split statistic is a special case of Equation 7

$$T_{j*}^{A}(L_n, \boldsymbol{w}) = vec(\sum_{i=1}^{n} w_i I\big(X_{j*i} \in A\big) h(Y_i, (Y_1, \ldots, Y_n))^T) \in \mathbb{R}^q \quad (8)$$

where A is a subset of the sample space of the predictor X_{j*} (Hothorn et al., 2004). Equation 9 measures the discrepancy between the two sample spaces under consideration. Again, the conditional expectation μ_j^A and covariance Σ_j^A are computed using the permutation test procedure originally developed by Strasser and Weber (1999). The split A^* that maximizes the test statistic over all possible sets of A becomes the optimal split point:

$$A^* = argmax\ c(t_{j*}^{A}, \mu_{j*}^{A}, \Sigma_{j*}^{A}) \quad (9)$$

After partitioning the data at the optimal split point, the entire procedure repeats. The algorithm recursively partitions the data by repeatedly searching for a splitting variable and optimal split point until the null hypothesis is not rejected.

CHAPTER 3

MISSING DATA

Missing data occur when an observation contains no value for a given variable and is often due to situations beyond the researcher's control. There are numerous situations that lead to missing data, which makes it difficult to know exactly how and why each missing value appears in a data set. Rubin (1976) proposed using observed variables to predict the occurrence of missing values and coined the term *missing data mechanisms* to classify relationships between missing values and observed variables. Missing data mechanisms describe how the propensity for a missing value relates to other variables and itself. Rubin (1976) presented three types of missing data mechanisms: *missing completely at random* (MCAR), *missing at random* (MAR), and *missing not at random* (MNAR).

Data are MCAR when missingness on variable x is unrelated to both the observed variables (i.e., non-x variables) and the underlying values of x itself (Enders, 2003; Rubin, 1976). MCAR situations are desirable because missing data patterns are unsystematic and therefore unlikely to bias results. However, MCAR requires the strict assumption that missing values are not related to any of the studied variables, which is rarely met in practice (Enders, 2010; Muthén et al., 1987; Raghunathan, 2004). Data are MAR when missingness is systematic and correlated with other variables in the data set. Specifically, data are considered MAR when missing values on the variable x are related to other variables in a data set but not related to x itself (Enders, 2003; Rubin, 1976). Data are MNAR when missing values on x are dependent on the underlying values of x itself (Enders, 2003; Rubin, 1976).

The missing data mechanisms determine how well a given missing data approach will perform. According to Baraldi and Enders (2010), deletion approaches (i.e., list-wise, pairwise, etc.) perform well in situations when data are MCAR, whereas more advanced approaches, such as multiple imputation or full-information maximum likelihood, outperform deletion and produce unbiased parameter estimates when data are MCAR or MAR. However, many approaches commonly used to handle missing data (e.g., deletion, single or multiple imputation, full-information maximum likelihood, etc.) do not perform well when data are MNAR. In the subsequent sections, I provide descriptions of the following approaches for handling missing data: deletion, single imputation, multiple imputation, and full-information maximum likelihood.

Deletion

A traditional method for treating missing data involves deleting or removing cases that contain missing values from an analysis. Deletion methods have been widely adopted since they provide a relatively simple solution to missing data. Specifically, listwise deletion and pairwise deletion are the most popular methods for treating missing data across the social and behavioral sciences (Peugh & Enders, 2004; Enders, 2010). Due to their popularity, both methods are widely available in most statistical software (Enders, 2010).

Listwise deletion removes any case in a data set that contains one or more missing values. The advantage of this approach is its simplicity. Listwise deletion does not require special software, unlike more complicated missing data approaches. Once all the cases with missing values are removed from the data set, the complete cases make up a consistent sample that may be used across different analyses. However, given the number

of cases with missing values, listwise deletion could result in considerably reduced sample sizes.

Pairwise deletion removes cases with missing values on an analysis-by-analysis basis. That is, cases with missing values are removed only if a variable needed for the analysis contains the missing value. If a case contains missing values exclusively on variables that are not of interest to the researcher, then that case would be included in the analysis. In comparison with listwise deletion, pairwise deletion is more likely to retain a higher percentage of cases resulting in larger sample sizes. Since values are removed analysis-by-analysis basis, this method will typically result in different samples for every analysis.

When considering deletion approaches, it is important to note that there are limitations to these approaches. The reduction in sample size from deleting cases reduces statistical power. Therefore, deletion approaches should only be considered if the percentage of missing values is fairly small or there is a large sample of complete cases (Enders, 2010).

Imputation

Another common way of handling missing data is to impute (i.e., fill in) the missing values before conducting an analysis. This approach alleviates some of the problems of deletion approaches by filling in missing values to retain sample size and statistical power. Imputation approaches typically fall under the category of *single imputation*, which replaces each missing value with a single value, or *multiple imputation* which involves repeatedly copying a data set and filling in the missing values with slightly different estimates using random variation. This section will focus on methods

for single imputation such as regression imputation, stochastic regression imputation, and hot deck imputation.

Regression Imputation

Regression imputation predicts missing values from the complete data via regression equations. Since variables are often correlated, the complete variables can be used to predict missing values on other variables. In regression imputation, the complete cases are used to create a regression estimate of the missing values. The resulting predicted values replace the missing values to create a complete data set for subsequent analyses. As the number of variables increase (specifically those containing missing values) so does the potential for more complex missing data patterns. For a simple bivariate example, the regression equation would predict missing values on a variable X_I^* from the complete case variable, X_J

$$X_I^* = \beta_0 + \beta_1(X_J) \quad (10)$$

A major limitation of this approach is that all imputed values fall on the same regression line, which attenuates variability and increases R^2 (Enders, 2010). Regression imputation has been shown to produce biased results by overestimating correlations and R^2 (Beale & Little, 1975; Gleason & Staelin, 1975; Kromrey & Hines, 1994; Olinsky, Chen, & Harlow, 2003; Raymond & Roberts, 1987; Buck, 1960). However, methods such as stochastic regression imputation have been developed address this limitation.

Stochastic Regression Imputation

Since regression imputation produces biased results by creating imputed variables that perfectly correlate with missing data, stochastic regression imputation was developed to eliminate this bias by introducing variability in the imputed values (Enders, 2010).

Stochastic regression imputation uses the same methods as regression imputation but adds a normally distributed residual term to each predicted score. Consider the bivariate example from regression imputation where missing values on a variable X_I^* are predicted from the complete case variable, X_J. Stochastic regression imputation differs from regression imputation by including the normally distributed residual term, Z

$$X_I^* = \beta_0 + \beta_1(X_J) + Z \quad (11)$$

where

$$Z \sim N(0, \hat{\sigma}^2_{X_I|X_J})$$

Stochastic regression uses the same method as standard regression, but includes the addition of the residual term, Z. The residual term is a random value from a normal distribution with a mean of zero and variance equal to the residual variance from the regression of the variable with missing values X_I on the complete case variable X_J Overall, stochastic regression imputation produces unbiased parameter estimates when data are MAR. In fact, the stochastic regression imputation often performs well in comparison to the typically favored multiple imputation approach since both methods share the same imputation procedure. However, a limitation of stochastic regression imputation is that it is a *single* imputation approach, which ultimately produces attenuated standard errors and risks inflating Type-I errors.

Hot Deck Imputation

Hot deck imputation replaces missing values with scores from similar respondents. An observation containing a missing value on a target variable will be matched with other observations on pre-specified matching variables (e.g., race, gender, social economic status, grade level, marital status, etc.). Matching can be done with both

categorical and continuous variables. The researcher then replaces the missing value with a random draw from the distribution of the matched responses.

A limitation of hot deck imputation is that this approach has been shown to produce substantially biased estimates of correlations and regression coefficients (Enders, 2010; Schaffer & Graham, 2002). As with most imputation procedures, hot deck imputation underestimates standard errors and often requires additional procedures (e.g., jackknife) to increase sampling error in the imputed values (Enders, 2010).

Multiple Imputation

Multiple imputation is generally the favored imputation approach because it produces estimates that are consistent, asymptotically efficient, and asymptotically normal when data are MAR and all assumptions are met (Allison, 2002). The multiple imputation approach involves a three-step procedure that includes the imputation phase, analysis phase, and pooling phase.

Imputation Phase

It has been shown that single imputation (without random variation) tends to underestimate the variances and covariances of variables that contain missing data, which leads to biased parameter estimates (Allison, 2002). Multiple imputation, on the other hand, is more effective at producing unbiased estimates by increasing variance in the variable with missing values. Specifically, this is done in the imputation phase by repeatedly copying a data set and filling in the missing values with slightly different estimates using random variation.

Many procedures have been developed for filling in missing values with random variation (Lavori, Dawson, & Shera, 1995; Raghunathan, Lepkowski, Van Hoewyk, &

Solenberger, 2001; Royston, 2005; Schafer, 1997; van Buuren, 2007). However, the most popular imputation method in the social and behavioral sciences (Allison, 2002; Enders, 2010) is the data augmentation algorithm (Schafer, 1997; Tanner & Wong, 1987). Data augmentation relies heavily on Bayesian methodology in a two-step procedure, where missing values are repeatedly imputed in the I-Step and parameters are repeatedly updated using posteriors in the P-Step (Enders, 2010). The data augmentation process starts with the I-Step followed by the P-Step to update the estimated values, which informs the next I-Step, and so forth. These two steps toggle back and forth until a specific convergence criterion is reached.

I-Step. The purpose of the imputation phase is to repeatedly impute unique values for all missing values in a data set. The I-Step first uses regression equations to predict missing values where random residuals are added to the predicted scores to create random variance in the imputed values (identical to stochastic regression). From the imputed data set, the estimated mean vector and covariance matrix are used to create a conditional distribution (also known as posterior predictive distribution) in the P-Step, which will be discussed further in the next section. In the next I-Step, imputed values are drawn from the conditional distributions estimated in the previous P-Step. This process is summarized in the following equation from Enders (2010)

$$Y_t^* \sim p(Y_{\text{min}} | Y_{obs}, \theta_{t-1}^*) \quad (12)$$

where each I-Step is represented in terms of t and imputation values for a particular I-Step is represented by Y_t^*. The proportion of missing data is Y_{min} and the proportion of observed data is Y_{obs}. Whereas, θ_{t-1}^* represents the parameters used to generate the imputation regression equation; specifically, θ_{t-1}^* is the estimated mean vector and

covariance matrix from the P-Step proceeding a particular I-Step (i.e., $t-1$; Enders, 2010).

P-Step. As mentioned previously, the purpose of the P-Step is to estimate the mean vector and covariance matrix from imputed values in the previous I-Step and create a conditional distribution that can then be used to draw values from in the next I-Step. The P-step uses a Bayesian framework to estimate the mean vector and covariance matrix from imputed values and then new parameter values are generated using a Monte Carlo simulation.

First, the P-Step computes sample means ($\hat{\mu}$) as well as sum of squares and cross products matrix ($\hat{\Lambda}$) from imputed values in the proceeding I-Step to define the posterior distribution of the covariance matrix

$$(\hat{\Sigma}\,|\hat{\mu}, Y) \sim W^{-1}(N-1, \hat{\Lambda}) \tag{13}$$

where $p(\hat{\Sigma}\,|\hat{\mu}, Y)$ is the posterior with mean vector $\hat{\mu}$ and imputed data matrix Y from previous I-Step, which follows an inverse Wishart distribution $\sim W^{-1}$ with $N-1$ degrees of freedom and the sum of squares and cross products matrix $\hat{\Lambda}$ (Enders, 2010). Next, a Monte Carlo simulation uses the posterior defined in Equation 14 to draw a new covariance matrix which is referred to as the simulated covariance matrix $\Sigma*$ (Enders, 2010).

Once the data augmentation algorithm obtains a new covariance matrix, then a new set of means is derived. This procedure involves estimating sample means and using the simulated covariance matrix to create the new set of means. First, the posterior distribution of the mean vector is defined by

$$p(\hat{\mu}|Y,\Sigma) \sim MN(\hat{\mu}, N^{-1}\Sigma^{*}) \quad (14)$$

where $p(\hat{\mu}|Y,\Sigma)$ is the posterior, which follows a multivariate normal distribution (i.e., $\sim MN$) with sample mean vector $\hat{\mu}$ and simulated covariance matrix Σ^{*} (Enders, 2010). Next, a Monte Carlo simulation is used to draw a new set of means $\hat{\mu}^{*}$ from the posterior defined in Equation 15.

The purpose of the P-Step is to use imputed values to create μ^{*} and Σ^{*} that can be then used for the next I-Step. Overall, the P-Step can be summarized in the following equation

$$\theta_t^{*} = p(\theta|Y_{obs}, Y_t^{*}) \quad (15)$$

where θ_t^{*} is the estimated parameters (μ^{*} and Σ^{*}) from the P-Step based on Y_{obs} observed data and Y_t^{*} imputed values from the previous I-Step (Enders, 2010). The new parameter values $\hat{\mu} *$ and Σ^{*} from the P-Step are used in the next I-Step to generate a new set of regression coefficients. The regression coefficients generate new imputed values and create an imputed data set. The new imputed values from that I-Step are then used for the next P-Step. The I- and P-Steps are repeatedly cycled to create imputed data sets until a convergence criterion is reached.

