! TECHNICAL GUIDE #3 !

CLASSIFICATION AND REGRESSION TREES: AN INTRODUCTION

Yisehac Yohannes
John Hoddinott

International Food Policy Research Institute
2033 K Street, N.W.
Washington, D.C. 20006 U.S.A.

March, 1999

CONTENTS

1. INTRODUCTION[1]

Household food security (HFS) represents the guiding principle underlying many rural development projects. It plays an important role in the targeting of projects, the selection of appropriate interventions, and the monitoring and evaluation of projects. HFS is a multifaceted concept that does not necessarily lend itself to measurement by single, discrete indicators. Further, such indicators should reflect the behavior and livelihood conditions of target populations—those that are most often, and more severely, affected by acute food insecurity (Borton and Shoham 1991). These include the rural poor, women-headed households, asset-poor pastoralists, the landless, recently resettled households, and households constrained by a high dependency ratio.

The multifaceted nature of HFS implies that reliance on a single indicator is unlikely to capture all dimensions of food security. Consequently, Borton and Shoham (1991) suggest 20 core indicators; Frankenberger (1992), and Seaman, Holt, and Allen (1993) each take between 20 and 30 indicators as the starting point; Riely (1993) and Downing (1993) both suggest more than 50 variables; while Currey (1978), one of the earliest practitioners in the field, started with 60 variables for his analysis of vulnerability in Bangladesh. The large number of potential indicators presents development practitioners with several, interlinked analytical problems. First, it is not always clear what criteria should be used to select a set of indicators out of those available. Second, all other things being equal, there is a strong argument for using as parsimonious a set of variables as possible, but the precise number is difficult to identify in advance. In order to do so, it is necessary to determine which variables are influencing each other and are therefore not "independent" (additive) indicators of vulnerability. It is also necessary to attach weights to the variables selected as indicators and the existing literature does not provide adequate guidance as to how this should be undertaken. Finally, one would like to have a sense of the predictive value of these indicators.

This guide introduces development practitioners to a statistical software package, Classification and Regression Tree (CART), that addresses these problems. CART is a

[1] Funding for data collection and analysis of these data has been supported by the International Fund for Agricultural Development (TA Grant No. 301-IFPRI). We gratefully acknowledge this funding, but stress that ideas and opinions presented here are our responsibility and should, in no way, be attributed to IFAD.

nonparametric technique that can select from among a large number of variables those and their interactions that are most important in determining the outcome variable to be explained. (Two other sets of methods—working closely with local people who can help define indicators of local significance—and parametric methods for choosing outcome indicators of food security are described in Technical Guide #6 and #7, respectively.) In order to illustrate the basic principles of CART methodology, and to demonstrate the power of this methodology, the guide begins with an extended example. It then outlines reviews a number of technical details, including the hardware and software requirements and how to program in CART. The concluding section outlines additional applications as well as describing the strengths and weaknesses of CART methodology. Appendix 1 discusses in more detail how CART constructs a classification tree and Appendix 2 provides an annotated guide to a sample of CART output.

Development practitioners interested in using CART methodology are encouraged to consult three documents that provide more information than can be contained in this short guide. These are: *Classification and Regression Trees: A User Manual for Identifying Indicators of Vulnerability to Famine and Chronic Food Insecurity* (Yohannes and Webb 1998); *Classification and Regression Trees* (Breiman, Friedman, Olshen, and Stone, 1984). This volume provides a detailed overview of the theory and methodology of CART, and illustrates a number of examples in many disciplinary areas. A third document is *CART: Tree-Structured Non-Parametric Data Analysis* by Steinberg and Colla (1995)—the CART software manual that provides many details on customizing CART programs.

2. A CART EXAMPLE: INDICATORS OF HOUSEHOLD FOOD INSECURITY IN NORTHERN MALI

Suppose we want to target an intervention to villages that have a high concentration of food insecure households. We do not have the resources to conduct a large-scale household census that measures food security status in all these households, but we do have (1) a smaller household survey with a measure of food security status, and (2) the ability to ask a few simple

questions of each household in our intervention area. How can we use our existing information to learn what variables would provide us with an indication of which households are most likely to be food insecure?

The information available to us consists of calories available per person for 275 households (these data are taken from a survey in the Zone Lacustre). Households are separated into two groups: food insecure, those where caloric availability is less than 2,030 kilocalories per day; and food secure, those where caloric availability exceeds 2,030 kilocalories per day. Table 1 lists approximately 40 possible correlates of household food security. Given the large number of potential correlates with household food security—and the even larger number of potential interactions between these, we would like to know how to derive from these data some simple correlates of food security. CART is a way of doing so. Here the dependent variable is categorical (food secure or not food secure), and so CART produces a classification tree. Where the variable is continuous, say calories available per person per day, it produces a regression tree. Regardless of the nature of the dependent variable, CART proceeds in the following fashion.

CART begins with the entire sample of households. This sample is heterogeneous, consisting of both food-secure and food-insecure households. It then divides up the sample according to a "splitting rule" and a "goodness of split criteria." Splitting rules are questions of the form, "Is the dependency ratio less than two?" or put more generally, is $X \leq d$, where X is some variable and d is a constant within the range of that variable. Such questions are used to divide or "split" the sample. A goodness of split criteria compares different splits and determines which of these will produce the most homogeneous subsamples. Following on from our example, we would like to disaggregate our sample into food-secure and food-insecure households. As there are many variables to consider, there are a large number of possible disaggregations of the sample. The approach taken by CART is to produce a very large disaggregation and then apply a set of rules to reduce these.

