Spring 5-2016

Classification Trees and Rule-Based Modeling Using the C5.0 Algorithm for Self-Image Across Sex and Race in St. Louis

Rohan Shirali
Washington University in St. Louis

Recommended Citation
Shirali, Rohan, "Classification Trees and Rule-Based Modeling Using the C5.0 Algorithm for Self-Image Across Sex and Race in St. Louis" (2016). *Arts & Sciences Electronic Theses and Dissertations*. 718.
https://openscholarship.wustl.edu/art_sci_etds/718

WASHINGTON UNIVERSITY IN ST. LOUIS

Department of Mathematics

Classification Trees and Rule-Based Modeling Using the C5.0 Algorithm for Self-Image Across Sex and Race in St. Louis

by

Rohan Shirali

A thesis presented to the
Graduate School of Arts and Sciences
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Arts

May 2016
St. Louis, MO

Contents

List of Tables

Acknowledgements

I would first like to thank my advisor Dr. Todd Kuffner for working with me in such a patient and helpful manner. With the guidance of Dr. Kuffner and others in the mathematics department, I've been able to learn so much about the theory, practice, and history of statistics and mathematics. I would also like to thank Dr. Susan Racette and Dr. Ruth Clark for providing me with an opportunity to work with and refine a large data set, giving me a thorough sense of the research process. I would also like to thank Professor Han Gan for taking the time to attend and critique the oral defense of this thesis.

Finally, I thank my classmates and parents for their constant support.

ABSTRACT OF THE THESIS

Classification Trees and Rule-Based Modeling Using the C5.0 Algorithm for Self-Image Across Sex and Race in St. Louis

by

Rohan Shirali

Master of Arts in Statistics

Washington University in St. Louis, 2016

Professor David Wright, Chair

The study population comprised children, adolescents, and adults who were residents of the city of St. Louis at the time of data collection in 2015. The data collected includes sex, age, race, measured height and weight, self-reported height and weight, zip code, educational background, exercise and diet habits, and descriptions and strategies of participants' weight (i.e. overweight and trying to lose weight, respectively). I use the C5.0 algorithm to create classification trees and rule-based models to analyze this population. Specifically, I model a binary self-image variable as a function of sex, age, race, zip code, and a ratio of reported versus measured BMI (body mass index), and a multi-level categorical weight description variable as a function of sex, age, race, zip code, BMI ratio, and weight strategy. I compare the performance of the C5.0 algorithm with and without rules and boosting for independent and grouped categories, for both the binary and multi-level outcome. This comparison is limited due to sample size constraints. Ultimately, C5.0 performed best when modeling the binary variable and using either rules or boosting for independent categories.

1 Introduction

Background

In this paper, the research team will be using classification tree modeling algorithms to examine self-image across different demographics in St. Louis. The study population comprised children, adolescents, and adults who were residents of the city of St. Louis at the time of data collection in 2015. The research team collected data from a variety of community sites, including public schools, YMCA, faith organizations, university settings, and more. Data was collected through a survey and by taking certain measurements, and survey information was entered into REDCap, an online data entry tool. REDCap can take certain quality control checks, and in exporting our data for further manual quality control checks, we omitted subjects whose consent was not obtained. For inclusion in the study population, subjects' data were required to have measured height, measured weight, and sex. This study was approved by the Washington University in St. Louis Institutional Review Board.

The researchers at the Washington University in St. Louis School of Medicine initially gathered this data for a public health survey, examining weight status across different demographics and areas in St. Louis. They intend to identify areas and demographics with particularly affected weight status, and improve ongoing public health initiatives to address obesity according to the behaviors and/or resource allocation that influences the distribution of obesity. The data we collected, which I will describe in depth soon, is also suited to allow the researcher to analyze individuals' body image perceptions across these demographics, and I find self-image across race and sex to be especially pertinent today.

Research is continually surfacing that points to disparities in the way certain demographics are treated and portrayed in the media, and how that can influence individuals' self-image. Specifically, women are often depicted in ways that set unreasonable standards of beauty. This depiction affects individuals, specifically younger women, in how they perceive themselves, and may lead women to want to lose weight even when they are underweight or

Table 1: Gender Distribution (n=3307)

Gender	Percentage of Sample	Total
Male	47.0%	1554
Female	53.0%	1753

have a healthy weight. Additionally, people of different races tend to be treated and portrayed differently in the media. Recently this has been relevant with issues like "Hollywood whitewashing," an issue in which actors and actresses of color are seldom recognized and awarded for their work. We might expect white participants to have a more positive self-image than most minorities, in a way that parallels their media depiction. The research team is interested in examining whether our data from St. Louis in 2015 reflects these expected discrepancies in self-image across race and sex.