Convergence. Using Markov Chain Monte Carlo procedures (Jackman, 2000), the algorithm cycles between the I- and P-Steps to create the following data augmentation

$$Y_1^{*}, \theta_1^{*}, Y_2^{*}, \theta_2^{*}, Y_3^{*}, \theta_3^{*}, \ldots Y_t^{*}, \theta_t^{*} \quad (16)$$

where Y_t^{*} denotes imputed values from I-Step t and θ_t^{*} denotes the estimated parameters from P-Step t (Enders, 2010). The long chain of imputed values across the I-Steps are essentially drawn from a distribution that averages over the entire range of the posterior distribution. Similarly, the simulated parameters from the long chain of P-Steps are

drawn from the posterior distribution that averages over all possible values of missing data.

A feature of the I- and P-Step cycle is that each sequential step (i.e., t and $t+1$) produces imputed values that are dependent on the previous imputations. This is because the simulated parameters of a P-Step are determined by the imputed values of the previous I-Step. While imputed values derived from a particular I-Step are determined by the simulated parameters from the previous P-Step. The I- and P- step dependency produces successive data sets that are correlated to some degree.

Data augmentation converges when distributions become stationary and do not change in a systematic way. A complicated aspect of this definition is the dependent nature of the I- and P-Steps. Therefore, it is important to assess how many cycles are needed before imputed data sets from t and $t+k$ can be considered independent. A common method for evaluating the number of k imputations that need to be conducted before t and $t+k$ are independent involves assessing the behavior of the simulated parameters θ_t^* over many P-Steps (Enders, 2010). If θ_t^* and θ_{t+k}^* are correlated, then the imputed values will likely be dependent. If θ_t^* and θ_{t+k}^* are not correlated, then the two parameter sets should produce independent imputations. Once desired number of cycles are completed, the imputed values from the final iteration of a single chain are used to create the first imputed data set (Azur, Stuart, Frangakis, & Leaf, 2011). The entire imputation process is repeated as many times is needed to reach the desired number of imputed data sets, m. The resulting imputed data sets will be analyzed in the analysis phase.

Analysis Phase

This phase involves running the analysis of interest m times; that is, once for each complete data set created in the imputation phase. This process does not involve any other variables or additional procedures. The results from each analysis are collected and stored so that they can be combined in the pooling phase.

Pooling Phase

The purpose of the pooling phase is to combine the results from all m analyses to create a single set of results. The analysis phase produces m parameters and standard errors. The purpose of the pooling phase is to combine all results into a single parameter estimate and standard error estimate.

Parameters. Rubin (1987) developed a simple method to calculate a single parameter estimate by taking the average over all m estimates:

$$\bar{\theta} = \frac{1}{m}\sum_{t=1}^{m}\hat{\theta}_t \tag{17}$$

where the single pooled estimate $\bar{\theta}$ is calculated by taking the sum of each estimate $\hat{\theta}_t$ for every data set t and dividing by the total number of imputations, m (Enders, 2010). However, this method assumes that the parameters are asymptotically normally distributed. In situations where this assumption does not hold, Schafer (1997) suggests applying transformations (e.g., z transformations for Pearson correlation coefficients) before the pooling phase.

Standard Errors. In addition to estimated parameters, the standard errors need to be pooled. Rubin (1987) also developed methods for averaging standard errors. There are two types of variances involved in pooling standard errors: within-imputation variance and between-imputation variance. Within-imputed variance estimates the sampling

variability if there was no missing data. Using complete cases only, the average of sampling variances can be calculated in the following equation

$$V_W = \frac{1}{m}\sum_{i=1}^{m} SE_t^2 \quad (18)$$

where the within-imputation variance V_W is calculated by summing each squared standard error SE_t^2 from data set t and dividing by the total number of imputations m (Enders, 2010).

As mentioned previously, multiple imputation is more effective than single imputation at producing unbiased estimates (Allison, 2002; Enders, 2010). This is because missing data has no variance and single imputation simply fills in missing values but does not account for sampling error, which typically leads to underestimated standard errors. Multiple imputation, however, introduces sampling error for the missing values by repeatedly copying a data set and filling in the missing values with slightly different estimates using random variation. Between-imputation variance accounts for the additional source of sampling error introduced across multiply imputed data sets. The average variance of parameter estimates across all m imputations (i.e., between-imputation variance) is calculated as

$$V_B = \frac{1}{m-1}\sum_{i=1}^{m} (\hat{\theta}_t - \bar{\theta})^2 \quad (19)$$

where the between-imputation variance represented by V_B is the average squared difference of each estimated parameter $\hat{\theta}_t$ for data set t and the single pooled estimate $\bar{\theta}$ (Enders, 2010)

Together the within-imputation and between imputation variances make up the total variance to create a single point estimate of standard error. Specifically, the total variance is

$$V_T = V_W + V_B + \frac{V_B}{m} \quad (20)$$

where the total variance is the sum of the within-imputation variance V_W, between-imputation variance V_B, and the variance of the single pooled estimate of $\bar{\theta}$ (i.e., representing sampling error of the mean estimate), $\frac{V_B}{m}$ (Enders, 2010). The last term in this equation represent serves as a correction for using finite number of imputation and essentially drops to 0 as m approaches infinity (Enders, 2010). The square root of V_T is the pooled multiple imputation standard error estimate.

In conclusion, multiple imputation is a three-phase process that can be computationally intense. The analysis phase requires repeatedly calculating possible imputed values by drawing from observed data and posterior distributions. The analysis and pooling phases are relatively the easy to implement, but these methods can become cumbersome with complex statistical models and many imputations. However, sophisticated software has been developed for user-friendly implementation of multiple imputation for a wide variety of statistical models. Though there are many options and popular software programs for employing multiple imputation (Allison, 2002; Enders, 2010) include NORM (Schaffer, 1997), PROC MI (SAS Institute Inc., 2015), and Multivariate Imputation by Chained Equations (MICE; van Buuren & Groothuis-Oudshoorn, 2011).

Predictive Mean Matching

Another popular method for implementing multiple imputation is predictive mean matching, which was originally proposed by Rubin (1986) and Little (1988). Predictive mean matching is the default approach in the `mice` package (van Buuren & Groothuis-Oudshoorn, 2011). First, this approach predicts values on a target variable using a specified imputation model. Then a set of complete cases with similar prediction as the missing entry are identified. The complete cases matched with the missing data entry are known as *donors*. One of the donors is then randomly selected and its observed value replaces the missing entry. The predictive mean matching approach optimizes each target variable separately and only requires a one-number summary that relates to the covariates and target variable (van Buuren, 2018).

There are various ways to select donors. Four popular methods for donor selection that have been identified by Andridge and Little (2010) will be described in this section. In the following descriptions, $\hat{y}_i$ represents the predicted value of the rows with observed y_i where $i = 1, \dots, n_i$. Whereas $\hat{y}_j$ denotes the predicted value of the rows with missing value y_j where $j = 1, \dots, n_j$. For the four donor selection approaches: (1) the first involves choosing a threshold η and take all i for which $|\hat{y}_i - \hat{y}_j| < \eta$. Then randomly selecting one donor from the candidates and replacing the missing value with the observed value of the donor. (2) The second approach involves selecting the closest donor candidate for which $|\hat{y}_i - \hat{y}_j|$ is minimized. (3) The third approach involves pre-specifying the number of candidate donors d for which $|\hat{y}_i - \hat{y}_j|$ is minimal and then randomly sampling one of them. (4) The last approach involves sampling one donor with

a probability that depends on $|\hat{y}_i - \hat{y}_j|$. For some approaches, any number of candidate donors can be specified; however, it is typical to use five candidate donors in a set.

In addition to donor selection, there are several ways to match candidate donors with missing values. Van Buuren (2018) describes four matching procedures labeled Type 0-3. In Type 0, $\hat{y} = X_{obs}\hat{\beta}$ is matched to $\hat{y}_j = X_{mis}\hat{\beta}$ where $\hat{\beta}$ is the estimate of β. This approach ignores sampling variability in $\hat{\beta}$ and therefore leads to improper imputations. Type 1 procedure involves $\hat{y} = X_{obs}\hat{\beta}$ is matched to $\dot{y}_j = X_{mis}\dot{\beta}$ where $\dot{\beta}$ is a value randomly drawn from the posterior distribution of β. In Type 2, $\dot{y} = X_{obs}\dot{\beta}$ is matched to $\dot{y}_j = X_{mis}\dot{\beta}$. Type 3 uses the following procedure: $\dot{y} = X_{obs}\dot{\beta}$ is matched to $\ddot{y} = X_{mis}\ddot{\beta}$ which represents two draws for β (one for the donor and one for the recipient). The advantages and disadvantages of using each matching procedure Type 0-3 can be found in van Buuren (2018).

Overall predictive mean matching is a popular approach to treating missing data. This method is robust to misspecifications of imputation model and provides imputation that possess characteristics of the complete data (van Buuren, 2018).

Limitations of Multiple Imputation

When considering multiple imputation, it is important to note the limitations. A major limitation is that it is possible to get different estimates for every application of multiple imputation. If multiple imputation is implemented correctly, the differences in estimates should be negligible. However, two researchers to use multiple imputation on the same data using the same methods will arrive at different parameter estimates and standard errors (Allison, 2002). Another limitation is that multiple imputation is

computationally cumbersome and sophisticated software is needed to implement multiple imputation (Allison, 2002).

Full Information Maximum Likelihood

Full information maximum likelihood (FIML) estimation is an approach for obtaining parameter estimates even when data contain missing values. Like multiple imputation, FIML is almost always better than traditional methods (e.g., deletion, mean substitution, single imputation, etc.) at producing unbiased parameter estimates when data are MAR or MCAR (Baraldi & Enders, 2010; Enders, 2010). FIML has become a popular approach that is widely available in statistical software packages (Enders 2010). Due to its popularity, this section will include a brief overview of FIML even though this approach does not apply to machine learning techniques like decision trees (which will be discussed in further detail in this section).

Maximum likelihood (ML) estimation uses all available data to identify which population parameters most likely produced the observed values in a dataset (Baraldi & Enders, 2010; Enders, 2010). The starting point of ML is to specify a population distribution, which is often the multivariate normal distribution in the social and behavioral sciences. Once the distribution is specified, a set of population parameters are evaluated. The likelihood that the observed data were drawn from a particular set of population parameters is calculated using the log-likelihood. Log-likelihood is a measurement of the standardized distance between a set of observed values and a specific set of population parameters like mean and variance. For example, the log-likelihood for a set of observed scores from the univariate normal distribution is

$$logL = \Sigma_{i=1}^{N} \log \left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-5\left(\frac{y_i - \mu}{\sigma}\right)^2} \right] \quad (21)$$

where the term in brackets is the probability density function which describes the shape of the normal curve. The squared z-score that appears in the exponent of the function is the standardized distance between an observed value and the population mean. Large z-scores produce small log-likelihood values, whereas small z-scores produce large log-likelihood values. A log-likelihood value is collected for each individual observation and the sum of the individual log-likelihood values produce the *sample log-likelihood*. ML repeatedly substitutes different population parameters until it identifies which set of parameters produce the highest sample log-likelihood.

This process can be extended to the multivariate framework. Consider the following matrix formula to calculate log-likelihoods for each individual case

$$logL_i = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log\ |\Sigma| - \frac{1}{2}(Y_i - \mu)^T \Sigma^{-1} (Y_i - \mu) \quad (22)$$

where k is the number of variables, Y_i is a vector of scores for an individual, μ represents the population mean and $\sum$ represents the population covariance. The formula still includes the squared z-score, which evaluates the standardized distance between a set of observed values from an individual and the population mean from a multivariate normal distribution. As previously mentioned, the sample log-likelihood is computed by summing together individual log-likelihood values. The ML process repeats this process with different population parameters until it identifies which set of parameters produce the highest sample log-likelihood.

ML estimation may be conducted with either complete or incomplete data. When working with the individual log-likelihood, treating missing data is often referred to as

full information maximum likelihood (FIML). Though the procedures are similar, the difference between working with complete and incomplete data is that the log-likelihood value needs to be calculated for each case when data are missing;

$$logL_i = -\frac{k_i}{2}\log(2\pi) - \frac{1}{2}\log\ |\sum_i| - \frac{1}{2}(Y_i - \mu_i)^T \sum_i^{-1}(Y_i - \mu_i) \qquad (23)$$

where k_i is the number of variables with complete data, Y_i is a vector of scores for an individual, μ represents the population mean and Σ represents the population covariance. The difference between Equation 23 and Equation 24 is that the population parameters now have a subscript i, which indicates that the matrices can vary across individuals so that log-likelihood is computed for each case using only the variables and parameters that have complete data (Enders, 2010). For example, if a dataset contained four variables, $x_1 - x_4$, and a particular case was missing a value on x_4, then FIML would calculate the individual log-likelihood value using the population parameters for $x_1 - x_3$ and ignore the parameters for x_4. The likelihood calculation for that particular case would not contain any reference to x_4. It is possible that the formula for calculating log-likelihoods could be different for each missing data pattern (Enders, 2010). Once the log-likelihoods are calculated for each case, the individual values are summed together to obtain a sample log-likelihood value. Even with missing data, the sample log-likelihood value represents the probability of drawing the observed values from the multivariate normal distribution with a particular mean vector and covariance matrix. Regardless of whether the data are complete or incomplete, the estimation process repeatedly computes various combinations population parameters until it identifies the set of parameters that produce the highest sample log-likelihood value.

While FIML is an effective method for obtaining parameter estimates across various statistical techniques (e.g., regression, structural equation modeling, item response theory, etc.), the procedure is not appropriate when working with a data-driven technique such as decision trees. By nature, decision trees do not assume a population distribution and estimate population parameters. However, other techniques like deletion and imputation have been adopted and applied in the machine learning framework. These traditional missing data techniques as well as ones specifically designed for decision trees are covered in the following section.