Figure 1 is taken directly from the output produced by CART. (A detailed explanation of how CART works, and other output produced by the program, are found in Appendices 1 and 2.) We assume that caloric availability per person ("CALSDUM") is a good proxy for household food insecurity (see Technical Guide #7 for a further discussion). Approximately 35 percent of all households are food insecure by this definition. This is shown at the top of Figure 1 in the

Table 1 Possible correlates of household food security

VILL1	Village dummy (=1 if village=1, =0 o/w)
VILL2	Village dummy (=1 if village=2, =0 o/w)
VILL3	Village dummy (=1 if village=3, =0 o/w)
VILL4	Village dummy (=1 if village=4, =0 o/w)
VILL5	Village dummy (=1 if village=5, =0 o/w)
VILL6	Village dummy (=1 if village=6, =0 o/w)
VILL7	Village dummy (=1 if village=7, =0 o/w)
VILL8	Village dummy (=1 if village=8, =0 o/w)
VILL9	Village dummy (=1 if village=9, =0 o/w)
VILL10	Village dummy (=1 if village=10,=0 o/w)
HHSIZE	Household Size
CASHGIVE	Dummy (=1 if household was given Cash, =0 o/w)
CASHSENT	Dummy (=1 if household was sent Cash, =0 o/w)
REMIT	Dummy (=1 if household received remittances, =0 o/w)
ASSTVM1	Total value of male assets
CEMENT	Dummy (=1 if floor of a house is cement, =0 o/w)
MFERT	Dummy (=1 if male farmer used fertilizer, =0 o/w)
MPEST	Dummy (=1 if male farmer used pesticides, =0 o/w)
MSEED	Dummy (=1 if male farmer used improved seeds, =0 o/w)
MLLABOR	Dummy (=1 if male farmer labor is used, =0 o/w)
MINPUT	Dummy (=1, if male farmer used any of the inputs, =0 o/w)
NONAGDUM	Dummy (=1, if any males engaged in non-agricultural activities =0 o/w)
BOENUMM	Number of bullocks owned by male household members now
BOEVM	Present value of bullocks owned by male household members now
VACNUMM	Number of cows owned by male household members now
VACVM	Present value of cows owned by male household members now
FFERT	Dummy (=1 if female farmer used fertilizer, =0 o/w)
FPEST	Dummy (=1 if female farmer used pesticides, =0 o/w)
FSEED	Dummy (=1 if female farmer used improved seeds, =0 o/w)
FLLABOR	Dummy (=1 if female labor is used, =0 o/w)
FINPUT	Dummy (=1, if female farmer used any of the inputs, =0 o/w)
ADTNUMF	Number of draft animals owned by female household members now
ADTVF	Present value of draft animals owned by female household members no
ASSTVF1	Total value of female assets
FNNAGDUM	Dummy (=1, if any females engaged in non-agricultural activities =0 o/w)
DEPRAT	Dependency ratio
CALSDUM	Calorie Dummy (=1 if Per capita daily calories > 2030, =0 o/w)

Source: Mali household survey data set, 1998.

box labeled "Node 1." The "N" refers to the sample size, which recall is 275. This box is referred to as the *root node*.

The first split of the root node is based on female asset holdings being less than or equal to 33825 FCFA. CART divides the sample into two parts based on this criterion. The right-hand branch of the tree goes to a box marked "Node 6." This refers to households where female asset

Figure 1 Classification tree

holdings exceed this value (hence "class"). There are 118 households in this class (N = 118), of which 92 are food secure and 26 are insecure. This node is further divided into two boxes, Terminal nodes 6 and 7, based on whether household size is less than or equal to, or greater than, 8.5 people. The left-hand split, Terminal node 6, includes those households where female assets are greater than 33825 FCFA *and* where household size is less than or equal to 8.5. It contains 93 households, virtually all of which are food secure. As it is not possible to find any variable that separates this subsample into any more homogeneous subgroups, CART terminates disaggregation at this level, hence this is a *terminal node*. The right-hand split, includes households where female assets are greater than 33825 FCFA and household size is greater than 8.5. Here the subsample is evenly divided between food-insecure and -secure households.

The left-hand split leading off from the root node contains households where female assets are less than or equal to 33825 FCFA. This contains 157 households, evenly divided between food-secure and food-insecure households. This group is then divided by a number of additional criteria, household size, whether they are residents of village 2 and the value of male assets. These disaggregations produce terminal nodes 1 through 5. Taken collectively, CART has divided this sample into seven mutually exclusive groups. Three of these groups (Terminal nodes 2, 4, and 6) are almost exclusively made up of food-secure households. In the remaining four groups, there is a slight predominance of food-insecure households. These disaggregations are based on four variables (female assets, household size, location, and male assets). It took CART less than 10 seconds to produce these results.

3. TECHNICAL DETAILS: COMPUTING REQUIREMENTS AND PROGRAMMING

CART is stand-alone software that can run under either DOS or Windows platforms. The software comes with two completely documented manuals (Steinberg and Colla 1995; Steinberg, Colla, and Martin 1998), which are very easy to follow. The first manual, the main user's guide (Steinberg and Colla 1995) provides a comprehensive background and conceptual basis to CART and the art-of-tree-structured data analysis, detailed listings and explanations of CART command modes, and discusses how to use CART techniques and interpret results. It also contains a number of examples and detailed discussions of these. The second manual (Steinberg, Colla, and

Martin 1998) is for Windows Operating systems. In addition to providing a detailed tutorial, the manual covers the use of menus, the graphic interface, and many other features that are specific to Windows environment (Windows 3.X, Windows 95/NT).

For the data analyst who is not familiar with CART, the Window's tutorial is a very good starting point to learn about CART. The tutorial provides a guided tour to perform CART analysis from a simple example, and introduces the analyst to the use of menus (the **FILE, VIEW, SELECT**, **MODEL, LIMIT, TREE**, and the **WINDOW** and **HELP** menus), the interactive "tree navigator," and many other features of Windows.

Although both the DOS and Windows versions produce the same output, there are several features of the Windows version that make it particularly attractive. Most notably, it provides a graphical display of the tree diagrams—Figure 1 is taken directly from the CART output. Under DOS, this diagram has to be prepared manually. Another useful feature of the Windows version is that, if the analyst is not satisfied with the optimal or minimum cost tree that is produced by CART, he/she can make use of a feature called "TREE NAVIGATOR" and immediately examine/explore different tree topologies from the sequence of trees provided, and pick another tree for analysis if he/she wishes to do so (Appendix 1 explains the usefulness of this feature). CART for the Windows user is not limited to only using menus and menu items. He/she can also write programs in batch mode and submit these for analysis.

Hardware and software requirements for CART are listed below in Table 2.