Data Description

Data was gathered through a survey in which participants gave information about their age, sex, race, ethnicity, zip code, education, exercise habits, and diet. Subjects also estimated and then measured their own height and weight. The distribution of age was skewed right, ranging from 3 - 87 years old, with mean 16.57 and median 11.10 years old. Most participants were between the ages of 9 and 16 years old. Those who were at least 20 years old were categorized as adults, and subjects under 20 were considered youth; 83.7% of the population were youth, and 16.3% adult. 53% of the subjects were female and 47% were male (n=3307). The distribution of weight was skewed right, presumably relating to the similarly skewed distribution of age, with mean 51.26 and median 46.00 kilograms. The distribution of height was approximately normal, with mean 149.5 centimeters. Body Mass Index, or BMI, was calculated using the formula:

$$\text{BMI} = \frac{\text{Weight in kg}}{(\text{Height in cm})^2}$$

The BMI distribution is also skewed right, with mean 22.07 and median 20.40 kg/cm^2.

Some participants did not fill out certain information in their surveys, so smaller subsets

Table 2: Race Distribution (n=2892, 415 missing)

Race	Percentage of Sample	Total
Black	72.6%	2100
White	22.3%	646
Asian	4.2 %	121
American Indian	0.8%	23
Native Hawaiian	0.1%	2

of the population will be taken into account for analysis pertaining to those variables. The racial distribution of the population was as follows: 72.6% Black, 22.3% White, 4.2% Asian, 0.8% American Indian, 0.1% Native Hawaiian (n=2892). All participants (adult and youth) were asked to describe their current strategy for gaining, or losing weight, with the following distribution: 52.2% are trying to lose weight, 11.0% are trying to gain weight, 19.8% are trying to stay the same weight, and 17.0% are not trying to do anything about their weight (n=864). In order to examine weight strategy as a binary variable for self-image, we will associate negative self-image with trying to either lose or gain weight, and positive self-image with trying to stay the same weight or not trying to do anything about one's weight. By this categorization, 63.2% have negative self-image and 36.8% have positive self-image (n=864). Additionally, subjects categorized as youth were asked to describe their own weight from a list of options, with the following results: 55.9% said about the right weight, 22.8% slightly overweight, 14.4% slightly underweight, 3.8% very overweight, 3.1% very underweight (n=320). I will be using the subsets of the population that have data for weight strategy and weight description, respectively, for my two sets of analyses.

As previously mentioned, participants were asked to report their height and weight prior to having measured their height and weight. This data required a bit more quality control, since certain subjects reported heights and weights that are physiologically unreasonable. Self-reported height values were excluded if they were at least 8 feet or less than 3 feet. Self-reported weight values were excluded if they were at least 650 pounds or less than 50 pounds. Self-reported BMI values, calculated from self-reported height and weight, were excluded if they were at least 100 kg/cm^2 or less than 12 kg/cm^2. These exclusions were

Table 3: Gender Distribution for Self Report Data (n=777, 2530 missing)

Gender	Percentage of Sample	Total
Male	41.8%	325
Female	58.2%	452

not chosen arbitrarily; the CDC recommends these as exclusion criteria for self-reported information [CDC (2015)]. For the sake of having only one subset of self-reported data, and since the exclusion criteria are quite lenient, any individual who failed to meet all three criterion was excluded entirely. After accounting for the exclusions, the distribution of the ratio of reported weight to measured weight is approximately normal with slight left skew and mean 0.9951. The distribution of the ratio of reported height to measured height is approximately normal with slight right skew and mean 1.006. This information gives us the sense that on average, the population tended to slightly underestimate their weight and slightly overestimate their height. Being slightly taller and skinnier would reflect a decrease in BMI, and as expected, the distribution of the ratio of reported BMI (calculated from reported height and weight according to BMI formula) to measured BMI is approximately normal with slight left skew and mean 0.9847.

I consider weight strategy, weight description, BMI ratio, height ratio, and weight ratio, as variables especially pertinent to self-image. The research team will be examining these self-image variables across different demographics; we should ensure that the distributions of demographics, particularly sex and race, are similar for the subsets of the population that we analyze as for the entire population. It is important to check these distributions because as previously mentioned, there are a lot of missing data for some of the self-image variables.