CHAPTER 4

MISSING DATA IN CONDITIONAL INFERENCE TREES

Missing data are problematic in decision trees because an observation with a missing value on the predictor variable is unable to be placed into a child node. Given the challenges of missing data handling in decision trees, multiple strategies have been developed, such as deletion approaches, surrogate splits, single imputation, and multiple imputation. A broad overview of each missing data approaches specifically designed for decision trees is described in the following sections.

Deletion

There are two deletion strategies that can be employed in conditional inference trees. The first is to simply remove observations where a missing value is present (aka listwise deletion). This approach is taken when preprocessing the data. The second strategy for conditional inference trees is to retain cases with missing values until a variable with missing values is selected (akin to pairwise deletion). For example, consider a case that contains a missing value on a predictor, X_J. This case would be retained in the decision tree until X_J is selected to partition the data. Thus, if X_J is not selected, then the case is retained in the model. Importantly, the case contributes to the formation of the decision tree until it cannot be placed into a child node because of the missing value.

Surrogate Splits

When an observation has a missing value on the splitting variable, surrogate splits use another variable in the data set to place the observation in the decision tree. Surrogate split is the default method for handling missing data in the `ctree` package (Hothorn et al., 2006) in R (R Core Team, 2020). In this approach, a missing observation for a

particular predictor variable is given a case weight set to zero. If the variable containing the missing observation is selected as a splitting variable, then the case weight is set to zero again and the splitting value is determined using complete observations. A surrogate split is implemented by finding where to place the missing observation that would lead to roughly the same division of observations as the original split (Hothorn et al., 2006). The missing observation is replaced with a binary variable which codes the split.

Single Imputation

Imputation strategies use information from the complete data to estimate what a missing value *could be* if it was observed. Single imputation draws a plausible value from a predictive distribution based on available data (Little & Rubin, 2002) to fill in the respective missing value. Since single imputation is employed prior to conducting a conditional inference tree analysis, the same variety of imputation techniques can be employed as for other statistical models. Mean/mode imputation or random replacement are a few simple single imputation techniques. More sophisticated imputation models are typically built on a linear or logistic regression model depending on the nature of the variable with the missing values. However, imputation models have also been built upon partitioning algorithms, such as decision trees and random forest imputation (Tang & Ishwaran, 2017).

Multiple Imputation

As mentioned previously, multiple imputation involves three phases: imputation, analysis, and pooling. Identical to single imputation, the imputation phase can be easily implemented in conditional inference trees since imputation is conducted prior to analysis. However, the analysis phase of multiple imputation requires the same statistical

model fit to each imputed data set, and this is unlikely to happen with conditional inference trees because of its exploratory nature. The analysis phase of multiple imputation in conditional inference trees may result in completely different tree structures (i.e., variables and splitting values). Different tree structures make the pooling phase impossible to implement because of the variable selection and data partitioning components of decision tree algorithms. There are several articles that mention multiple imputation as a possible strategy for treating missing data in decision trees but do not describe the methods for *pooling* results from each imputed tree (García-Laencina, Sancho-Gómez, Figueiras-Vidal, & Verleysen, 2008; Saar-Tsechansky & Provost, 2007). Several researchers employed multiple imputation on missing data by simply ignoring the different tree structure and averaging together predicted values (Feelders, 1999; Twala, 2009), which is a viable strategy when researchers are primarily concerned about prediction. This approach is akin to *bagging* (Breiman, 1996) and will almost always result in better prediction accuracy (Twala, 2009). The problem with averaging predicted values over different trees is that it does not provide a single set of splitting rules or tree structure and makes interpretation challenging.

To my knowledge, little research has been conducted on how to implement the analysis and pooling phase of multiple imputation in conditional inference tree analyses. However, a few studies have employed multiple imputation and compared multiple imputation with other missing data methods (Feelders, 1999; Twala, 2009). These studies are discussed in greater detail in *Comparison Studies* section. Although multiple imputation methods specifically designed for conditional inference trees seem lacking,

decision tree algorithms have been proposed as imputation engine for predicting missing values for other statistical models (Burgette & Reiter, 2010; Van Buuren, 2012).

Comparison Studies

Several studies have compared approaches for treating missing data in decision trees, which can be applied to conditional inference trees (Batista & Monard, 2003; Beaulac & Rosenthal, 2020; Feelders, 1999; Rodgers, Jacobucci, & Grimm, 2021; Twala, 2009). In this section, I briefly summarize across the studies and report which missing data approaches produced the best results.

Across the reviewed studies, the following approaches were compared: listwise deletion, single imputation (*k*-nearest neighbor imputation, EM/logistic imputation, decision tree imputation, distribution-based imputation), mean/mode imputation, multiple imputation with prediction averaging, surrogate splits, and methods that were developed and implemented in other decision tree algorithms (e.g., `C4.5` and `C5.0`). Most studies used complete data sets from the UCI machine learning repository and artificially imposed missing values, while a few studies conducted simulations (Beaulac & Rosenthal, 2020; Rodgers et al., 2021).

The best performing missing data approaches for decision trees were determined based on the findings of all five comparison studies. Multiple imputation outperformed all approaches it was compared against when data were MCAR and MAR (Feelders, 1999; Rodgers et al., 2021; Twala, 2009). However, this approach should be used with caution because prediction accuracy always improves as predicted values are averaged across trees. Additionally, the results from multiple imputation cannot be interpreted as the predicted values were likely produced from different tree structures. To address this

problem, Rodgers et al. (2021) proposed a modified imputation approach specifically for CART that would produce a single tree structure, which allows for interpretation of decision rules. The modified multiple imputation approach outperformed single imputation, surrogate splits, and deletion methods when data were MAR, especially with small sample sizes (Rodgers et al., 2021).

In the remaining studies, the second-best performing approaches to multiple imputation were single imputation approaches that were applied to MAR and MCAR data (Feelders, 1999; Twala, 2009). However, it is important to consider the different single imputation approaches. For example, EM single imputation performed well for numeric variables (Twala, 2009), whereas decision tree single imputation and *k*-nearest neighbor imputation performed best with categorical variables (Twala, 2009; Batista & Monard, 2003). Additionally, surrogate splits performed well when there are high correlations among variables (Twala, 2009). Although the author does not report the magnitudes of the correlations, data sets are available on the University of California, Irvine Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php. It is important to note listwise deletion performed poorly and therefore is not recommended (Twala, 2009). When data were MNAR, the separate class approach was shown to have the best performance (Beaulac & Rosenthal, 2020); however, the way in which missing values were generated in the Beaulac and Rosenthal (2020) study perfectly aligned with the separate class procedure for treating missing data. Any observation with a certain outcome value was recoded as missing, and separate class recodes missing values as its own category. Therefore, more research needs to be done on missing data approaches for treating MNAR data.

In conclusion, the current method of employing multiple imputation (i.e., averaging predicted values over different imputed tree structures) is recommended if a researcher is only interested in prediction accuracy and not interested in interpretability. Multiple imputation and single imputation (specifically EM, decision tree, and *k*-nearest neighbor imputation) are recommended when MAR and MCAR. The surrogate split approach is an appropriate approach to use when variables are highly related. More research should be done to determine the best approach for treating MNAR data.

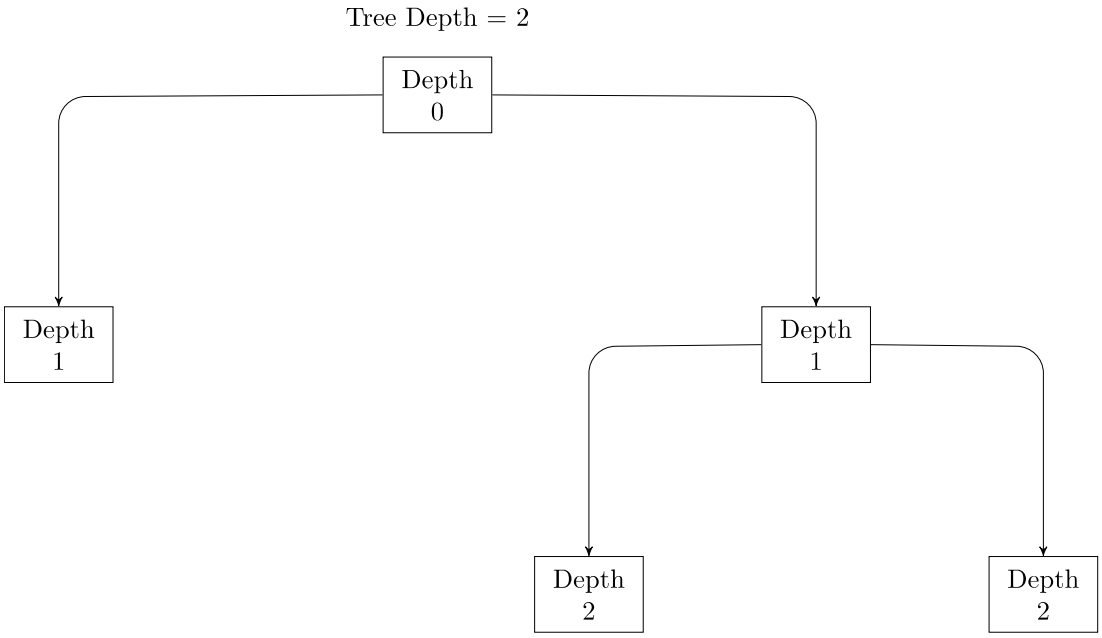
Multiple imputation is generally recommended over other approaches across the various studies; however, the researchers have primarily focused on prediction accuracy of the collection of decision trees as opposed to obtaining a single tree that can be interpreted. Alternatively, Rodgers et al. (2021) proposed a modified multiple imputation procedure that allows for tree interpretability by producing a single tree structure. Instead of averaging predicted values across trees for each imputed set, the modified procedure fits a single tree to all of the multiply imputed datasets at once. That is, the imputed datasets are combined and stacked into a single dataset and evaluated simultaneously. The traditional multiple imputation approach (with prediction averaging) will almost always outperform the modified procedure or any other approach in prediction accuracy because this method is akin to bagging or random forests. However, the modified approach allows for tree interpretability and has been shown to outperform other deletion, single imputation, and surrogate split approaches when data are MAR (Rodgers et al. 2021). Since this approach was initially developed for CART, the purpose of this study is to expand the modified imputation approach to conditional inference trees.

Proposed Approach for Handling Missing Data in Conditional Inference Trees

The purpose of this project is to modify the multiple imputation approach specifically for conditional inference trees. The proposed approach follows the first three steps of multiple imputation (i.e., *impute*, *analyze*, and *pool*); however, the *pooling* step is different. First, data are imputed from a distribution specifically modeled for the missing data. Second, a conditional inference tree is fit to the imputed data, and the depth of the resulting tree is recorded. Tree depth measures the length of the path from the furthest terminal node to the root (example shown in Figure 3).

Figure 3

Tree Depth Example



Third, the first two steps are repeated multiple times (e.g., $\boxed{20}$). Figure 4 depicts a simple example of the first three steps. Fourth, the imputed datasets are stacked to create a single, large data set consisting of $\boxed{m \cdot N}$ rows, where $\boxed{m}$ is the number of imputed datasets and $\boxed{N}$ is the sample size for each imputed dataset. A CTREE is then fit to the

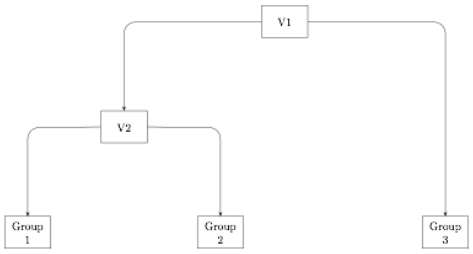
stacked dataset with the tree depth fixed to the average depth (rounded to nearest whole number) obtained when a tree was fit to each imputed dataset. Thus, in this *pooling step*, we pool the tree depth and then use the average depth as the maximum depth when fitting a tree to the stacked data. This leads to a single decision tree that is indirectly determined to the stacked multiply imputed dataset with a single set of decision rules that are easily interpreted (shown in Figure 5).

Figure 4

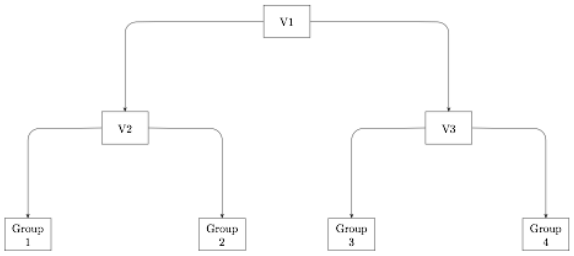
Multiple imputation approach for decision trees

Case	V1	V2	V3
1	1	7	9
2	*NA*	2	2
3	2	*NA*	4
4	6	*NA*	6
5	9	5	*NA*

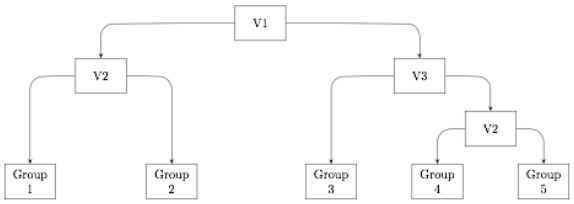
Case	V1	V2	V3
1A	1	7	9
2A	3	2	2
3A	2	6	4
4A	6	4	6
5A	9	5	8



Case	V1	V2	V3
1B	1	7	9
2B	2	2	2
3B	2	5	4
4B	6	5	6
5B	9	5	9



Case	V1	V2	V3
1C	1	7	9
2C	2	2	2
3C	2	4	4
4C	6	5	6
5C	9	5	10



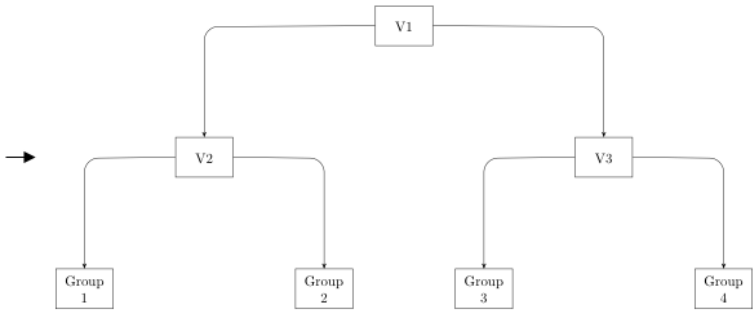
Note. This figure illustrates the imputation and analysis phase of the modified multiple imputation approach for decision trees.