Before running CART, it is necessary to prepare the data set. This involves selecting variables for analysis and saving them to a separate file. This file can be either in SAS, SPSS, STATA, or virtually any other format. Since CART assumes that all of the variables in the sample are for use in the analysis, those variables not intended for analysis should not be included in CART file(s). It is possible to keep a few variables in the data set that can be excluded during CART session. But it is a good practice to keep the number of excludable variables as few as possible. This saved data file should be then converted to a SYSTAT file using DBMSCOPY or any other file translator that comes with CART software. As in SPSS/PC, SYSTAT files have extensions 'SYS' as in *.SYS. Therefore, proper documentation of CART files and the directories and subdirectories in which CART files reside are essential.

Table 2 Hardware and software requirements of CART for personal computer, and prices

Hardware and Software Requirements	
Operating Systems Supported: Windows 3.X/ 95/ NT, DOS	
Memory Requirements:	This may vary with versions of CART software. CART for Windows is compiled for machines with at least 32 megabytes of RAM. For Optimal performance, Pentium machines with at least 32 megabytes of RAM are recommended.
Hard Disk Space:	• A Minimum of 10 megabytes of free space for program files, additional disk space for scratch files (required space depends on the data set), and • Disk drive for reading 3 ½-inch disks.
Company name:	Salford Systems
Address:	8880 Rio San Diego Dr., Suit 1045 San Diego, California 92108 U.S.A.
Web Address:	http://www.salford-systems.com
Telephone:	(619) 543-8880
Fax:	(619) 543-8888
Technical Support:	Available either by telephone, fax or letter.
Number of variables and observations:	Computing requires a minimum of 16 megabytes of free memory. Number of observations and variables supported depend on the available memory.

Source: Fax message received from Salford Systems, February 1998, and
http://www.salford-systems.com/technical-CART.html, July 9, 1998.

The next step involves putting together the essential commands to run the software in a logical order. As the following example illustrates, the basic program is straightforward, consisting of only a few lines of code. These can be entered interactively or submitted as a batch job.

```
Use   'C:\Ifad\cart\mali11.sys'
exclude hhid
Category vill1 vill2 vill3 vill4 vill5 vill6 vill7 vill8 vill9 vill10  MPEST  MINPUT
MSEED                NONAGDUM REMIT FNNAGDUM   FINPUT  FLLABOR
FSEED  FPEST
           FFERT  MLLABOR   MFERT  CASHGIVE   CEMENT  CALSDUM
Model calsdum
Misclass cost = 1.4 class 0 as 1
Output 'c:\ifad\cart\output1.dat'
Build
```

The first line locates the data set to be used. The second tells CART to exclude one variables from its analysis, HHID. The third line indicates which variables are categorical variables. The next line specifies the dependent variable, here Calsdum. The Misclass cost line specifies the penalty associated for misclassifying class 0 households as class 1 households. Inclusion of the OUT command sends the results to a file, here c:\ifad\CART\output1.DAT. Finally, BUILD tells CART to produce a classification tree. These commands, and further options are outlined in Table 3.

4. FURTHER APPLICATIONS, STRENGTHS, AND WEAKNESSES OF CART

There are two important further applications of CART. First, it can provide a means of understanding household food insecurity at the household level. In this case, all variables are expressed at the household, rather than locality level. Though some caution is needed in interpreting these results—CART produces correlates of food insecurity rather than uncovering causal links, these can be useful during the exploratory phase of work. Second, CART has been used extensively in the commercial finance field as a tool for determining who is most likely to apply, receive, and default on credit. Drawing on an established data base, CART can identify what individual-, household-, or locality-level characteristics are associated with say, a higher rate of loan application or of default. This information could then be fed back into program design.

CART's strengths lie in two areas. Practically, it is easy to install and run. Once the data base is established, a simple program generates the results in an easy to understand format. In addition

1. CART makes no distributional assumptions of any kind, either on dependent or independent variables. No variable in CART is assumed to follow any kind of statistical distribution.
2. The explanatory variables in CART can be a mixture of categorical, interval, and continuous.

Table 3 Basic CART software commands in SYSTAT

Command	Syntax	Function (Purpose)	Examples
USE	USE *filename*	Specifies to CART a file to read	USE c:\cart\test1.sys
EXCLUDE	EXCLUDE *variable list*	Excludes from file the variables not needed in the analysis	EXCLUDE hhid code
KEEP	KEEP *variable list*	Reads from the file only the variables needed in the analysis	KEEP age sex income
CATEGORY	Category *variable list*	Specifies to CART list of categorical variables in the data set, including the dependent variable in Classification tree; this is compulsory.	CATEGORY sex
MODEL	MODEL *variable name*	Specifies dependent variable	MODEL vulner
BUILD	BUILD	Tells CART to produce a tree	BUILD
QUIT	QUIT	If submitted while in Build, it tells CART to quit the session; if submitted after CART session, it tells CART to go to DOS.	
SELECT	SELECT *variable name* relation operator or constant/character Or	Selects a subset of the data set for analysis	SELECT age> 15 SELECT sex=1 SELECT X>=20 Select x1='M'
SELECT	SELECT *variable name* relation operator or constant/character	Selects a subset of the data set for analysis	SELECT age > 15, Wage >300
PRIORS	PRIORS *option* (Choose 1 option only)	Specifies to CART which priors to use	PRIORS data PRIORS equal PRIORS mixed PRIORS=n1, n2,,..,na (n's are real numbers)
MISCLASS COST	Misclass cost=n classify I as k1,k2,k3/, Cost=m classify I as k1/, Cost=l classify k1,k2,..,kn as x	Assigns non unit misclassification costs	Misclass cost=2 classify 1 as 2,3,4/, Cost=5 classify 3 as 1 Cost=3 classify 1,2,3 as 4
METHOD	Method=options (choose 1 option only)	Specifies splitting rule	Method=gini(default) or Method=twoing or Method=LS or LAD Method=LINEAR
OUTPUT	OUTPUT filename	Sends output to a named file	OUTPUT=LMS
TREE	TREE tree filename	Specifies a file name of a tree to be saved	TREE Vulner1
SAVE	SAVE filename options with predicted class(es), select options to save	Specifies filename for a data set	SAVE predct1
CASE	CASE options	Runs data one-by-one down a tree, select option(s) to use	CASE

3. CART has a built-in algorithm to deal with the missing values of a variable for a case, except when a linear combination of variables is used as a splitting rule.
4. CART is not at all affected by outliers, collinearities, heteroscedasticity, or distributional error structures that affect parametric procedures. Outliers are isolated into a node, and do not have any effect on splitting. Contrary to situations in parametric modeling, CART makes use of collinear variables in "surrogate" split(s).
5. CART has the ability to detect and reveal interactions in the data set.
6. CART is invariant under monotone transformation of independent variables; that is, the transformation of explanatory variables to logarithms or squares or square roots has no effect on the tree produced.
7. CART's effectively deals with higher dimensionality; that is, from a large number of variables submitted for analysis, it can produce useful results using only a few important variables.