Of those subjects with information about their weight strategy, 57.8% were female and 42.2% were male, 61.7% were adults and 38.3% were youth (n=864). The racial distribution was as follows: 52.4% Black, 37.2% White, 8.2% Asian, 2.0% American Indian, and 0.2% Native Hawaiian (n=858, 6 were missing information on race). Of those with a description of their own weight, all of whom are youth subjects, 47.2% were female and 52.8% were

Table 4: Weight Strategy Data (n=864, 2443 missing)

Weight Strategy	Percentage of Sample	Total
Trying to gain weight	11.0%	95
Trying to lose weight	52.2%	451
Trying to stay the same weight	19.8%	171
Not trying to do anything about their weight	17.0%	147

Table 5: Gender Distribution for Weight Strategy Data (n=864)

Gender	Percentage of Sample	Total
Male	42.2%	365
Female	57.8%	499

male (n=320). The racial distribution was as follows: 64.6% Black, 25.4% White, 5.6% Asian, 3.8% American Indian, and 0.6% Native Hawaiian (n=319, 1 missing). Later when I model weight description, weight strategy will be included as a predictor; the weight strategy distribution within those who had weight description is: 16.7% trying to gain weight, 42.6% trying to lose weight, 19.2% trying to stay the same weight, and 21.5% not trying to do anything about their weight (n=317, 3 missing). Of those subjects whose self reported height, weight, and BMI were included for analysis, 58.1% were female and 41.9% were male, 65.2% were adults and 34.8% were youth (n=777). The racial distribution was as follows: 52.1% Black, 38.0% White, 8.0% Asian, 1.7% American Indian, 0.1% Native Hawaiian (n=771, 6 missing). With the exception of the proportion of adults and youth in these subsets, specifically the subset for weight description which includes only youth, the distributions tend to resemble that of the study population.

Since the distributions of variables in the subsets of the data are similar to those for the

Table 6: Race Distribution for Weight Strategy Data (n=858, 6 missing)

Race	Percentage of Sample	Total
Black	52.4%	450
White	37.2%	319
Asian	8.2%	70
American Indian	2.0%	17
Native Hawaiian	0.2%	2

Table 7: Weight Description Data (n=320, 2987 missing)

Weight Description	Percentage of Sample	Total
Very underweight	3.1%	10
Slightly underweight	14.4%	46
About the right weight	55.9%	179
Slightly overweight	22.8%	73
Very overweight	3.8%	12

Table 8: Gender Distribution for Weight Description Data (n=320)

Gender	Percentage of Sample	Total
Male	52.8%	169
Female	47.2%	151

Table 9: Race Distribution for Weight Description Data (n=319, 1 missing)

Race	Percentage of Sample	Total
Black	64.6%	206
White	25.4%	81
Asian	5.6%	18
American Indian	3.8%	12
Native Hawaiian	0.6%	2

Table 10: Weight Strategy for Weight Description Data (n=317, 3 missing)

Weight Strategy	Percentage of Sample	Total
Trying to gain weight	16.7%	53
Trying to lose weight	42.6%	135
Trying to stay the same weight	19.2%	61
Not trying to do anything about their weight	21.5%	68

whole population, the data are well-suited for analysis. I will utilize some of the variables to try and model and predict the self-image binary variable I have created, and then create a second model to predict how youth might describe their weight based on the same predictors but including their weight strategy. This will allow the researcher to utilize classification tree modeling for categorical and binary outcomes, evaluating their relative performances. In modeling self-image, we expect to see higher self image scores for white adult males and lower scores for younger women of color.

2 Methodology

Decision Trees

I will analyze the data using classification trees, which are a kind of decision tree. Decision tree algorithms begin with a set of training vectors $x_i \in R^n, i = 1, ..., l$ and a label vector $y \in R^l$. First, decision trees recursively partition the data such that observations with the same label are grouped together. We will refer to the data at node m as Q. For each candidate split $\theta = (j, t_m)$ with feature j and threshold t_m, divide the data into subsets $Q_{left}(\theta)$ and $Q_{right}(\theta)$, so that

$$Q_{left}(\theta) = (x, y)|x_j \leq t_m$$

$$Q_{right}(\theta) = \frac{Q}{Q_{left}(\theta)}$$

We calculate a statistic called impurity at m using an impurity function $H()$ specified within different algorithms. Let n_{left} be the number of observations in Q_{left} and n_{right} be the number of observations in Q_{right}, so that $N_m = Q_{left} + Q_{right}$ is the total number of observations at a given node,

$$G(Q, \theta) = H(Q_{left}(\theta))\frac{n_{left}}{N_m} + H(Q_{right}(\theta))\frac{n_{right}}{N_m}$$