Figure 5

Modified Multiple Imputation Approach for Decision Trees

Case	V1	V2	V3
1	1	7	9
2	*NA*	2	2
3	2	*NA*	4
4	6	*NA*	6
5	9	5	*NA*

Case	V1	V2	V3
1A	1	7	9
2A	3	2	2
3A	2	6	4
4A	6	4	6
5A	9	5	8
1B	1	7	9
2B	2	2	2
3B	2	5	4
4B	6	5	6
5B	9	5	9
1C	1	7	9
2C	2	2	2
3C	2	4	4
4C	6	5	6
5C	9	5	10



Note. This figure illustrates the pooling phase of the modified multiple imputation approach. Multiply imputed datasets are stacked into a single data frame, a decision tree is fit to the stacked dataset, and the decision tree is pruned based on the average tree depth from individual trees.

Fitting the final conditional inference tree to the stacked multiply imputed dataset provides a single set of decision rules, but ignores the variability across imputed datasets. While imputation variability is an important component of the calculation of standard errors in the application of multiple imputation with a theoretically driven statistical model (e.g., multiple regression model), standard errors are not part of decision trees. The splitting values in conditional inference trees are considered point estimates, and conditional inference trees do not provide information the uncertainty of the point estimate.

Pooling the tree depth is an important aspect of the modified multiple imputation approach. We note that the optimal tree depth cannot be determined through statistical significance of the stacked multiply imputed data because sample size is inflated. For example, say we have a dataset with 10% MCAR missingness on ten variables. We conduct $m = 20$ imputations and stack the multiply imputed data. Approximately 35% of the sample will have complete data leading to the same data appearing in the stacked data 20 times. Another ~39% of the sample will be missing one value leading to 90% of their data appearing in the stacked data 20 times. The high degree of the same data appearing in the dataset and inflated sample size will affect the statistically motivated stopping criterion. Thus, using statistical significance with the stacked multiply imputed data leads to an overgrown (overfit) CTREE. Instead, determining tree size based on pooling tree depth leads to more appropriately sized decision trees.

A Monte Carlo simulation study examined the performance of the modified multiple imputation approach outlined above and compared its performance to the missing data methods currently implemented with decision trees in terms of its predictive performance, variable selection, variable importance, and tree size.

CHAPTER 5

METHODS

A Monte Carlo simulation study was conducted to compare how well different missing data approaches perform with conditional inference trees. Data were generated from a population tree structure, missing values were generated following different missing data protocols, conditional inference trees were fit to these datasets using each missing data handling approach, and I examined various indices of the resulting prediction model. This process was repeated 1,000 times for every condition. Baseline measures were taken from complete datasets (i.e., containing no missing values) and used for comparison. The performance of each missing data approach was examined with respect to prediction accuracy, variable selection, and variable importance.

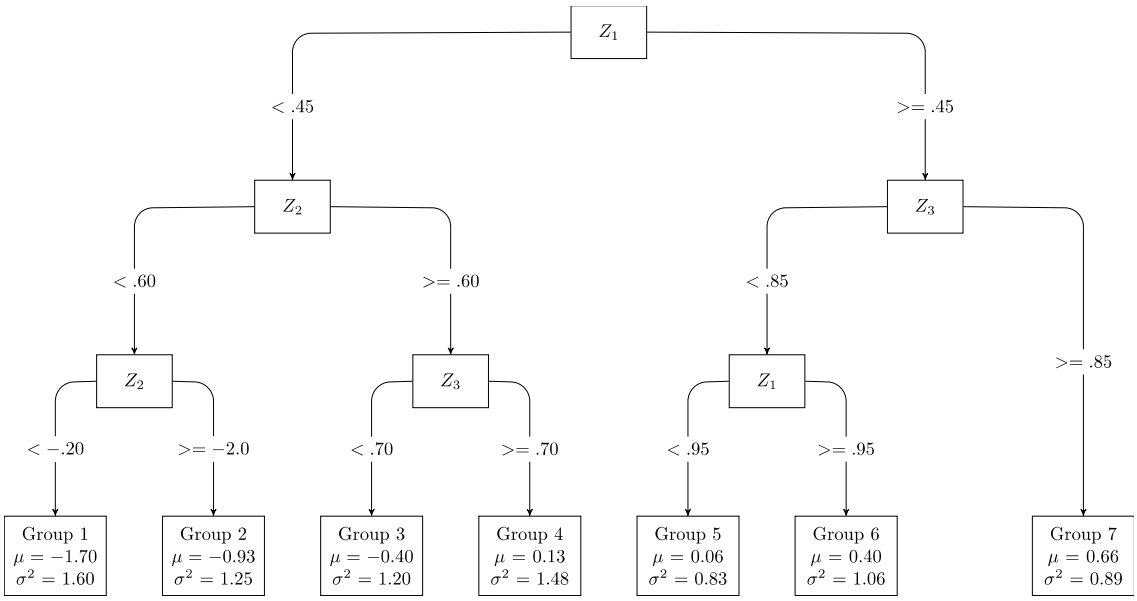
Data Generation

Data were generated using `R` (R Core Team, 2020). All predictor variables were drawn from a standard normal distribution (i.e., $\mu = 0, \sigma = 1$). Depending on the condition, one (x_1) or four (x_1, x_2, x_3, x_4) predictor variables were created. Three predictor variables, z_1, z_2, and z_3, were generated to correlate .4 or .6 with the x variables, and z_1, z_2, and z_3 were subsequently used to generate the outcome using a series of decision rules from a population decision tree. The population tree structure includes six splits and seven terminal nodes. The outcome variable, y, was generated from the population tree shown in Figure 6 with values generated from a normal distribution with the mean and variance reported in each terminal node. Of note, the first split in the population tree is on z_1. Additionally, six distractor predictor variables, z_4 through z_9 were generated from a standard normal distribution and correlated .15 with z_1,

Z_2, and Z_3. Depending on the condition, distractor variables either correlated .02 or .09 with the single predictor X_1 or all four predictors (X_1, X_2, X_3, X_4). The simulated dataset includes 10 or 13 predictor variables (i.e., three used in the population decision tree, one or four used for missing data generation, and six distractors), and the outcome variable.

Figure 6

Population Tree Structure



Manipulated Features

Manipulated features include sample size and characteristics of missing values. The sample sizes include $N = 200$, $N = 500$, and $N = 1{,}000$ to cover a range of sample sizes commonly seen in the social and behavioral sciences. Missing values were imposed across all predictors; however, the nature of the missing values varied for Z_1, which will be the first splitting variable in the population tree structure. Varied aspects included the missing data mechanism, the percentage of missing data, the number of variables that the likelihood of a missing value is dependent on, and the degree of association between likelihood of missingness and the other variable(s) in the dataset. Missing data generation

on all other predictors (all variables not including z_1) were MCAR with a 2.5% likelihood of being recoded as missing.

Missing Data Generation. The method for imposing missing values on variable z_1 closely follows methods from Mazza, Enders, and Ruehlman (2015). Missing values were designed to either be missing at random (MAR) or missing completely at random (MCAR). In the MAR condition, missing values on z_1 were generated to relate to one (x_1) or four variables (x_1, x_2, x_3, and x_4). The association between the likelihood of missingness and the other variable(s) in the dataset were specified using a logistic regression model (Agresti, 2012; Johnson & Albert, 1999; Mazza et al., 2015), with slope and intercept parameters chosen to produce the desired level of association between the underlying missingness probability and the complete variable(s) as well as the overall percentage of missing values. Slopes were selected such that the strength of association between the underlying missingness probability and the complete variable(s) was either R^2 = .2 for a moderate association or R^2 = .4 for a strong association. Intercepts were selected so that percentage of missing values on z_1 will either be 15% or 30%, which are rates commonly found in psychological and educational research (Enders, 2003). The MCAR condition will have fewer manipulated features than the MAR conditions because missingness was unrelated to any other variables in the dataset. Since MCAR occurs when the likelihood of missingness occurs at random, the slope for the logistic regression model was 0 and intercepts were chosen such that the percentage of missing values are either 15% or 30% on z_1.

Approaches for Handling Missing Data. Listwise deletion, delete if selected, majority rule, surrogate splits, single imputation, a multiple imputation with prediction

averaging, and the proposed multiple imputation approach were used to handle the missing data. Listwise deletion was employed by deleting cases with missing values prior to analyses. Delete if selected was applied using the control settings (i.e., `maxsurrogate=0`) from the `ctree` package (Hothorn, Hornik, Zeileis, 2006) in `R` (R Core Team, 2020). The majority rule approach was also employed using the control function by specifying that no surrogates would be used in the analyses (i.e., `majority = TRUE`). The surrogate split approach used the default method (previously described) to place observations with missing values.

For single and multiple imputation, data were imputed using the `mice` package (Buuren & Groothuis-Oudshoorn, 2011) in `R` (R Core Team, 2020). The elementary imputation method was specified using program defaults, which uses predictive mean matching. In the single imputation approach, missing values will be imputed once to create a single dataset (i.e., m = 1), which will then be analyzed. In the multiple imputation approaches, missing values were imputed 20 times (i.e., m = 20). According to Buuren and Groothuis-Oudshoorn (2011), `mice` assumes that the multivariate distribution of an incomplete variable is completely specified by a vector of unknown parameters, θ. Sampling iteratively, the algorithm models the conditional distributions of the incomplete variable given the other variables to obtain a posterior distribution of θ. Using Gibbs sampling, the algorithm selects and fills in plausible values for the missing values on the incomplete variables. The distributions are assumed for each variable instead of the whole dataset. The chained equations within `mice` refers to concatenating univariate procedures to fill in missing data (Buuren & Groothuis-Oudshoorn, 2011).

Stopping Criteria. CTREEs recursively partition data until there is no significant association between predictors and the outcome. Optimal tree sizes are determined by significance tests for listwise deletion, delete if selected, majority rule, surrogate splits, and single imputation. In multiple imputation with prediction averaging, each tree obtained optimal size by significance tests, but the predicted values from each tree was averaged. In the modified multiple imputation approach, the multiply imputed data was stacked and analyzed with the maximum tree depth set to the average depth when a CTREE was fit to each imputed dataset separately.

Evaluation Metrics

Four evaluation metrics were examined to assess and compare the performance of the missing data approaches. The metrics are the mean square error (MSE) in a test dataset, the proportion of replicates where the first splitting variable was $\overline{z_1}$, variable importance metrics, and the median number of splits.

The final decision tree from each missing data approach was used to generate predicted values in the test dataset with N = 10,000 drawn from the same population. The test dataset contained no missing values and was not used to estimate any of the models. The predicted values in the test dataset were calculated and used to determine the MSE. Lower MSE values indicated stronger prediction accuracy, whereas higher MSE values indicated weaker prediction accuracy. The performance of missing data approaches was compared to each other and with the conditional inference tree estimated using the complete data.

The second evaluation metric was the proportion of replicates where $\overline{z_1}$ is the first variable selected to split the data. Recall that variable $\overline{z_1}$ is the first variable split in the

population tree. Thus, the proportion of times z_1 (i.e., the target variable) is correctly selected for the first split indicates the CTREE properly selected the primary splitting variable. The third evaluation metric is variable importance. Variable importance assessed the degree to which each variable contributes the prediction of the outcome. Variable importance was calculated for every predictor by summing together the decrease in error for every split using the variable as the splitting variable. Variable importance values for z_1, z_2, and z_3 were compared across each missing data approach and with the complete data.

The median number of splits was the last evaluation metric. Seven decision trees were fit (i.e., complete data and the six missing data approaches) for each replication within a condition. The median number of splits across all replications within a condition was recorded for each approach. The number of splits were compared across missing data approaches as well as in the population decision tree as an indication of proper tree size.

CHAPTER 6

RESULTS

Overall, simple missing data techniques such as majority rule, treat missing as its own category, and listwise deletion performed better than the proposed multiple imputation approach, single imputation, and surrogate splits. Notably, listwise deletion was highly influenced by sample size (i.e., producing the greatest amount of MSE in small sample size conditions) but correctly selected the first splitting variable more often than all other approaches. The proposed multiple imputation approach (closely followed by single imputation) performed better than surrogate splits when data were MAR with multiple variables strongly predicting missing values and when dealing with small sample sizes. Multiple imputation with prediction averaging had the greatest prediction accuracy but did not produce an interpretable tree structure. The following sections summarize and compare the approaches for each outcome.