An important weakness of CART is that it is not based on a probabilistic model. There is no probability level or confidence interval associated with predictions derived from using a CART tree to classify a new set of data. The confidence that an analyst can have in the accuracy of the results produced by a given model (that is, a tree) is based purely on its historical accuracy—how well it has predicted the desired response in other, similar circumstances.

APPENDIX 1
TECHNICAL DETAILS: BUILDING A CLASSIFICATION TREE

The previous section has provided an extended introduction to CART. In this section, we provide a more detailed and more technical explanation as to how CART builds these classification and regression trees.

The tree building process starts by partitioning a sample or the "root node" into binary nodes based upon a very simple question of the form: is $X \leq d$? where X is a variable in the data set, and d is a real number. Initially, all observations are placed at the root node. This node is impure or heterogenous since it contains observations of, say both food-secure and food-insecure localities. The goal is to devise a rule that will initially break up these observations and create groups or binary nodes that are internally more homogenous than the root node. These disaggregations, or splits from the root node, are generated in the following fashion.

1. Starting with the first variable, CART splits a variable at all of its possible split points (at all of the values the variable assumes in the sample). At each possible split point of the variable, the sample splits into two binary or child nodes. Cases with a "yes" response to the question posed are sent to the left node and the "no" responses are sent to the right node. It is also possible to define these splits based on linear combinations of variables.
2. CART then applies its goodness of a split criteria to each split point and evaluates the reduction in impurity, or heterogeneity due to the split. This is based on the *goodness of split criterion*. This works in the following fashion. Suppose the dependent variable is categorical, taking on the value of 1 (if, say a locality is food secure) and 2 (if the locality is food insecure). The probability distributions of these variables at a given node t are $p(1|t)$ and $p(2|t)$, respectively. A measure of heterogeneity, or impurity at node, $i(t)$ is a function of these probabilities, $i(t) = \phi(p(1|t), p(2|t))$. Clearly, $i(t)$ is a generic function. In the case of categorical dependent variables, CART allows for a number of specifications of this function. The objective is to maximize the reduction in the degree of heterogeneity in $i(t)$.

3. It selects the best split on the variable as that split for which reduction in impurity is the highest, as described above.
4. Steps 1-3 are repeated for each of the remaining variables at the root node. CART then ranks all of the "best" splits on each variable according to the reduction in impurity achieved by each split.
5. It selects the variable and its split point that most reduced impurity of the root or parent node.
6. CART then assigns classes to these nodes according to a rule that minimizes misclassification costs. Although all classification tree procedures will generate some errors, there are algorithms within CART designed to minimize these. For example, in famine vulnerability, misclassifying a vulnerable household as a nonvulnerable might be considered a more severe error than misclassifying a nonvulnerable household as vulnerable. It is possible for the user to define a matrix of variable misclassification costs that recognizes such costs, which are then incorporated into the splitting rule(s). Alternatively, the analyst can use the default category of assuming that all misclassifications are equally costly.
7. Steps 1-6 are repeatedly applied to each non-terminal child node at each of the successive stages.
8. CART continues the splitting process and builds a large tree. The largest tree can be achieved if the splitting process continues until every observation constitutes a terminal node. Obviously, such a tree will have a large number of terminal nodes that are either pure or very small in content.

Having generated a large tree, CART then prunes the results using cross-validation and creates a sequence of nested trees. This also produces a cross-validation error rate and from this, the optimal tree is selected.

APPENDIX 2
SAMPLE CART OUTPUT

This appendix provides an annotated example of output from a CART program.

CART Batch Program Code

```
      Use    C:\Ifad\cart\mali11.sys
      exclude hhid
      Category   vill1 vill2 vill3 vill4 vill5 vill6 vill7 vill8 vill9 vill10
MPEST
                 MINPUT  MSEED NONAGDUM REMIT
FNNAGDUM
                 FINPUT  FLLABOR   FSEED  FPEST FFERT
MLLABOR
                 MFERT  CASHGIVE   CEMENT  CALSDUM
      Model calsdum
      Misclass cost = 1.4 class 0 as 1
      Output  c:\ifad\cart\output1.dat
      Build
```

This program produces an optimal tree with seven terminal nodes.

CART Output Report (partial output for illustrative purposes)

```
RECORDS READ: 275
RECORDS WRITTEN IN LEARNING SAMPLE: 275

LEARNING SAMPLE VARIABLE STATISTICS
===========================
                              CLASS
VARIABLE                      0          1
                            OVERALL

-------------------------------------------------------------
-----------------
  VILL1   MEAN       0.062      0.067      0.065
            SD       0.242      0.251      0.248
             N          97        178        275
           SUM       6.000     12.000     18.000
 HHSIZEMEAN         6.897      5.202      5.800
            SD       4.091      3.163      3.604
             N          97        178        275
           SUM     669.000    926.000   1595.000
 REMIT MEAN         0.309      0.275      0.287
            SD       0.465      0.448      0.453
             N          97        178        275
           SUM      30.000     49.000     79.000
```

AUTOMATIC LEVEL SETTINGS
(partial output for illustrative purpose)

- **This output is only for categorical variables declared in the category command line in the program.**

NAME	LEVELS	MINIMUM
VILL1	2	0
VILL2	2	0
REMIT	2	0
MFERT	2	0
FNNAGDUM	2	0
CALSDUM	2	0

MIX PRIORS: 0.426 0.574

- **These are priors used in this analysis (Mixed priors). The probability of observing the food insecure in the population is 43% where as the probability of observing the food secure group in the population is 57%.**