We choose the parameters such that the impurity is minimized

$$\theta^\star = \text{argmin}_\theta G(Q, \theta)$$

and continue recursively for subsets $Q_{left}(\theta^\star)$ and $Q_{right}(\theta^\star)$ until either the maximum allowed depth is attained, $N_m < \min_{samples}$, or $N_m = 1$. When the target variable outcome can take a finite set of classes with values $0, 1, ..., K - 1$, we use classification trees. In a node m

representing region R_m with N_m observations, let

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

be the proportion of observations from class k in node m. For classification trees, we use proportion $\hat{p}_{mk}$ with an appropriate impurity function. A few frequently used measures of impurity are:

$$\text{Entropy: } H(X_m) = -\sum_k \hat{p}_{mk} \log(\hat{p}_{mk}),$$

$$\text{Gini Index: } H(X_m) = \sum_k \hat{p}_{mk}(1 - \hat{p}_{mk}), \text{ and}$$

$$\text{Misclassification: } H(X_m) = 1 - max(\hat{p}_{mk})$$

This is how decision trees are formally created, using a minimization problem to create the tree sequentially. I will now explain more about classification trees in general and aspects of the specific models we use.

Classification Trees

The classification tree model is appropriate for my data since they can be easily interpretable and are capable of managing many predictors and missing data [Kuhn & Johnson (2013)]. We assume that all of the predictor variables have finite domains, and that the response belongs to a finite set of mutually exclusive classes. Classification trees are meant to model a finite set of values for the target variable, whereas regression trees are used when the target variable is continuous. Classification trees can often handle predictor variables that are continuous and categorical. The model will use a series of if-then statements, which can be visualized as a tree. The tree is split into branches that meet at nodes which specify different paths for different values of predictors, and prediction proceeds from the top of the tree on the path that corresponds to the predictors' values for each observation. This culminates at a terminal node, which represents a value of the target variable attained from the specific values of the predictors.

In practice, these models rely more on algorithms than theoretical foundation, and as such there are several different approaches for creating and pruning classification trees. Many of these algorithms have been written for packages in R, and I will use the C5.0 package to model and predict self-image. I will also examine the robustness of the procedure as it varies with different options, all of which I explain below.

Classification trees are used as a descriptive and predictive modeling approach in machine learning and data mining. The goal is to predict the value of a response variable based on given values of predictors. The way this takes place in classification trees is to divide the data into smaller subsets in which the nodes of the splits in the tree contain larger proportions of one class. In other words, the splits filter different values of predictor variables, and are meant to correspond to a more homogeneous set of response variables, wherein the rate of misclassification is minimized. The criterion for dividing the data into groups to minimize classification error is referred to as purity. There is an important distinction between purity as minimizing misclassification as opposed to maximizing accuracy, since maximizing accuracy might lead us to partition the data so that we sample mostly from one class of the target variable, which is not the intention and would not yield good results. Usually purity is evaluated using either the Gini index or information statistics, depending on which algorithm is being used for classification tree modeling. Split points are evaluated based on their purity, and there are partitioning algorithms that examine almost all split points and create new partitions based on which split optimizes the purity criterion. For continuous predictors, deciding the split will involve a simple optimization of the purity criterion over the domain of the variable. For categorical predictors, we have a choice of whether to optimize the purity criterion across the categories independently or in groups, the pros and cons of which are discussed below. It is important to note that partitioning algorithms tend to favor categorical predictors with many levels q, as the number of partitions grows exponentially in q and the more choices are available, the more likely it is the algorithm can find a good option for the given data [Hastie et al. (2008)]. The purity criterion for C4.5 and C5.0, also defined below, is based on the concept of information entropy.

The algorithm continues iteratively, making the tree more complex and adding split points according to the optimality criterion until it meets some specified stopping criteria. It is important not to overfit the tree with too many split points, because then the model would not be generalizable at all. Having too many split points is analogous to overfitting in regression in cases where, for example, an n-degree polynomial is fit to $n+1$ data points. So the algorithms then typically include a pruning step to simplify the tree, in which the branches of the tree are evaluated to see if they are individually contributing enough to the classification decisions to remain in the tree. There are several different methods for pruning, such as cost-complexity pruning or reduced-error pruning. I will describe the pruning process for C4.5 and C5.0 below. Classification trees can have many different specifications and variations to better adjust to different kinds of data, such as boosting, rules, and grouped categories. These features are also described below as they appear in C5.0, the algorithm I am using and evaluating, and C4.5, its predecessor.