Mean Square Error (MSE)

Analyses of variance (ANOVAs) were conducted to assess which simulation conditions (i.e., missing data approach, missing data pattern, sample size, percent of missing values, strength of relationship among predictors, and the number of predictors) had the greatest impact on MSE. On average, MSE was most influenced by sample size ($\sqrt{\eta^2}$ = .88). Other important conditions included the percent of missing values ($\sqrt{\eta^2}$ = .04) and strength of the relationship among predictors ($\sqrt{\eta^2}$ = .04). The method for treating missing data (e.g., deletion, imputation, surrogate splits, etc.) was also influential ($\sqrt{\eta^2}$ = .01).

Each missing data handling approach (e.g., listwise deletion, treating missing as its own category, majority rule, surrogate splits, single imputation, multiple imputation with prediction averaging, and the proposed multiple imputation approach were used to handle the missing data) was compared with the control condition where a CTREE was fit to the complete datasets. All MSEs reported in the following tables and graphs represent the percent increase in MSE over the complete data conditions to allow for direct comparisons. For example, a MSE of zero for a given missing data approach would indicate that the approach performed identical to having no missing data (i.e., the respective control condition with complete data). Comparisons were made across simulation conditions that were shown to have the greatest impact on MSE: sample size (Table 1), percent missing (Table 2), and the strength of the relationship among predictors (Table 3). Missing data patterns were also evaluated across the comparisons.

Missing data approaches generally produced minimal differences in MSE values. Multiple imputation with prediction averaging consistently produced the least MSE, which was likely because it is essentially an ensemble approach like bagging (Breiman, 1996). The average MSE for this approach most closely resembled the results when the CTREE was fit to the complete data (see Figures 7-8). Majority rule and treating missing as its own category led to greater MSE than the multiple imputation approach with prediction averaging but lower MSE than the remaining approaches. Differences between majority rule and treat missing as its own category approaches were minimal (i.e., average MSE differed by .01 at most) and became less apparent in the larger sample size conditions. These three approaches consistently produced the lowest MSE across the conditions (see Tables 1-3).

Table 1. Percent Increase in MSE Across Sample Size

	N = 200			*N* = 500			*N* = 1,000		
	MCAR	MAR (.2)	MAR (.4)	MCAR	MAR (.2)	MAR (.4)	MCAR	MAR (.2)	MAR (.4)
Complete Data	0	0	0	0	0	0	0	0	0
Listwise Deletion	0.08	0.07	0.07	0.03	0.03	0.03	0.02	0.02	0.02
Own Category	0.05	0.04	0.04	0.03	0.02	0.01	0.02	0.01	0.01
Majority Rule	0.04	0.04	0.04	0.03	0.02	0.01	0.02	0.01	0.01
Surrogate Splits	0.05	0.04	0.05	0.03	0.02	0.02	0.02	0.02	0.02
Single Imputation	0.06	0.05	0.05	0.04	0.03	0.03	0.02	0.02	0.02
Multiple Imputation*	0.05	0.04	0.04	0.03	0.03	0.03	0.03	0.03	0.03
Prediction Averaging	0.01	<0.01	<0.01	0.01	<0.01	<0.01	0.01	<0.01	<0.01

(*) indicates the proposed modified multiple imputation approach

Table 2. Percent Increase in MSE Across Percentage of Missing Values

	15%			30%		
	MCAR	MAR (.2)	MAR (.4)	MCAR	MAR (.2)	MAR (.4)
Complete Data	0	0	0	0	0	0
Listwise Deletion	0.03	0.03	0.03	0.05	0.05	0.05
Own Category	0.02	0.01	0.01	0.05	0.03	0.03
Majority Rule	0.02	0.01	0.01	0.05	0.03	0.03
Surrogate Splits	0.02	0.01	0.02	0.05	0.04	0.04
Single Imputation	0.02	0.02	0.02	0.06	0.05	0.05
Multiple Imputation*	0.02	0.02	0.02	0.05	0.04	0.05
Prediction Averaging	<0.01	-0.01	-0.01	0.02	0.01	0.01

(*) indicates the proposed modified multiple imputation approach

Table 3. Percent Increase in MSE Across Relationship Among Predictors

	r = .16			*r* = .36		
	MCAR	MAR (.2)	MAR (.4)	MCAR	MAR (.2)	MAR (.4)
Complete Data	0	0	0	0	0	0
Listwise Deletion	0.04	0.04	0.04	0.04	0.04	0.04
Own Category	0.03	0.02	0.02	0.04	0.02	0.02
Majority Rule	0.03	0.02	0.02	0.04	0.02	0.02
Surrogate Splits	0.03	0.03	0.03	0.04	0.03	0.03
Single Imputation	0.04	0.04	0.04	0.04	0.03	0.03
Multiple Imputation*	0.04	0.03	0.04	0.04	0.03	0.03
Prediction Averaging	0.01	0.01	0.01	0.01	<0.01	<0.01

(*) indicates the proposed modified multiple imputation approach

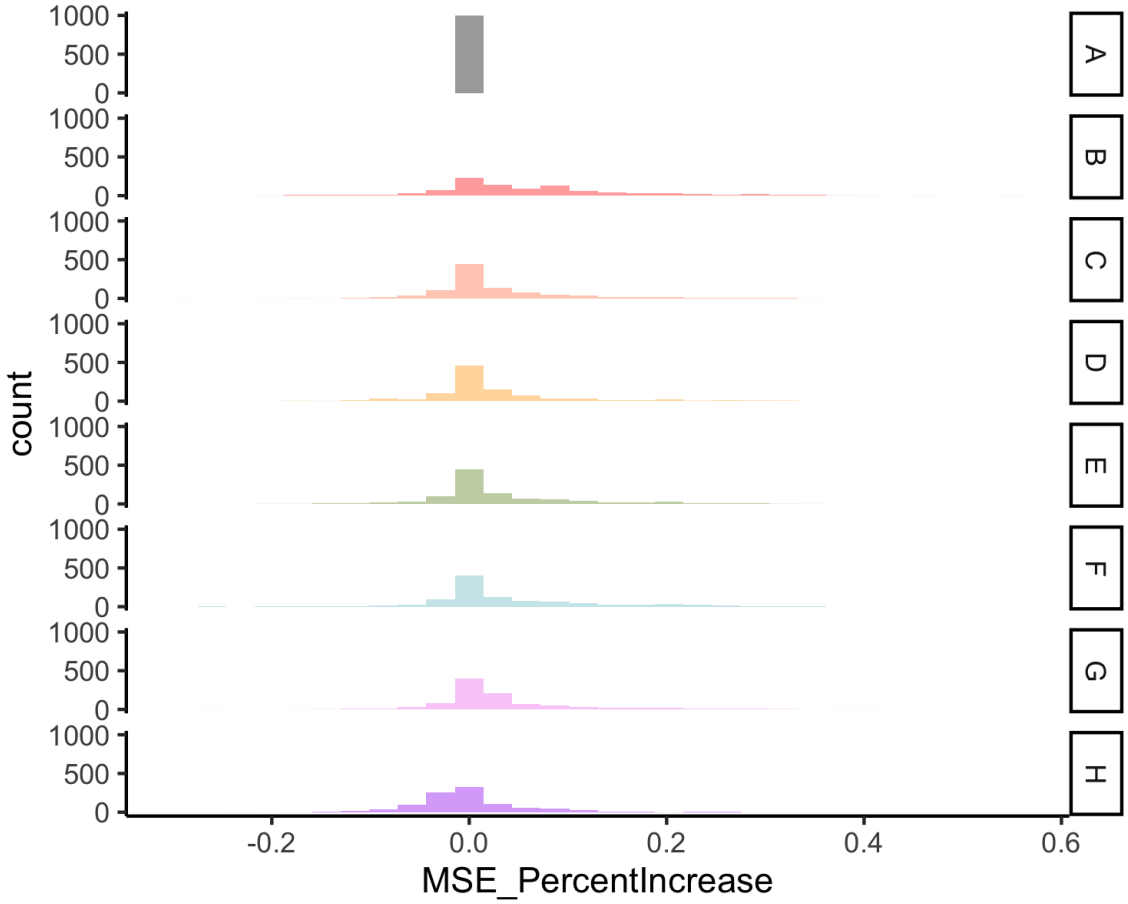
The remaining approaches (the proposed multiple imputation approach, single imputation, surrogate splits, and listwise deletion) produced greater MSE than the multiple imputation approach with prediction averaging, majority rule, and treat missing as its own category. Out of the remaining approaches, surrogate splits performed slightly better than the imputation approaches across most conditions. Surrogate splits produced

lower MSE in conditions with larger sample sizes (Table 1), weak relationships among predictors (Table 3), and when data were MCAR. The proposed multiple imputation approach produced lower MSE in conditions with small sample sizes (i.e., when *N* = 200; Table 1), and performed as well as surrogate splits when there were strong relationships among predictors (Table 3), high percentage of missing values (Table 2), and data that were MAR. Single imputation produced slightly greater MSE than the proposed multiple imputation approach and surrogate splits in small sample size and MCAR conditions but performed well in larger sample size conditions (Table 1). Listwise deletion had the greatest MSE in small sample size conditions but performed fairly well in large sample size conditions (Table 1). In fact, listwise deletion performed similar to surrogate splits and single imputation, as well as outperformed the proposed multiple imputation approach when *N* = 1,000.

MSE in Extreme Simulation Conditions. MSE values for each missing data approach were compared across extreme simulation conditions. The least severe condition in regards to missingness had 15% percent missing values on predictor z_1 that were MCAR and a weaker relationship among predictors ($r = 0.16$). The most severe missingness condition had 30% percent missing values on predictor z_1, four predictors that were more strongly related to missing values ($R^2 = 0.4$) in the MAR condition, and a relatively stronger relationship among the predictors ($r = 0.36$). Since sample size had the greatest impact on MSE, the least and most severe missingness conditions were compared across the same sample size (i.e., $N = 200$).

Figure 7

Mean Square Error in the Least Severe Missingness Condition



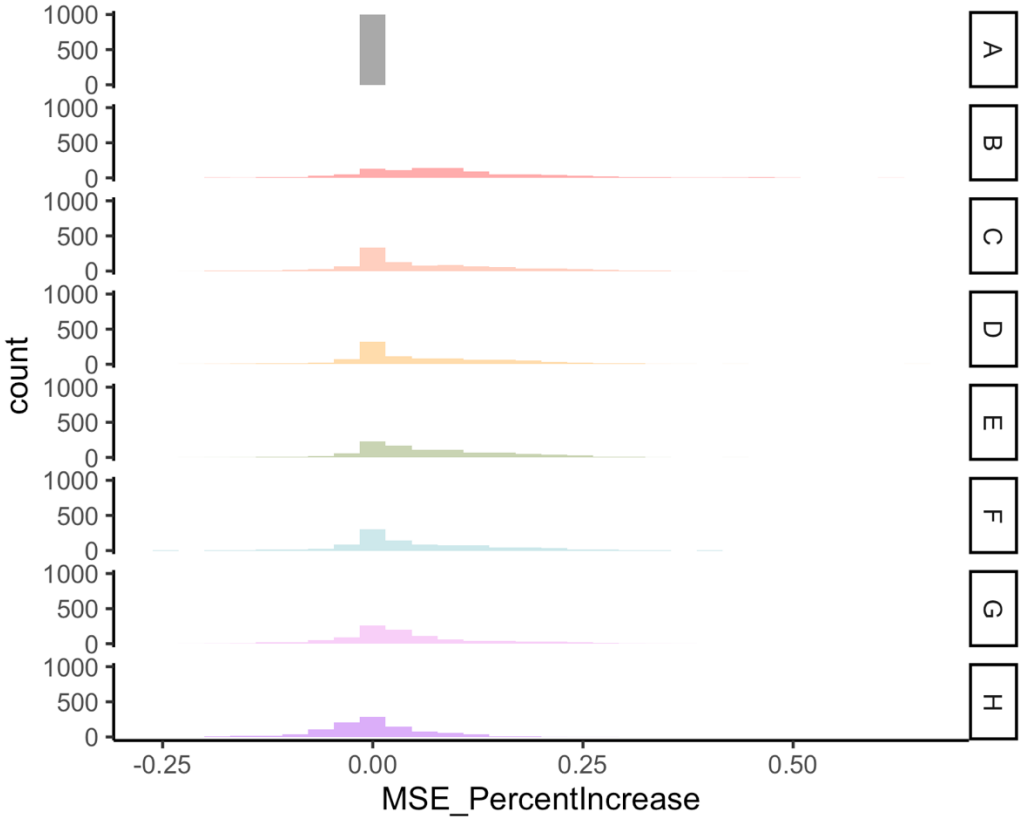
Note. MSE produced across each missing data approach in the least severe missingness condition where 15% of the data on $\overline{z_1}$ were MCAR, the predictors were correlated .16,and N = 200. Missing data approaches include: (A) Baseline - Complete Data; (B) Listwise Deletion; (C) Treat Missing as Own Category; (D) Majority Rule; (E) Surrogate Splits; (F) Single Imputation; (G) Proposed Multiple Imputation Approach; (H) Multiple Imputation with Prediction Averaging.

Across the missing data approaches, MSE values in the least severe missingness condition are illustrated in Figure 7. As expected, the multiple imputation approach with prediction averaging (H) outperformed all the approaches and most closely resembled the

complete data condition (A). The amount of MSE produced by the remaining missing data approaches was fairly similar. Specifically, treating missing as its own category (C), majority rule (D), surrogate splits (E), single imputation (F), and the proposed multiple imputation approach (G) had similar distributions of MSE. The distribution of MSE values in the listwise deletion approach (B) had more spread and greater right skew than the other approaches illustrating larger percent increase in MSE over the complete data condition.