CURRENT MEMORY REQUIREMENTS
TOTAL: 48434 DATA: 10725 ANALYSIS: 37709
AVAILABLE: 2000000
SURPLUS: 1951566

BUILD PREPROCESSOR CPU TIME: 00:00:00.33

275 Observations in the learning sample.
File: C:\MALI\CART\MALI11.SYS

- **Location of the file the data is read from.**

Tree 1 of 11 CPU TIME: 00:00:00.81

Cross Validation CPU TIME: 00:00:06.48

==========
TREE SEQUENCE
==========

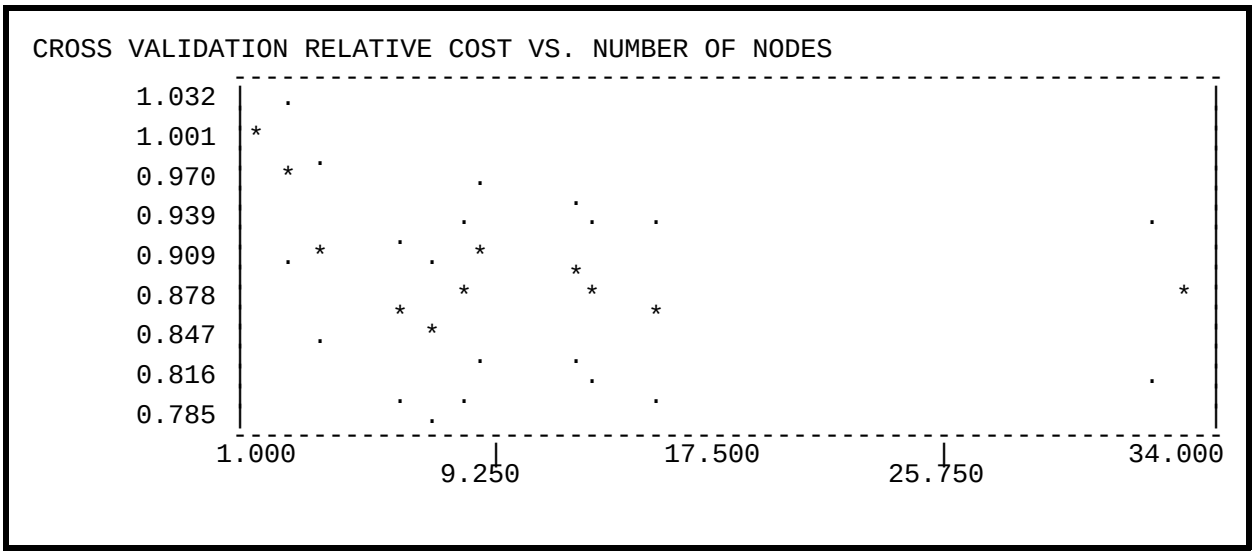
● **This is a sequence of subtrees generated from the largest tree by pruning and cross-validation test.**