C4.5 and C5.0 Algorithms

I will be using a modified version of Quinlan's C4.5 classification model, which became popular in the late 2000s, called C5.0. The algorithm has more features available to it than C4.5 to further specify the model. The C5.0 algorithm is faster, more memory efficient, and creates simpler trees than C4.5. C5.0 also offers winnowing, boosting, which can give the tree more accuracy, and unequal costs for different types of errors [Kuhn & Johnson (2013)]. Both C4.5 and C5.0 allow rule-based models and evaluation of variable importance. I will explain the C4.5 algorithm and the changes and additions made for C5.0 that contribute to these claimed advantages.

The core algorithm C4.5 builds decision trees using information entropy, and at each node chooses splits using normalized information gain as the purity criterion. Information gain is based on the idea of entropy, a measure of uncertainty from information theory. With p_i referring to the probability of a given class as the outcome for each of m possible classes for the target variable, information entropy is defined as

$$I_E(p) = -\sum_{i=1}^{m} p_i \log p_i = info(p)$$

If the probabilities of the classes are more balanced, the information entropy will be higher. In other words, the more equally balanced the family of probabilities, the more uncertainty associated with randomly guessing the true class of an observation. This makes sense: it's harder to correctly guess a coin flip ($p_1 = p_2 = 0.5$) than whether a die will land on 1 or not ($p_1 = \frac{5}{6}, p_2 = \frac{1}{6}$), so it is desirable to have more distinct probabilities and low information entropy.

For a given split in the classification tree, the information gain, or mutual information, refers to the difference in information entropy before and after the split. The information entropy before the split is straightforward to calculate, using each of the class probabilities as p_i from the equation above. But we must condition on the split to calculate the probabilities for each class after the split. So for a given split S with k partitions, let $info_i$ be the sum of the information entropy in the i^{th} resulting partition and n_i equal to the number of samples that would result in the i^{th} partition from S. So,

$$info^S_{before} = -\sum_{i=1}^{m} p_i \log p_i$$

$$info^S_{after} = \sum_{i=1}^{k} info_i \frac{n_i}{n}$$

We assign weights to each of the k partitions according to the ratio of the number of samples in that partition to the total number of samples after the partition. So the information gain is calculated as

$$gain(S) = info^S_{before} - info^S_{after}$$

If the difference between the class probabilities before versus after the split is substantial, we would expect a higher information gain. Note that both the information entropy and the information gain are convex functions. Higher information gain is desirable, since this would correspond to higher information entropy, or uncertainty, before the split as compared with after. We normalize the information gain to level our consideration of each class.

This normalized information gain is the optimality criterion that C4.5 uses to choose splits. The class with the highest normalized information gain is selected, and the algorithm continues this process for smaller subsets, and prunes the tree for branches that do not contribute to the purity of classification. The branches not contributing enough to the model are removed methodically according to different pruning processes in different algorithms.

The specific pruning process used by C4.5 and C5.0 is called pessimistic pruning. In pessimistic pruning, we evaluate each subtree individually to decide whether the tree as a whole should be simplified. Pessimistic pruning estimates the errors with and without each subtree by following the subsequent paths on the tree and using the class frequencies at the terminal nodes. For each error rate p, and s nodes in a given subtree,

$$p_{\text{without subtree}} = \frac{\text{Total number of misclassified samples}}{\text{Total number of samples}}$$

$$p_{\text{with subtree}} = \sum_{i}^{s} \frac{\text{Number of misclassified samples at node } i \text{ of subtree}}{\text{Total number of samples at node } i}$$

We choose α according to our confidence level and calculate the upper bounds for the errors using the normal approximation to the binomial distribution:

$$\text{Pessimistic upper bound} = p + z_{\alpha/2}\sqrt{\frac{p(1-p)}{n}}$$

For these situations, it is accepted to take $\alpha = 0.25$. Of course increasing the confidence would give us a larger tree, just as a 99% confidence interval is wider than a 90% confidence interval for the same data. We then compare the pessimistic upper bound for the error with and without the subtree: if the error rate is greater when we include the subtree then we prune it entirely.