Figure 8

Mean Square Error in the Most Severe Missingness Condition



Note. MSE produced across each missing data approach in the most severe missingness condition with 30% of the data on z_1 were MAR with a multiple variables predicting missing values (R^2 = .4), predictors were correlated .36, and *N* = 200. Missing data approaches include: (A) Baseline - Complete Data; (B) Listwise Deletion; (C) Treat Missing as Own Category; (D) Majority Rule; (E) Surrogate Splits; (F) Single Imputation; (G) Proposed Multiple Imputation Approach; (H) Multiple Imputation with Prediction Averaging.

Histograms of MSE values from the most severe missingness condition are shown in Figure 8. Again, the multiple imputation approach with prediction averaging (H) had a small spread in the distribution of MSEs centered around zero, indicating it performed similar to the complete data condition. most closely resembled the complete data condition (A). The proposed multiple imputation approach (G) performed better than the remaining approaches. The distribution of MSEs peaked around zero with fewer scores in the right skew tail. Single imputation, treating missing as its own category, majority rule, and surrogate splits had a wider spread across MSE values in comparison with the proposed approach. Listwise deletion had the longest right skew tail indicating that it had the greatest percent increase in MSE.

Table 4. Percent Increase in MSE Across Severe Missingness Conditions

	Least Severe	Most Severe
(A) Complete Data	0	0
(B) Listwise Deletion	0.06	0.09
(C) Own Category	0.03	0.05
(D) Majority Rule	0.02	0.05
(E) Surrogate Splits	0.03	0.06
(F) Single Imputation	0.04	0.05
(G) Multiple Imputation*	0.03	0.04
(H) Prediction Averaging	-0.01	-0.01

(*) indicates the proposed modified multiple imputation approach

Number of Splits

The number of splits in each tree was recorded. Table 5 summarizes the number of splits found across each missing data approach. Multiple imputation with prediction averaging did not produce a single tree structure, so the number of splits was not recorded.

Trees produced in the proposed multiple imputation approach had large differences in number of splits. Though this approach seemed to generally follow the other approaches on average (it also had a median of three splits), there were situations when the proposed approach grossly overfit and produced trees with as many as 31 splits. To further investigate which conditions contributed to overfitting, all trees that contained more than 13 splits were evaluated. The pattern indicated that the proposed multiple imputation tended to overfit when dealing with large sample sizes, such as when $\overline{N}$ = 1,000 (see Table 6). Each approach had a total of 20,000 replications within each sample size condition. Out of the 20,000 replications, the proposed approach produced decision trees with more than 13 splits 13 times when N = 200, 221 times when N = 500, and 2,287 times when N = 1,000. Overfit trees in the large sample size conditions were likely contributing to high MSE in large sample size conditions. The average MSE in the proposed approach improved by 0.01 in the N = 1,000 condition when all replications with more than 13 splits are excluded. While having more than 13 splits was a clear indication of overfitting in a population with six splits, it is likely that there was overfitting in the remaining replications with less than the arbitrary cut off of 13 splits.

Table 5. Number of Splits

	MCAR				MAR (.2)				MAR (.4)			
	median	mean	min	max	median	mean	min	max	median	mean	min	max
Complete Data	3	2.91	0	9	3	2.86	1	9	3	2.87	0	8
Listwise Deletion	2	2.36	0	9	2	2.31	0	8	2	2.32	0	8
Own Category	3	3.25	0	10	3	2.93	0	9	3	2.94	0	10
Majority Rule	3	3.23	0	11	3	2.96	0	13	3	2.99	0	10
Surrogate Splits	3	3.00	0	10	3	3.17	0	11	3	3.36	0	12
Single Imputation	3	3.05	0	10	3	3.11	0	9	3	3.22	0	9
Multiple Imputation*	3	4.85	0	31	3	4.74	0	31	3	4.92	0	31
Prediction Averaging	-	-	-	-	-	-	-	-	-	-	-	-

(*) indicates the proposed modified multiple imputation approach

Table 6. Prevalence of Overfitted Models Produced by the Proposed Approach

	Number of Splits $\geq$ 13		Number of Splits < 13	
	Frequency	Average MSE	Frequency	Average MSE
N = 200	13	0.21	19987	0.04
N = 500	222	0.11	19778	0.03
N = 1,000	2294	0.07	17706	0.02

Note. The population tree structure had a total of six splits. Trees that contained 13 or more splits were considered overfitted models.

While the proposed approach had a problem overfitting, listwise deletion seemed to have a problem underfitting more often than the other approaches and averaged two splits. Overall, all approaches except the proposed multiple imputation and listwise deletion produced relatively similar tree structures in terms of average number of splits.

Proportion of Correct First Splits

The proportion of times that $\overline{z_1}$ was chosen for the first split was recorded. Figure 9 illustrates the performances of each approach in the least severe and most severe missingness conditions. Across all approaches, higher rates of missing values and smaller sample sizes led to fewer instances where $\overline{z_1}$ was chosen for the first split.

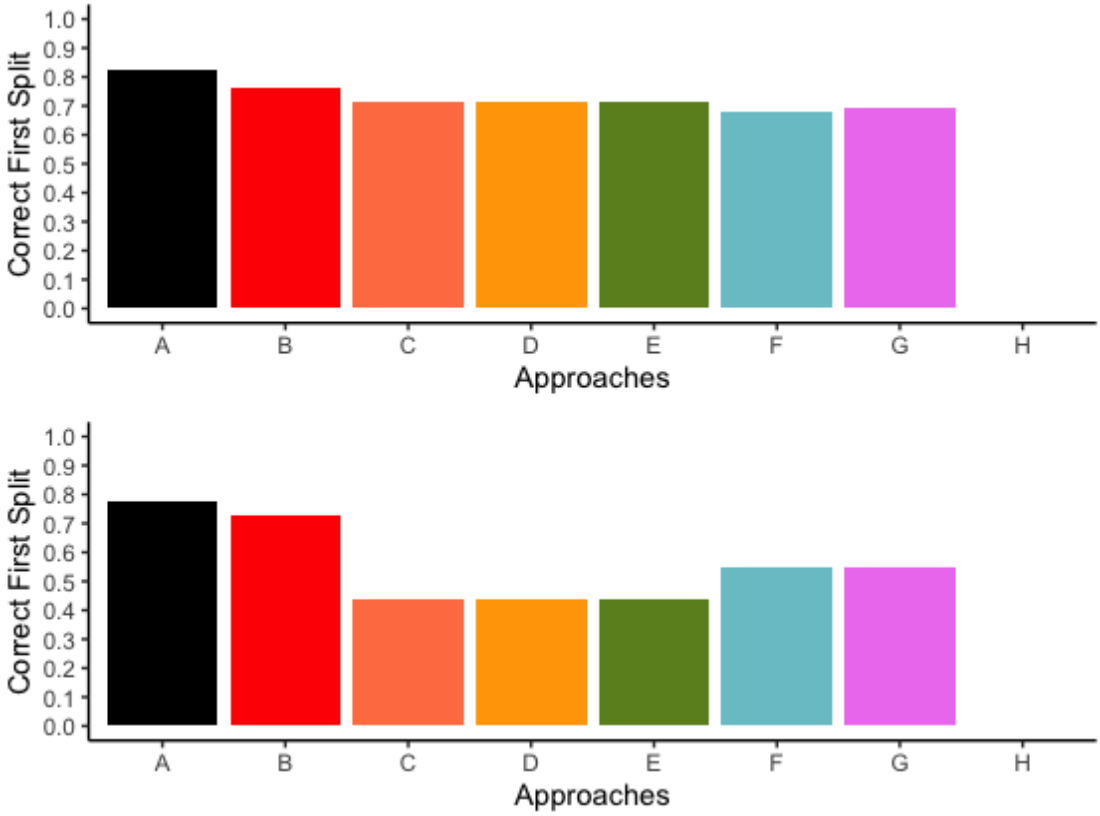
Listwise deletion correctly selected first split more frequently than the other approaches and most closely resembled the complete data conditions. The performance of the other approaches depended on the missing data pattern, strength of association among predictors and missing values, and the percentage of missing data. All approaches were compared with the complete data condition where the correct first split was made 82% of the time. In the least severe simulation condition, listwise deletion selected $\overline{z_1}$ most frequently, which was 76% of the time. Majority rule, treat missing as its own category, and surrogate splits selected $\overline{z_1}$ for the first split 72% of the time. The proposed multiple imputation approach correctly selected $\overline{z_1}$ for the first split 69% of the time, and single imputation did so 68% of the time.

However, the pattern of results switched in the most severe missing data condition, where there was 30% missingness on $\overline{z_1}$, four predictors that were more strongly related to missing values, and a stronger relationship among the predictors.

Again, listwise deletion outperformed the other approaches and selected z_1 for the first split roughly 73% of the time, which most closely resembled the complete data condition (78%). The proposed multiple imputation approach and single imputation correctly selected z_1 for the first split 55% of the time, whereas majority rule, treat missing as its own category, and surrogate splits only selected the correct first split 44% of the time.

Figure 9

Proportion of Correct First Splits: Severe Missingness Conditions



Note. The first panel (top) represents the least severe missingness condition where 15% of the data on z_1 were MCAR, the predictors were correlated .16, and N = 200. In the second panel (bottom), 30% of the data on z_1 were MAR with a multiple variables predicting missing values (R^2 = .4), predictors were correlated .36, and N = 200. Missing data approaches include: (A) Baseline - Complete Data; (B) Listwise Deletion; (C) Treat Missing as Own Category; (D) Majority Rule; (E) Surrogate Splits; (F) Single Imputation; (G) Proposed Multiple Imputation Approach; (H) Multiple Imputation with Prediction Averaging.

Variable Importance

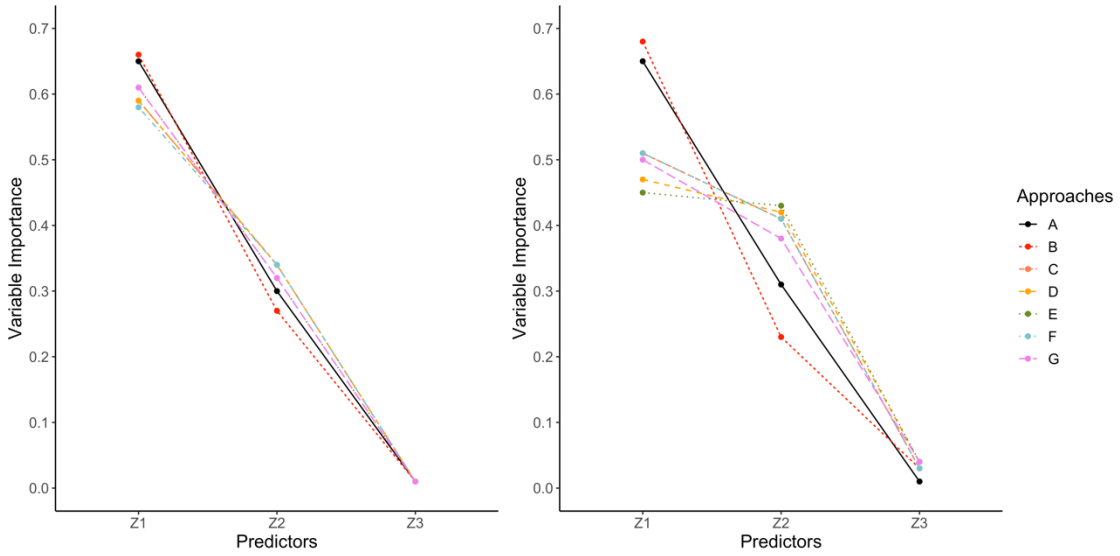
Variable importance values ranged from 0 to 1 for z_1, z_2, and z_3. Recall that z_1 was the target variable containing missing values and it was the first splitting variable, which is often associated with the greatest importance value. A rank order correlation was computed to evaluate whether approaches were in agreement with order of predictor importance. A rank order correlation of 1.00 was found across each pair of approaches, which suggested that all approaches placed the exact same order of predictor importance – with z_1 having the greatest importance value and z3 least importance value. However, the degree to which approaches placed importance on each of the predictors varied. Average importance values in the extreme missingness conditions are illustrated in Figure 10.

In the least severe missingness condition, each missing data approach was compared with the complete data condition (A) which produced the following importance values: 0.65 for z_1, 0.30 for z_2, and 0.01 for z_3. Listwise deletion (B) most closely resembled complete data (A) because, on average, it had only slightly higher importance placed on z_1 (0.66) and slightly lower importance placed on z_2 (0.27). Following listwise deletion, the next best performance was the proposed multiple imputation approach (G) and surrogate splits (E), which produced identical importance values. In comparison with the complete data condition, these approaches assigned a slightly lower importance value on z_1 (0.61) and a slightly higher importance on z_2 (0.32). This pattern of having lower importance values for z_1 and higher importance for on z_2, was also found in the remaining approaches (C, D, and F) with slightly larger differences with

complete data condition (i.e., importance values were either 0.58 or 0.59 for $\overline{z_1}$ and 0.34 for $\overline{z_2}$). Overall, there were minimal differences in variables importance across missing data approaches in the least severe condition.

Figure 10

Variable Importance: Severe Missingness Conditions



Note. The first panel (left) represents a condition where 15% of the data on $\overline{z_1}$ were MCAR, the predictors were correlated .16, and N = 200. In the second panel (right), 30% of the data on $\overline{z_1}$ were MAR with a multiple variables predicting missing values ($\overline{R^2}$ = .4), predictors were correlated .36, and N = 200. Missing data approaches include: (A) Baseline - Complete Data; (B) Listwise Deletion; (C) Treat Missing as Own Category; (D) Majority Rule; (E) Surrogate Splits; (F) Single Imputation; (G) Proposed Multiple Imputation Approach; (H) Multiple Imputation with Prediction Averaging.