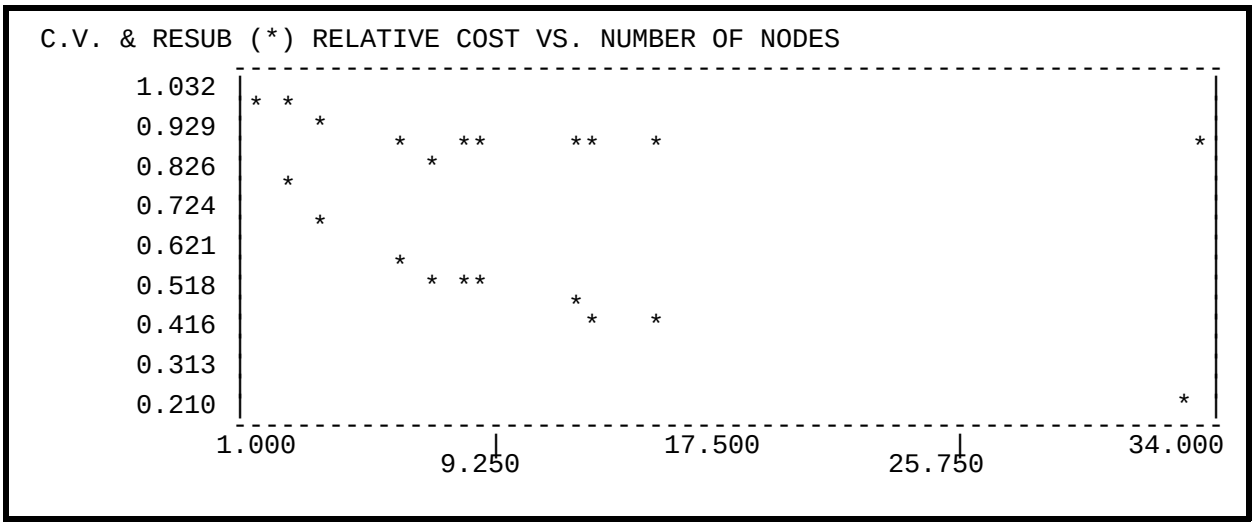
Dependent variable: CALSDUM

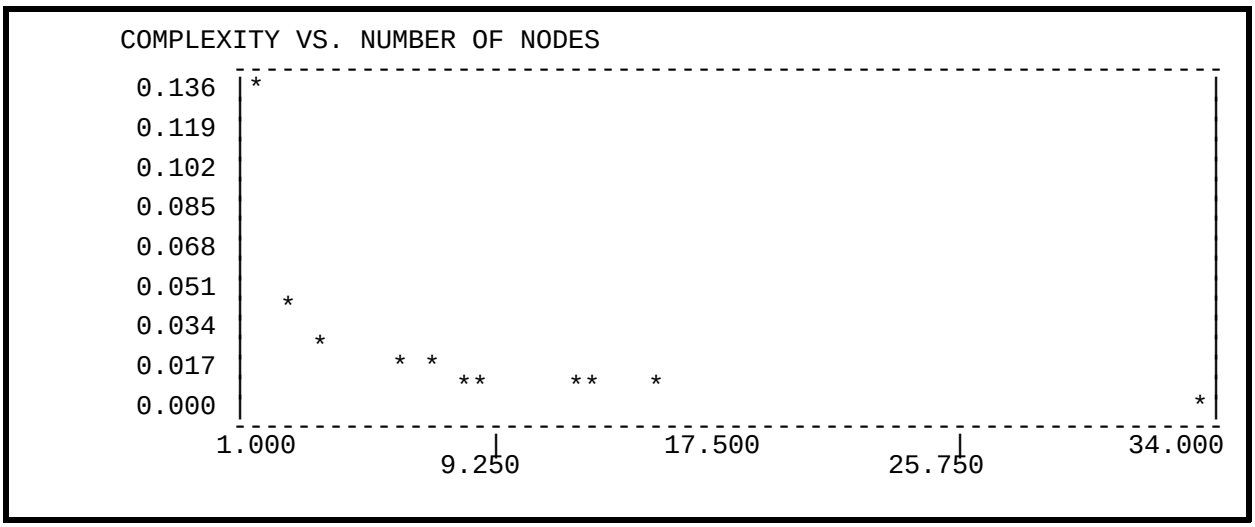
Tree	Terminal Nodes	Cross-Validated Relative Cost	Resubstitution Relative Cost	Complexity Parameter
1	34	0.873 +/- 0.064	0.210	0.000
9	15	0.870 +/- 0.064	0.400	0.008
10	13	0.876 +/- 0.064	0.432	0.009
11	12	0.898 +/- 0.064	0.448	0.010
12	9	0.903 +/- 0.064	0.502	0.010
13	8	0.872 +/- 0.064	0.519	0.010
14**	7	0.849 +/- 0.064	0.542	0.013
15	6	0.862 +/- 0.064	0.573	0.018
16	3	0.916 +/- 0.064	0.690	0.022
17	2	0.967 +/- 0.064	0.762	0.041
18	1	1.000 +/- 0.000	1.000	0.136

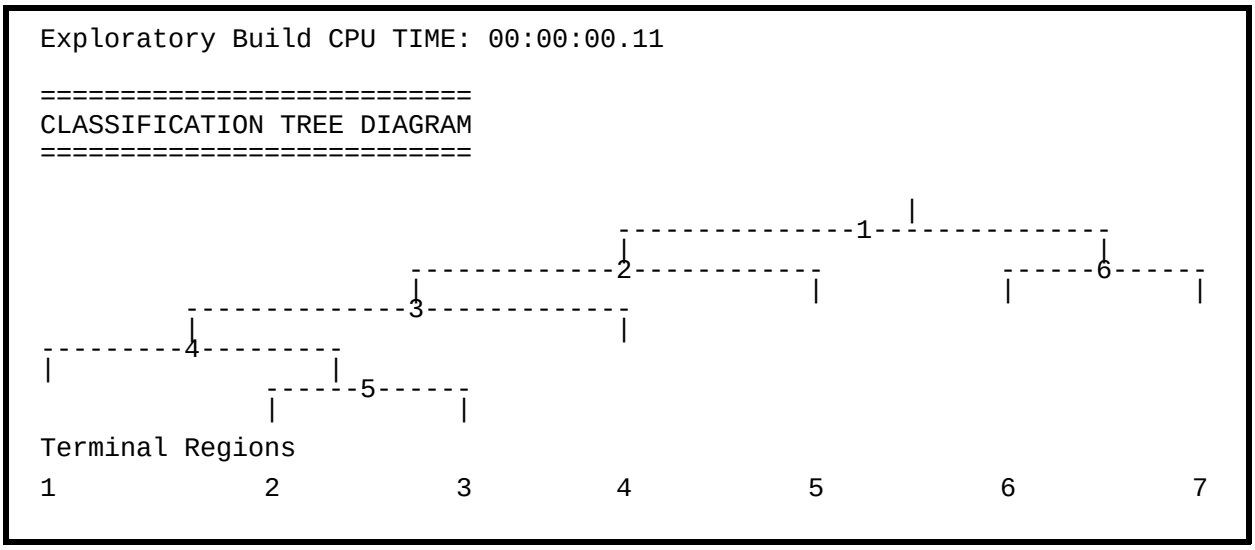
Initial misclassification cost = 0.574
Initial class assignment = 0

● **Each tree is identified either by the number under the tree column or by the number of nodes under the Nodes column. Usually, a minimum cost tree is identified by a single asterisk(*) while the optimal cost tree is identified by double asterisk(**). In this example, the minimum cost tree is itself an optimal tree. For each tree, Cross-validated relative cost and Resubstitution relative cost are provided. The cross-validated relative cost is the misclassification cost generated by the application of cross-validation test while pruning the tree. The Resubstitution relative cost is the misclassification cost generated by using the learning sample as a test sample. As the number of nodes increase, the cross validation cost initially drops, reaches a minimum, and then starts rising. The tree for which the cross-validated cost is the minimum is the minimal cost tree . The resubstitution relative cost keeps decreasing as the number of nodes increase. This cost behaves (although in reverse direction) just like an R-square in regression where R-square keeps increasing as the number of variables are added into the model. The following graphs show these behaviors. The Complexity parameter column depicts, the complexity values used by CART in the tree pruning algorithm.**


CROSS VALIDATION RELATIVE COST VS. NUMBER OF NODES
1.032
1.001
0.970
0.939
0.909
0.878
0.847
0.816
0.785
1.000
9.250
17.500
25.750
34.000


C.V. & RESUB (*) RELATIVE COST VS. NUMBER OF NODES
1.032
0.929
0.826
0.724
0.621
0.518
0.416
0.313
0.210
1.000
9.250
17.500
25.750
34.000


COMPLEXITY VS. NUMBER OF NODES
0.136
0.119
0.102
0.085
0.068
0.051
0.034
0.017
0.000
1.000
9.250
17.500
25.750
34.000


Exploratory Build CPU TIME: 00:00:00.11
CLASSIFICATION TREE DIAGRAM
1
2
3
4
5
6
Terminal Regions
1 2 3 4 5 6 7