For C4.5 and C5.0, pessimistic pruning serves as a less mathematical but often more practical approach than cross-validation, since it is quite effective and far more efficient in determining the tree size. The author of the C4.5 algorithm, Quinlan, notes that this procedure is not well-grounded theoretically, but does tend to yield good results, which only reflects my earlier comment that these algorithms are often more practical than theoretically based. There is shell script, called *xval*, included in C5.0 for cross-validation techniques. Cross-validation is meant to give a reliable estimate of the predictive accuracy of a given model. With k-fold cross-validation, split the training set into k equally sized blocks. For each block, we would construct a classifier from the rest of the blocks and test it on the selected block. One may choose to use multiple trials of a k-fold cross-validation to be really confident about the accuracy. The default for C5.0 is to compute 1 trial of a 10-fold cross-validation.

C5.0 offers an option for a rule-based model, which utilizes independent conditional statements to evaluate individual rules and prune the model accordingly. First, we create an initial tree including every path, and collapse each path into a rule. We evaluate each rule on independent conditional statements to assess if they can be generalized by eliminating terms in the conditional statement. For example, if we are evaluating rule d out of D possible rules, we consider

$$p(x_d|x_1, x_2, ..., x_{d-1}, x_{d+1}, x_{d+2}, ..., x_D)$$

and then eliminate terms and evaluate

$$p(x_d|x_2, ..., x_{d-1}, x_{d+1}, x_{d+2}, ..., x_D), p(x_d|x_3, ..., x_{d-1}, x_{d+1}, x_{d+2}, ..., x_D)$$

and so on. We then compare our error rate to a pre-determined pessimistic error rate (distinct from the pessimistic error rate upper bound mentioned for the tree model pruning step) and remove the worst rule if any are not above the pessimistic error rate. The pessimistic error rate is recalculated and the same test is conducted for the smaller tree. This continues iteratively until all conditions are above the baseline error rate or all are removed, in which case the rule is removed. Then C5.0 has each rule vote for the most likely class and the class with the highest vote is used. The predicted confidence value will be that associated with the most specific active rule [Kuhn & Johnson (2013)]. I will utilize C5.0 models both with and without rules.

Boosting, in which many weak classifiers are combined into a strong classifier, is available in C5.0 as well. Boosting is meant to reduce bias and variance. The boosting algorithm utilized by C5.0 is similar to the AdaBoost algorithm from the early 1990s [Kuhn & Johnson (2013)]. It fits models sequentially, adjusting each case weight based on accuracy. C5.0 constructs trees to have approximately the same size as the first tree, forcing them to have a similar number of terminal nodes. Consider training set size N, with $N_{incorrect}$ incorrectly classified samples. Let w_r be the sample weight for the observation at the rth boosting iteration, and $S_{correct}$ and $S_{incorrect}$ be the sum of weights for the correctly and incorrectly classified samples, respectively. Then the boosting algorithm finds the midpoint between the sum of the weights for incorrectly classified samples and half of the overall sum of weights:

$$midpoint = \frac{1}{2}[\frac{1}{2}(S_{incorrect} + S_{correct}) - S_{incorrect}] = \frac{1}{4}(S_{correct} - S_{incorrect})$$

Now the correctly classified samples are adjusted accordingly:

$$w_r = \frac{w_{r-1}(S_{correct} - midpoint)}{S_{correct}}$$

and the incorrectly classified samples take:

$$w_r = w_{r-1} + \frac{midpoint}{N_{incorrrect}}$$

With this algorithm, the weights change substantially for each misclassification, but the weights decrease slowly at a decreasing rate when the samples are correctly classified. C5.0 includes an automatic stop feature, in which the algorithm detects when the model is especially effective or ineffective and stops boosting. Specifically, if $S_{incorrect} < 0.10$ or if the average w_r for the incorrect samples $\bar{w}_{r,incorrect} > 0.50$.

There are two ways to treat categorical predictor data for tree and rule-based models. Both are available as options in the C5.0 algorithm. Categories will be either grouped or independent. Grouped categories have each categorical predictor entered as a single entity such that the model itself decides how to split them. Independent categories have the predictors decomposed into binary dummy variables, and each is considered independently. If only some of the categories are highly predictive, usually the grouped category approach will be better. With my data, for example, the model may decide to group predictors sex and race together if they are highly predictive of the target and the others are not. Since we do not have that many predictor variables and do not expect any to be significant more predictive than the other variables, we might expect the independent category approach to be more appropriate. Because these approaches have their respective advantages and disadvantages depending on the data and model, we utilize both and then evaluate and compare their performance.