The variable importance values produced in the most severe missingness condition are also shown in Figure 10. Again, each missing data approach was compared with the complete data. Listwise deletion (B) most closely resembled the complete data (A) and was the only approach that overestimated the importance of $\overline{z_1}$ (0.68) and underestimated the importance of $\overline{z_2}$ (0.23). The remaining approaches had a reverse pattern (i.e., underestimated the importance of $\overline{z_1}$ and overestimated the importance of $\overline{z_2}$

). Every approach (including listwise deletion) overestimated the importance of z_3 with values equal to 0.03 or 0.04. Out of the remaining approaches, the next best was the proposed approach (G) closely followed by single imputation (F) and treat missing as its own category (C). The proposed approach produced an importance value of 0.50 for z_1 and 0.38 for z_2, whereas the other approaches had an importance value of 0.51 for z_1 and 0.41 for z_2. The worst performing approaches were majority rule (D) and surrogate splits (E), which indicated that z_1 and z_2 had equal importance. Majority rule (D) and surrogate splits (E) produced importance values of 0.47 and 0.45 for z_1 and 0.41 and 0.42 for z_2 respectively.

CHAPTER 7

DISCUSSION

A modified multiple imputation approach was proposed for handling missing data in CTREEs. The proposed approach involved four steps: (1) Impute missing values, (2) Fit a decision tree to the imputed dataset and retain the tree depth, (3) Repeat steps 1 and 2 multiple times, and (4) Stack all imputed datasets into a single data frame and fit a CTREE to the stacked dataset with the tree depth set to the average tree depth from each imputed dataset. A simulation was conducted to compare the proposed approach to listwise deletion, treat missing as its own category, majority rule, surrogate splits, single imputation, and multiple imputation with prediction averaging under multiple MAR and MCAR conditions.

Simulation results revealed that simple techniques, such as majority rule, treat missing as its own category, and listwise deletion were effective approaches for handling missing data in CTREEs. Specifically, majority rule and treat missing as its own category generally produced lower MSE than the other approaches and generated reasonably sized trees. These approaches correctly selected the first variable for splitting when missingness was less severe (i.e., not related to predictors and occurred less frequently). However, majority rule and treat missing as its own category approaches had trouble selecting the correct variable for the first split when there was a high percentage of missing values that were strongly related to the predictors. Any differences between majority rule and treat missing as its own category were minimal and often negligible. Listwise deletion performed nearly as well as these approaches in terms of MSE when

dealing with large sample sizes. Listwise deletion also consistently outperformed all other approaches in selecting the correct variable for the first split, which accurately reflected the population and increases confidence in interpreting its tree structure. Though the tree structure was more consistent with the population in terms of variable selection, the trees generated by this approach were often smaller than the population tree structure suggesting the average tree underfit the data. Another problem with listwise deletion was that it produced a greater MSE when working with small sample sizes.

The proposed multiple imputation did not perform very well against simple approaches in most conditions; however, this approach did seem best suited for small sample sizes and extreme missingness situations (i.e., when data contained a higher percent missing values, predictors that were more strongly related to missing values in the MAR condition, and there was a relatively stronger relationship among the predictors). In these instances, the proposed multiple imputation produced lower MSE and correctly selected the variable for first split. The setback to this approach is that it had issues with overfitting in the large sample size conditions.

Surrogate splits had similar performance to the proposed multiple imputation approach but produced lower MSE when there was a weak relationship among predictors and large sample sizes. It did not select the correct first variable for splitting in the extreme missingness conditions, but seemed to produce reasonably sized trees across the simulation conditions. Single imputation was comparable to the proposed multiple imputation approach; however, single imputation produced greater MSE when the data were MCAR, when there were weak relationships among the predictors, and a high

percentage of missing values. Single imputation performed similarly to the proposed multiple imputation approach in selecting the correct first variable split and generated reasonably sized trees.

Recommendations

The following recommendations are based on current simulation results. Multiple imputation with prediction averaging is recommended when the researcher is only concerned with prediction accuracy and has no interest in interpreting the tree's structure. Listwise deletion is recommended the sample size is large. When dealing with sample sizes of 500 or greater, listwise deletion had adequate prediction accuracy and its variable selection was shown to most closely aligned with population structure. Also, this approach is easy to implement. When dealing with small sample sizes, either majority rule or treat missing as its own category is recommended. The proposed multiple imputation is not recommended based on the current simulation results, but the implementation of the approach as this approach has limitations, which are described in detail below. Perhaps future research will address the limitations and modify the proposed approach to improve its effectiveness in treating missing data in CTREEs, particularly in large samples.

Limitations and Future Directions

A limitation of the proposed modified multiple imputation approach is the pooling method. That is, when analyzing the stacked multiply imputed dataset, the tree depth was set to the average depth obtained when each imputed dataset was analyzed separately. The goal of using the average tree depth when analyzing the stacked multiply imputed

dataset was to obtain an appropriate tree size. While this adaptation of the pooling phase in multiple imputation helped with controlling tree size, it ultimately was not effective at preventing overfitting, especially with large sample sizes. Pooling tree depth in the proposed approach became problematic as the average tree depth and sample size increased. As tree depth increased, the variation in the total number of possible splits increased. For example, a tree with a depth of two has a minimum of two splits and a maximum of three splits, whereas a tree with a depth of five has a minimum of five splits and maximum of 31 splits. CTREE uses a statistically motivated stopping criteria (i.e., *p*-values), which is highly influenced by sample size. Stacking multiple imputed datasets inflates sample size by the number of imputations ($m \times N$). Inflated sample sizes led to overfitting because the algorithm was more likely find statistically significant splits on trivially related predictors and max out the possible number of splits in a given tree depth. Therefore, the proposed approach had serious problems with overfitting when sample size was larger and this issue is likely exacerbated when dealing with more complex tree structures.

Since averaging tree depth was unreliable in controlling tree size, future research should consider additional methods to prevent overfitting. For example, a researcher might adjust the *p*-value criterion based on sample size. Consider a situation where a researcher has a sample size of 1,000 and the proposed multiple imputation approach (with m = 20 imputations) produces a stacked, multiply imputed dataset with 20,000 cases. Using an adaptive approach to setting the *p*-value criterion, such as reducing the *p*-value for 20,000 cases that is equivalent to the *p*-value criterion for 1,000 cases, will

likely produce a more appropriately sized trees than using a fixed criterion like $p < .05$. If averaging tree depth and adjusting *p*-values still leads to overfit trees, then perhaps a researcher could also implement a Bonferroni correction.

Another limitation of this study is that the missing data were handled with a single imputation approach. A variety of imputation methods have been developed, which are typically built upon linear or logistic regression models. However, imputation models have also been built upon partitioning algorithms, such as decision trees and random forest imputation (Tang & Ishwaran, 2017), and these imputation approaches were not considered. Lastly, future studies might consider expanding the missing conditions to include higher percentage of missing values, consider missing values on the dependent variable, and include missingness across several variables (instead of primarily focusing on a single predictor such as $\overline{z_1}$).

Conclusion

The purpose of this research was to determine which missing data approaches commonly used in social and behavioral sciences could be applied to the CTREE machine learning algorithm. The first objective of this project was to survey the current literature for different approaches researchers use to handle missing data when working with decision tree algorithms. As a result, popular missing data approaches included: listwise deletion, majority rule, treat missing as its own category, single imputation, *k*-nearest neighbor imputation, mean/mode imputation, EM/logistic imputation, decision tree imputation, distribution-based imputation, multiple imputation, surrogate splits, and methods that were developed and implemented in other decision tree algorithms (e.g.,

C4.5 and C5.0). Generally, these approaches can be categorized as follows: deletion, forced partitioning (e.g., majority rule or treat missing as its own category), imputation, surrogate splits, or other approaches designed for specific algorithms outside the scope of this project. Missing data approaches like deletion and imputation are commonly used in social and behavioral sciences, whereas approaches like surrogate splits were developed specifically for decision trees. The second objective of this project was to propose modifying the pooling procedure from traditional multiple imputation so that this approach would produce a tree with a single set of decision splits and values. Multiple imputation is commonly used in theoretically driven statistical models; however, the current method for implementing this approach in the machine learning framework does not allow for tree interpretability. It seems social and behavioral researchers would be interested in interpreting decision rules in a tree model. Therefore, the proposed modification aimed to improve the application of multiple imputation in decision trees for researchers who prefer obtaining an interpretable tree structure over purely maximizing outcome prediction.

A motivation for this project was to develop an understanding of the classic missing data problem from an interdisciplinary standpoint. Most missing data approaches found in the machine learning literature review have been evaluated and implemented in computer science and related fields. To my knowledge, it is common practice for the methodologists in these fields to use complete data sets (mostly from the UCI machine learning repository) and artificially impose missing values to evaluate missing data approaches. Methodologists in psychology often conduct simulation studies to evaluate

and compare statistical methods. It seems that from a methodologist standpoint there may be some benefits to employing simulations in addition to the current methods using empirical data. The primary benefit of conducting simulations is having control over the population tree structure and relationships among the variables. Due to the nature of decision trees, which repeatedly mine for patterns in the data, there are instances when identical complete data sets can produce slightly different trees that vary in selected variables and splitting values. This could be problematic when only using empirical data sets as it is unknown whether variation in trees from complete and treated data are due to random variation in the algorithm or the actual treatment of missing data. Knowing the true generating tree structure allows the researcher to exercise more control by determining how well a missing data approach recovers the generating tree structure relative to the complete data sets. Therefore, the final object of this project was to conduct a simulation to evaluate and compare performance across approaches.

Machine learning methods are becoming increasingly popular in psychology where missing data is pervasive and must be addressed. However, the way in which missing data are typically handled with machine learning algorithms has been under-researched. Though some approaches like deletion and imputation are implemented across disciplines, there has been relatively little work done to compare their effectiveness, especially via simulation. Furthermore, the approaches that have been adopted in machine learning framework might not appeal to psychological researchers (e.g., deletion could reduce sample size and traditional multiple imputation does not produce an interpretable tree structure). The current study sought to bridge this gap and

provide more relevant recommendations within the context of psychological data. As machine learning becomes a more common tool in psychology and related fields, more work must be done to better adapt these approaches for situations that are typical in the social and behavioral sciences.

REFERENCES

Agresti, A. (2012). *Categorical Data Analysis* (3rd Edition). Wiley.

Allison, P. (2002). *Missing Data*. SAGE Publications, Inc. https://doi.org/10.4135/9781412985079

Azur, M. J., Stuart, E. A., Frangakis, C., & Leaf, P. J. (2011). Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*, *20*(1), 40–49. https://doi.org/10.1002/mpr.329

Baraldi, A. N., & Enders, C. K. (2010). An introduction to modern missing data analyses. *Journal of School Psychology*, *48*(1), 5–37. https://doi.org/10.1016/j.jsp.2009.10.001

Barros, R. C., Basgalupp, M. P., de Carvalho, A. C., & Freitas, A. A. (2012). A Survey of Evolutionary Algorithms for Decision-Tree Induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(3), 291–312. https://doi.org/10.1109/TSMCC.2011.2157494

Batista, G., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, *17*(5–6), 519–533. https://doi.org/10.1080/713827181

Beaulac, C., & Rosenthal, J. S. (2020). BEST: A decision tree algorithm that handles missing values. *Computational Statistics*, *35*(3), 1001–1026.

Becker, R. A., Chambers, J. M., & Wilks, A. R. (2017). *The New S Language: A Programming Environment for Data Analysis and Graphics*. Chapman and Hall/CRC. https://doi.org/10.1201/9781351074988

Berk, R. A. (2008). Classification and Regression Trees (CART). In *Statistical Learning from a Regression Perspective* (pp. 1–65). Springer. https://doi.org/10.1007/978-0-387-77501-2_3

Blake, C., Keogh, E. and Merz, C.J. (1998) *UCI Repository of machine learning databases* [http://www.ics.uci.edu/≈mlearn/MLRepository.htmll. Irvine, CA: University of California, Department of Information and Computer Science.

Brandmaier, A. M., von Oertzen, T., McArdle, J. J., & Lindenberger, U. (2013). Structural equation model trees. *Psychological Methods*, *18*(1), 71–86. https://doi.org/10.1037/a0030001

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123-140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. Taylor & Francis.

Buck, S. F. (1960). A Method of Estimation of Missing Values in Multivariate Data Suitable for use with an Electronic Computer. *Journal of the Royal Statistical Society. Series B (Methodological)*, *22*(2), 302–306.

Burgette, L. F., & Reiter, J. P. (2010). Multiple imputation for missing data via sequential regression trees. *American Journal of Epidemiology*, *172*(9), 1070–1076. https://doi.org/10.1093/aje/kwq260

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). *An Overview Of Machine Learning*.

Chollet, F., & Allaire, J. J. (2018). *Deep Learning with R*. Manning Publications.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*(4), 261–283. https://doi.org/10.1007/BF00116835

Dua, D. & Graff, C. (2019). *UCI Repository of machine learning databases* [http://www.ics.uci.edu/≈mlearn/MLRepository.htmll. Irvine, CA: University of California, Department of Information and Computer Science.