==========
NODE INFORMATION
==========

Only 1st and last node splits are provided here for illustrative purposes.

```
       *          Node  1 was split on variable ASSTV1F1
      * *         A case goes left if variable ASSTV1F1 <=    33825.000
     *   *        Improvement =  0.032        C. T. =  0.136
    *  1  *
     *   *        Node     Cases     Class        Cost
      * *          1       275         0         0.574
       *           2       157         0         0.470
      * *          6       118         1         0.390
     *   *
   157   118            Number Of Cases          Within Node Prob.
    *     *     Class   Top  Left  Right       Top      Left     Right
     *   *        0     97    71    26        0.426    0.530    0.278
    *     *       1    178    86    92        0.574    0.470    0.722
   * *   * *
  *   *   *   *    Surrogate              Split          Assoc.    Improve.
 *   2 *   * 6 *  1 ASSTVF1        s    47425.000        0.714     0.020
  *   *     *   *  2 ASSTVM1        s    33000.000        0.138     0.010
   * *       * *   3 VILL9          s           0         0.134     0.016
    *         *    4 VACVM1         s    37500.000        0.128     0.003
                   5 VACNUMM1       s        0.500        0.114     0.004

                    Competitor      Split          Improve.
                  1 HHSIZE          8.500           0.031
                  2 VILL10          1               0.024
                  3 ASSTVF1         49887.500       0.021
                  4 VILL7           1               0.016
                  5 VILL9           0               0.016
==========
```

- **Improvement reduction in impurity due to the split.**
- **Describes distribution of cases and classes in different nodes along with associated costs.**
- **Within node number of cases and class probability distribution.**

- **Diamond shapes in the figure indicate those nodes are not terminal nodes.**

- **Surrogate These are proxy variables for the main splitting variable. At each node, they mimic the split of the main variable used in the split. Surrogates are expected to split the sample into left and right nodes such that within a node, composition of the cases and class distribution is very similar to the primary splitting variable. They are also useful especially in situations where a case has missing value for a splitting variable. Letter s indicates that a split is a standard one. A split is called standard if, for example, as in the above, cases with ASSTV1F1 <= 33825.000 are sent to the left node, while cases with ASSTV1F1 > 33825.000 are sent to the right. The association (Assoc.) column measures the extent of the surrogate s variable capability to mimic or predict the actions of the primary splitting variable. It sounds like a correlation coefficient, but it is not. By default, CART produces 5 surrogate variables.**

● **Improvement (Improv.) Column indicates the reduction in impurity that could have been achieved had the variable been used as a splitting variable.**

● **Competitor These are competing variables with the primary splitting variable. Had anyone of these variables been used for splitting, the point at which the split could occur (Split column) and the improvement that would have been achieved (Improve. column) due to the split are provided. By default, CART produces 5 competitors. They are ranked based on the improvement each could yield. If the variables HHSIZE had been used as the 1st primary splitting variable, the split could have occurred at 8.500 and the reduction in impurity could have been 0.031. The level of improvement in impurity reduction is slightly less than the improvement obtained by the primary splitting variable ASSTV1F1.**

● **Descriptions for node 6 below are similar to the one given above.**

```
      *          Node  6 was split on variable HHSIZE
     * *         A case goes left if variable HHSIZE <=       8.500
    *   *        Improvement =  0.029         C. T. =  0.041
   *  6  *
    *   *        Node     Cases     Class     Cost
     * *           6       118         1      0.390
      *          -6        93         1      0.254
     * *         -7        25         0      0.404
    *   *
   *     *
   93    25      Number Of Cases              Within Node Prob.
   *      *     Class  Top  Left  Right       Top      Left      Right
  *       *      0     26    13     13        0.278    0.181     0.596
---*-    --*--   1     92    80     12        0.722    0.819     0.404
|    |    |    |
|    |    |    |   Surrogate                 Split          Assoc.      Improve.
| 6  |    | 7  |  1 BOENUMM1    s             1.500         0.318       0.004
|    |    |    |  2 BOEVM1      s         90000.000         0.284       0.004
|    |    |    |  3 ASSTVM1     s        204425.000         0.196       0.005
-----    ------   4 VACNUMM1    s             3.500         0.159    .332475E-03
                  5 VACVM1      s        225000.000         0.113    .108466E-03

                   Competitor      Split           Improve.
                  1 ASSTVM1     25375.000          0.023
                  2 DEPRAT          0.649          0.013
                  3 VILL10              1          0.010
                  4 ASSTV2F2    20750.000          0.005
                  5 ASSTV1F1   284575.000          0.005
==========
```

● **Negative numbers under Node column indicate a terminal node. Nodes 6 and 7 are terminal.**

● **Rectangular boxes indicate those nodes are terminal. They do not split any further.**

==========
TERMINAL NODE INFORMATION
==========

[Breiman adjusted cost, lambda = 0.035]

Node	N	Prob	Class	Cost	Class	N	Prob	Complexity Threshold
1	25	0.097	0	0.366	0	14	0.634	0.031
				[0.820]	1	11	0.366	
2	21	0.070	1	0.176	0	2	0.126	0.018
				[0.747]	1	19	0.874	
3	12	0.046	0	0.423	0	6	0.577	0.018
				[1.165]	1	6	0.423	
4	17	0.057	1	0.215	0	2	0.154	0.025
				[0.866]	1	15	0.846	
5	82	0.319	0	0.353	0	47	0.647	0.022
				[0.523]	1	35	0.353	
6	93	0.315	1	0.254	0	13	0.181	0.041
				[0.426]	1	80	0.819	
7	25	0.096	0	0.404	0	13	0.596	0.041
				[0.862]	1	12	0.404	

● **This table provides information for each terminal node. Among other things, it provides the number of terminal nodes ('Node' column), number of cases at each terminal node ('N' column), weighted probability of reaching the node ('Prob' column, if priors were data, no need for weighting), predicted class for the node 'Class' column, misclassification costs weighted by priors ('Cost' column), class distribution of the cases within each node ('Class column) and the number of cases within each class, weighted probability distribution of the cases within each class at a node, and finally the complexity parameter used to arrive at each node via pruning ('Complexity Threshold' column).**

==========
MISCLASSIFICATION BY CLASS
==========

		------CROSS VALIDATION------				-----LEARNING SAMPLE--------		
Class	Prior Prob.	N N	Mis-Classified	Cost	N N	Mis-Classified	Cost
0	0.426	97	42	0.606	97	17	0.245
1	0.574	178	71	0.399	178	64	0.360
Tot	1.000	275	113		275	81	

● The misclassification by class table summarizes the number of cases in each class in the data and prediction of the number of cases misclassified ('N Misclassified ' column) by class from cross validation test as well as from learning sample. The table is for the entire tree and consists of two panels 'CROSS VALIDATION' AND 'LEARNING SAMPLE'. The learning sample is the sample that produced the tree.