There are missing values for several of the variables, and we are assuming here the the mechanism for missing data is random. We must account for or consider whether to use these variables with many missing values, because we want to keep the sample size large enough for good analysis while including enough predictors to evaluate their relative importance. I will use predictor variables sex, age, race, zip code, and BMI ratio. Depending on their values, we will predict the value of a participant's self-image (positive or negative), based on the earlier binary self-image variable constructed from weight strategy information. I will then include weight strategy or the binary self-image variable with the predictors and try to model weight description for the subset of youth who have the corresponding information. I will compare the models in terms of their error rates, calculated simply as:

$$\text{Error rate} = \frac{\text{Total number of misclassified samples}}{\text{Total number of samples}}$$

These error rates are calculated both for the training set on which the models are built and on the test set on which the models are tested.

3 Results

I have read the data into R, and run several C5.0 models with different specifications. The corresponding R code is provided for each model. For all of the models, I use half of the sample as a training set and check the accuracy of the model with the remaining half of the data (the test set). The training set error rates are a result of the algorithm not overfitting the data with too many splits, and the test set error rates are those that result from prediction with the remaining half of the data. What follows is a discussion of the models' relative error rates and sizes, based on the different specifications for the C5.0 algorithm.

The first model fits the C5.0 algorithm to predict binary self-image variable from zip code, sex, race, BMI ratio, and age. All of the models for the binary self-image variable have training and test sample sizes $n = 432$, respectively. This model does not use rules or boosting, and has independent categories. The error rate was 21.5% for the training data and 42.1% for the test data, with tree size 35. The tree uses zip code in nearly all of the splits, age in over half, and the remaining three variables in less than 20% of the splits. For example, the beginning of out decision-making for this model proceeds as follows: depending on the value of zip code, prediction will proceed down a specific path. Junctions that follow will split predictions based on the subject's age, and later the subject's race, BMI ratio, and sex.

Next I fit a C5.0 model with rules, independent categories, and no boosting, using the same predictor and response variables. The training set error rate was 28.2% and the test set error rate was 36.8%, with 14 rules. This model utilizes race in nearly all of the splits, with BMI ratio and sex in about half of the splits. Although the error rate is still quite high, this model is better than the original, since it is simpler with a lower error rate.

The C5.0 model without rules or boosting and with grouped categories has training set error rate 23.4% and test set error rate 42.1%, with tree size 31. Although the test set error rate matches exactly that of the original model, they do still classify the data differently, as seen in the R output. The grouped model uses zip code in almost all splits, and sex in about a third of the splits, and can be considered a slight improvement from the original model since it is simpler and has less branches. Note that the output given from C5.0 actually does not specify which predictors are grouped together in the model.

Next I fit a C5.0 model with boosting (10 trials) and independent categories, but without rules. In spite of calling 10 trials, the boosting was reduced to 1 trial since the last classifier is inaccurate and there are few classifiers. Still, it performed better than the original: the training set error rate is 26.4% and the test set error rate is 35.6%, with tree size 24. The model boosted with 10 trials uses race in every split, and zip code in about half of the splits. The model with 10 trials can be considered better than the original and the grouped model, and is approximately as good of a model as the rule-based model, which is simpler but has a slightly higher test set error rate. To check if a model with real boosting (more trials) improves performance, I ran the model with 30 trials. The boosting was reduced to 4 trials for the same reasons, but the test set error rate actually increased slightly, to 38.0%.

For my data set and for independent categories, using either rules or boosting improves the performance of the model, with the rule-based model being slightly simpler with a slightly higher error rate. Even though there was only 1 trial for boosting, it still improved the model, which shows us that the core algorithm is affected by including boosting. Again, neither is particularly good at prediction, with test set error rates all exceeding 35%, which is pretty high. Noting that boosting cannot be used for rule-based models, we compare the two methods for grouped categories as well.

The model using rules and grouped categories has training set error rate 26.2% and test set error rate 38.4%, with 13 rules. This is a slightly worse performance than the model with rules and independent categories. Similar results follow for using boosting, which again was reduced to 1 trial for similar reasons as previously mentioned. The training set error rate was 20.1% and the test set error rate was 40.7%, with tree size 38.