Enders, C., Dietz, S., Montague, M., & Dixon, J. (2006). Modern Alternatives for Dealing with Missing Data in Special Education Research. In T. E. Scruggs & M. A. Mastropieri (Eds.), *Applications of Research Methodology* (Vol. 19, pp. 101–129). Emerald Group Publishing Limited. https://doi.org/10.1016/S0735-004X(06)19005-9

Enders, C. K. (2010). *Applied Missing Data Analysis*. Guilford Press.

Farhangfar, A., Kurgan, L., & Dy, J. (2008). Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, *41*(12), 3692–3705. https://doi.org/10.1016/j.patcog.2008.05.019

Feelders, A. (1999). Handling Missing Data in Trees: Surrogate Splits or Statistical Imputation? In J. M. Żytkow & J. Rauch (Eds.), *Principles of Data Mining and Knowledge Discovery* (Vol. 1704, pp. 329–334). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-48247-5_38

García-Laencina, P. J., Sancho-Gómez, J.-L., & Figueiras-Vidal, A. R. (2010). Pattern classification with missing data: A review. *Neural Computing and Applications*, *19*(2), 263–282. https://doi.org/10.1007/s00521-009-0295-6

García-Laencina, P. J., Sancho-Gómez, J.-L., Figueiras-Vidal, A. R., & Verleysen, M. (2009). K nearest neighbours with mutual information for simultaneous classification and missing data imputation. *Neurocomputing*, *72*(7), 1483–1493. https://doi.org/10.1016/j.neucom.2008.11.026

Gleason, T. C., & Staelin, R. (1975). A Proposal for Handling Missing Data. *Psychometrika*.

Gonzalez, O., O'Rourke, H. P., Wurpts, I. C., & Grimm, K. J. (2018). Analyzing Monte Carlo Simulation Studies With Classification and Regression Trees. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(3), 403–413. https://doi.org/10.1080/10705511.2017.1369353

Graham, J. W. (2012). *Missing Data: Analysis and Design*. Springer-Verlag. https://doi.org/10.1007/978-1-4614-4018-5

Grimm, K. J., & Jacobucci, R. (2020). Reliable Trees: Reliability Informed Recursive Partitioning for Psychological Data. *Multivariate Behavioral Research*, *0*(0), 1–13. https://doi.org/10.1080/00273171.2020.1751028

Grimm, K. J., Mazza, G. L., & Davoudzadeh, P. (2017). Model selection in finite mixture models: A k-fold cross-validation approach. *Structural Equation Modeling*, *24*(2), 246–256. https://doi.org/10.1080/10705511.2016.1250638

Groves, R. M. (2006). "*Nonresponse Rates and Nonresponse Bias in Household Surveys." Public Opinion Quarterly 70(5),* 646-675.

Groves, R. M., & Peytcheva, E. (2008). The Impact of Nonresponse Rates on Nonresponse Bias: A Meta-Analysis. *Public Opinion Quarterly*, *72*(2), 167–189. https://doi.org/10.1093/poq/nfn011

Grubinger, T., Zeileis, A., & Pfeiffer, K. P. (2014). evtree: Evolutionary Learning of Globally Optimal Classification and Regression Trees in R. *Journal of Statistical Software*, *61*, 1–29. https://doi.org/10.18637/jss.v061.i01

Hajjem, A., Bellavance, F., & Larocque, D. (2011). Mixed effects regression trees for clustered data. *Statistics & Probability Letters*, *81*(4), 451–459. https://doi.org/10.1016/j.spl.2010.12.003

Hothorn, T., Hornik, K., & Zeileis, A. (2006). Unbiased Recursive Partitioning: A Conditional Inference Framework. *Journal of Computational and Graphical Statistics*, *15*(3), 651–674. https://doi.org/10.1198/106186006X133933

Jackman, S. (2000). Estimation and Inference via Bayesian Simulation: An Introduction to Markov Chain Monte Carlo. *American Journal of Political Science*, *44*(2), 375–404. https://doi.org/10.2307/2669318

Jacobucci, R., Grimm, K. J., & McArdle, J. J. (2016). Regularized structural equation modeling. *Structural Equation Modeling*, *23*(4), 555–566. https://doi.org/10.1080/10705511.2016.1154793

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R*. Springer-Verlag. https://doi.org/10.1007/978-1-4614-7138-7

Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. *Applied Statistics*, *29*(2), 119. https://doi.org/10.2307/2986296

Khedr, A. E., Idrees, A. M., & Seddawy, A. I. E. (2016). Enhancing Iterative Dichotomiser 3 algorithm for classification decision tree. *WIREs Data Mining and Knowledge Discovery*, *6*(2), 70–79. https://doi.org/10.1002/widm.1177

Khosravi, P., Vergari, A., Choi, Y., Liang, Y., & Broeck, G. V. den. (2020). Handling Missing Data in Decision Trees: A Probabilistic Approach. *ArXiv:2006.16341 [Cs, Stat]*. http://arxiv.org/abs/2006.16341

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, *31*(3), 249–269.

Kotsiantis, S. B. (2013). Decision trees: A recent overview. *Artificial Intelligence Review*, *39*(4), 261–283. https://doi.org/10.1007/s10462-011-9272-4

Kromrey, J. D., & Hines, C. V. (1994). Nonrandomly Missing Data in Multiple Regression: An Empirical Comparison of Common Missing-Data Treatments. *Educational and Psychological Measurement*, *54*(3), 573–593. https://doi.org/10.1177/0013164494054003001

Lavori, P. W., Dawson, R., & Shera, D. (1995). A multiple imputation strategy for clinical trials with truncation of patient data. *Statistics in Medicine*, *14*(17), 1913–1925. https://doi.org/10.1002/sim.4780141707

Lavrakas, P. (2008). *Encyclopedia of Survey Research Methods*. https://doi.org/10.4135/9781412963947

Little, R. J. A. (1988). A Test of Missing Completely at Random for Multivariate Data with Missing Values. *Journal of the American Statistical Association*, *83*(404), 1198–1202. https://doi.org/10.1080/01621459.1988.10478722

Little, R. J. A., & Beale, E. M. (1975). Missing values in Multivariate Analysis. *Journal of the Royal Statistical Society: Series B (Methodological), 37*(1), 129-145. https://doi.org/10.1111/j.2517-6161.1975.tb01037.x

Little, R. J. A., & Rubin, D. B. (2002). *Statistical Analysis with Missing Data* (2nd Edition). Wiley-Interscience.

Loh, W.-Y. (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, *1*(1), 14–23. https://doi.org/10.1002/widm.8

Loh, W.Y., & Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, *7*(4), 815–840.

Marlin, B. M. (2008). *Missing Data Problems in Machine Learning*.

Masyn, K. E. (2013). Latent class analysis and finite mixture modeling. In *The Oxford handbook of quantitative methods: Statistical analysis, Vol. 2* (pp. 551–611). Oxford University Press.

Mazza, G. L., Enders, C. K., & Ruehlman, L. S. (2015). Addressing Item-Level Missing Data: A Comparison of Proration and Full Information Maximum Likelihood Estimation. *Multivariate Behavioral Research*, *50*(5), 504–519. https://doi.org/10.1080/00273171.2015.1068157

McNeish, D. M. (2015). Using Lasso for Predictor Selection and to Assuage Overfitting: A Method Long Overlooked in Behavioral Sciences. *Multivariate Behavioral Research*, *50*(5), 471–484. https://doi.org/10.1080/00273171.2015.1036965

Merz, C.J. & P.M. Murphy (1998), *UCI Repository of machine learning databases* [http://www.ics.uci.edu/≈mlearn/MLRepository.htmll. Irvine, CA: University of California, Department of Information and Computer Science.

Morgan, J. N., & Messenger, R. C. (1973). *THAID, a sequential analysis program for the analysis of nominal scale dependent variables*.

Morgan, J. N., & Sonquist, J. A. (1963). Problems in the Analysis of Survey Data, and a Proposal. *Journal of the American Statistical Association*, *58*(302), 415–434. https://doi.org/10.1080/01621459.1963.10500855

Murthy, S. K., & Salzberg, S. (1995). Decision Tree Induction: How Effective is the Greedy Heuristic? *KDD*.

Muthén, B., Kaplan, D., & Hollis, M. (1987). On structural equation modeling with data that are not missing completely at random. *Psychometrika*, *52*(3), 431–462. https://doi.org/10.1007/BF02294365

Olinsky, A., Chen, S., & Harlow, L. (2003). The comparative efficacy of imputation methods for missing data in structural equation modeling. *European Journal of Operational Research*, *151*(1), 53–79. https://doi.org/10.1016/S0377-2217(02)00578-7

Peugh, J. L., & Enders, C. K. (2004). Missing Data in Educational Research: A Review of Reporting Practices and Suggestions for Improvement. *Review of Educational Research*, *74*(4), 525–556. https://doi.org/10.3102/00346543074004525

Podgorelec, V., Šprogar, M., & Pohorec, S. (2013). Evolutionary design of decision trees. *WIREs Data Mining and Knowledge Discovery*, *3*(2), 63–82. https://doi.org/10.1002/widm.1079

Poulos, J., & Valle, R. (2018). Missing Data Imputation for Supervised Learning. *Applied Artificial Intelligence*, *32*(2), 186–196. https://doi.org/10.1080/08839514.2018.1448143

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Raghunathan, T. E. (2004). What do we do with missing data? Some options for analysis of incomplete data. *Annual Review of Public Health*, *25*, 99–117. https://doi.org/10.1146/annurev.publhealth.25.102802.124410

Raghunathan, T. E., Lepkowski, J. M., Hoewyk, J. V., & Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology 27*, 85-95.

Raymond, M. R., & Roberts, D. M. (1987). A comparison of methods for treating incomplete data in selection research. *Educational and Psychological Measurement*, *47*(1), 13–26. https://doi.org/10.1177/0013164487471002

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria.

Rodgers, D. M., Jacobucci, R., & Grimm, K. J., (2021). A Multiple Imputation Approach for Handling Missing Data in Classification and Regression Trees. *Journal of Behavioral Data Science*, *1* (1), 127-153. https://doi.org/10.35566/jbds/v1n1/p6

Rubin, D. B. (1976). Inference and Missing Data. *Biometrika*, *63*(3), 581–592. JSTOR. https://doi.org/10.2307/2335739

Rubin, D. B., & Schenker, N. (1991). Multiple imputation in health-are databases: An overview and some applications. *Statistics in Medicine*, *10*(4), 585–598. https://doi.org/10.1002/sim.4780100410

Rubin, D. B. (1986). Statistical Matching Using File Concatenation With Adjusted Weights and Multiple Imputations. *Journal of Business & Economic Statistics*, *4*(1), 87–94. https://doi.org/10.1080/07350015.1986.10509497

Saar-Tsechansky, M., & Provost, F. (2007). Handling Missing Values when Applying Classification Models. *Journal of Machine Learning Research*, *8*(57), 1623–1657.

Salzberg, S. L. (1994). C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, *16*(3), 235–240. https://doi.org/10.1007/BF00993309

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. Chapman and Hall/CRC. https://doi.org/10.1201/9780367803025

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*(2), 147–177. https://doi.org/10.1037/1082-989X.7.2.147

Schafer, J. L., & Olsen, M. K. (1998). Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst's Perspective. *Multivariate Behavioral Research*, *33*(4), 545–571. https://doi.org/10.1207/s15327906mbr3304_5

Sela, R. J., & Simonoff, J. S. (2012). RE-EM trees: A data mining approach for longitudinal and clustered data. *Machine Learning*, *86*(2), 169–207. https://doi.org/10.1007/s10994-011-5258-3

Strobl, C., Malley, J., & Tutz, G. (2009). An Introduction to Recursive Partitioning: Rationale, Application and Characteristics of Classification and Regression Trees, Bagging and Random Forests. *Psychological Methods*, *14*(4), 323–348. https://doi.org/10.1037/a0016973

Tang, F., & Ishwaran, H. (2017). Random forest missing data algorithms. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *10*(6), 363–377. https://doi.org/10.1002/sam.11348

Tanner, M. A., & Wong, W. H. (1987). The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association*, *82*(398), 528–540. https://doi.org/10.2307/2289457

Therneau, T., & Atkinson, B. (2019). *rpart: Recursive Partitioning and Regression Trees* (4.1-15) [Computer software]. https://CRAN.R-project.org/package=rpart

Twala, B. (2009). An empirical comparison of techniques for handling incomplete data using decision trees. *Applied Artificial Intelligence*, *23*(5), 373–405. https://doi.org/10.1080/08839510902872223

Twala, B., Jones, M. C., & Hand, D. J. (2008). Good methods for coping with missing data in decision trees. *Pattern Recognition Letters*, *29*(7), 950–956. https://doi.org/10.1016/j.patrec.2008.01.010

van Buuren, S. (2012). *Flexible Imputation of Missing Data*. CRC Press.

van Buuren, S. (2018). *Flexible imputation of missing data*. CRC press.

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, *45*(1), 1–67. https://doi.org/10.18637/jss.v045.i03

van Buuren, S. (2007). Multiple imputation of discrete and continuous data by fully conditional specification. *Statistical Methods in Medical Research*, *16*(3), 219–242. https://doi.org/10.1177/0962280206074463

Venables, W. N., & Ripley, B. D. (1997). *Modern applied statistics with S-Plus* (2nd ed). Springer.

Yarkoni, T., & Westfall, J. (2017). Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. *Perspectives on Psychological Science: A Journal of the Association for Psychological Science*, *12*(6), 1100–1122. https://doi.org/10.1177/1745691617693393

Zhang Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, *4*(11), 218. https://doi.org/10.21037/atm.2016.03.37