The next set of models will predict how youth describe their own weight from a list of 5 options (very underweight, slightly underweight, about the right weight, slightly overweight,

Table 11: Error Rates and Tree Size for C5.0 Variations Modeling Binary Self-Image Variable

Categories	**Rules**	**Boosting**	**Training Error Rate**	**Test Error Rate**	**Tree Size**
Independent	No	No	21.5%	42.1%	35
Independent	Yes	No	28.2%	36.8%	14
Independent	No	10 Trials	26.4%	35.6%	24
Independent	No	30 Trials	23.8%	38.0%	18.25
Grouped	No	No	23.4%	42.1%	31
Grouped	Yes	No	26.2%	38.4%	13
Grouped	No	10 Trials	20.1%	40.7%	38

Table 12: Variable Importance for C5.0 Modeling Binary Self-Image Variable

Variable: Importance	**Average**	**Max**	**Min**
Zip Code	70.40%	99.77%	21.06%
Age	51.09%	99.77%	18.29%
Race	35.09%	89.35%	9.95%
Sex	11.51%	25.23%	0.00%
BMI Ratio	7.74%	16.44%	2.08%

very overweight) based on their sex, age, race, BMI ratio, zip code, and weight strategy, as defined earlier. We will see if similar results follow for variations of the C5.0 algorithm for this multi-level categorical response variable compared with the earlier binary response variable. We must note that the sample size is less than half for this subset of the population compared with that for modeling binary self-image ($n = 160$ for training and test sets, respectively). Surely this will limit the performance of these models relative to the earlier models.

The basic C5.0 model fits a tree with 1 branch with training set error rate 43.1% and test set error rate 45.0%, which is clearly not a very useful model. Adding rules into the model gives training set error rate 39.4% and test set error rate 44.4% with 4 rules. Weight strategy is utilized in nearly all the splits and age is in about a fourth of the splits. This is hardly an improvement from the original model. Fitting a model with 10 trials for boosting, the boosting is reduced to 7 trials since the last classifier is very inaccurate. Although the boosted training set error rate is very low, 6.9%, the test set error rate, which is really what matters more, is 52.5%, which is a reduction in performance from the original model. The boosted model uses weight strategy in nearly all splits, and age and race in over half of the splits.

Table 13: Error Rates and Tree Size for C5.0 Variations Modeling Multi-Level Weight Description Variable

Categories	**Rules**	**Boosting**	**Training Error Rate**	**Test Error Rate**	**Tree Size**
Independent	No	No	43.1%	45.0%	1
Independent	Yes	No	39.4%	44.4%	4
Independent	No	10 Trials	6.9%	52.5%	3
Independent	No	30 Trials	32.5%	46.9%	1
Grouped	No	No	44.4%	43.8%	1
Grouped	Yes	No	30.0%	48.8%	5
Grouped	No	10 Trials	40.6%	45.6%	2

Oddly, when we specify 30 trials for boosting, the boosting is reduced to just 1 trial, and gives a tree with 5 branches, training set error rate 32.5% and test set error rate 46.9%. The discrepancy in when the boosting stops for different numbers of trials raises some concerns about the algorithm. None of the models above seem adequate to model weight description, but the rule-based model seems to be slightly more reliable than the boosted model, neither of which are substantial improvements in performance from the original model.

When we allow for grouped categories but do not use rules or boosting, the training set error rate is 44.4% and the test set error rate is 43.8%, again with only 1 branch, so this model is not very useful either. Using rules and grouped categories, we get a model with 5 rules, training set error rate 30.0% and test set error rate 48.8%, and again uses weight strategy in nearly all of the splits. Finally, we boost with 10 trials for grouped categories, and get a decision tree with size 2. The training set error rate was 40.6% and the test set error rate was 45.6%, which is again not a substantial improvement from the original model.

Although we must take into account the limitations of sample size, the C5.0 model is not adequate to predict weight description from the given predictor variables, since it has an error rate of about 50% with each variation on the algorithm.

In evaluating variable importance, we see that not much of the models' predictive abilities are attributed to race and sex. Zip code and age were used much more frequently in determining classification, which makes sense considering what we mentioned earlier about these algorithms favoring classification variables with many different levels. Noting that a continuous variable may be split in many different ways, the algorithm tends to favor continuous predictors over categorical variables with few levels as well. So in interpreting the results, it is hard to say whether sex and race have an impact on self-image in St. Louis, since we are not sure whether our model is being influenced by the nature of the variables or the content of their values.

References

CENTERS FOR DISEASE CONTROL AND PREVENTION (2015). Obesity Prevalence Maps.

HASTIE, T., TIBSHIRANI, R. & FRIEMAN, J. (2008). *The Elements of Statistical Learning*, Second Edition. New York: Springer.

KUHN, M. & JOHNSON, K. (2013). *Applied Predictive Modeling*. New York: Springer