● The performance of the tree was tested using cross validation. The results are shown under 'CROSS VALIDATION' panel. The number of cases misclassified is an estimate or 'best' prediction of cases that would be misclassified if the tree is applied to new data set.

● Under the Learning Sample panel, the number of cases misclassified is generated by dropping down the tree all cases in the learning sample, and counting the number of cases misclassified from each class. The 'Cost' column shows the misclassification cost for each class.

● **The following two tables provide detailed information on Cross validation. The outcome is similar to the tables generated from Logistic and Probit Models. Entries along the diagonals of the matrix represent correct classification, while off diagonal entries represent misclassification. The tables help the analyst see where misclassifications are actually occurring.**

```
====================================
CROSS VALIDATION CLASSIFICATION TABLE
====================================

     ACTUAL          PREDICTED CLASS            ACTUAL
      CLASS            0                1        TOTAL
-------------------------------------------------------------------------
        0             55.000          42.000      97.000
        1             71.000         107.000     178.000
-------------------------------------------------------------------------
PRED. TOT.          126.000         149.000     275.000
CORRECT               0.567           0.601
SUCCESS IND.          0.214          -0.046
TOT. CORRECT          0.589

   SENSITIVITY:       0.567    SPECIFICITY:        0.601
FALSE REFERENCE:  0.563   FALSE RESPONSE:   0.282
REFERENCE = CLASS 0, RESPONSE = CLASS 1
-------------------------------------------------------------------------


================================================
CROSS VALIDATION CLASSIFICATION PROBABILITY TABLE
================================================

     ACTUAL      PREDICTED CLASS      ACTUAL
      CLASS            0                1        TOTAL
-------------------------------------------------------------------
        0             0.567           0.433       1.000
        1             0.399           0.601       1.000
-------------------------------------------------------------------
```

● **This is the most useful table. It is derived from the above table. The goodness of any classification tree is judged from the entries in this table. It helps the analyst either to retain the current tree or think of refining the tree.**

```
====================================
LEARNING SAMPLE CLASSIFICATION TABLE
====================================

     ACTUAL        PREDICTED CLASS     ACTUAL
      CLASS             0            1          TOTAL
-------------------------------------------------------------------------------
         0            80.000       17.000       97.000
         1            64.000      114.000      178.000
-------------------------------------------------------------------------------
PRED. TOT.          144.000      131.000      275.00
CORRECT               0.825        0.640
SUCCESS IND.          0.472       -0.007
TOT. CORRECT          0.705

   SENSITIVITY:       0.825     SPECIFICITY:       0.640
FALSE REFERENCE:  0.444   FALSE RESPONSE:   0.130
REFERENCE = CLASS 0, RESPONSE = CLASS 1
-------------------------------------------------------------------------------
```

- **The description given for Cross validation classification table holds. However, these predictions are generated by using the learning sample as a test sample.**

```
================================================
LEARNING SAMPLE CLASSIFICATION PROBABILITY TABLE
================================================

     ACTUAL     PREDICTED CLASS     ACTUAL
      CLASS        0         1        TOTAL
-----------------------------------------------------------
         0         0.825     0.175      1.000
         1         0.360     0.640      1.000
-----------------------------------------------------------
```

- **This summary is produced from the above table. Diagonal entries are probabilistic predictions of correct classification. Again, these probability predictions are based upon the application of the tree to the learning sample. Predictions based on the learning sample underestimate 'true' misclassification rates. Predicted misclassification rates based on the data set from which the tree is constructed and are not good indicators of the predictive accuracy of a tree.**

==========
VARIABLE IMPORTANCE
==========

	Relative Importance	Number of Categories	Minimum Category
HHSIZE	100.000		
ASSTVM1	94.499		
ASSTV1F1	77.578		
ASSTVF1	50.548		
DEPRAT	36.527		
VILL9	33.614	2	0
VILL2	20.054	2	0
VILL10	14.715	2	0
MPEST	9.616	2	0
VACNUMM1	9.100		
BOENUMM1	7.784		
BOEVM1	7.625		
VACVM1	6.290		
MINPUT	3.663	2	0
MSEED	3.663		
ASSTV2F2	2.522		
CASHSENT	1.357	2	0
NONAGDUM	1.286	2	0
REMIT	0.120	2	0
ADTVF1	0.000		
ADTNUMF1	0.000		
VILL1	0.000	2	0
FNNAGDUM	0.000	2	0
VILL3	0.000	2	0
FINPUT	0.000	2	0
FLLABOR	0.000	2	0
FSEED	0.000	2	0
FPEST	0.000	2	0
FFERT	0.000	2	0
MLLABOR	0.000	2	0
MFERT	0.000	2	0
CASHGIVE	0.000	2	0
VILL4	0.000	2	0
VILL5	0.000	2	0
VILL6	0.000	2	0
VILL7	0.000	2	0
VILL8	0.000	2	0
CEMENT	0.000		

- **Variable importance table provides list of all the variables used and not used in the tree building process. A score is attached to each variable, and is based on the improvement each variable makes as a surrogate to the primary splitting variable. Variable importance measure gives due recognition to the variables whose significance is masked or hidden by other variables in the tree building process.**

==========
MISCLASSIFICATION COSTS
==========

- **Table of misclassification costs used in this analysis.**

Class	Cost if classified as 0	1
0	0.000	1.400
1	1.000	0.000

Total CPU TIME: 00:00:07.58

REFERENCES

Borton, J., and J. Shoham. 1991. *Mapping vulnerability to food insecurity: Tentative guidelines for WFP country offices*. London: Relief and Development Institute.

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. 1984. *Classification and regression trees*. Monterey, Calif., U.S.A.: Wadsworth, Inc.

Currey, B. 1978. Mapping of areas liable to famine in Bangladesh. Ph.D. thesis (unpublished). Geography Department, University of Hawaii, Honolulu.

Downing, T. E. 1993. Regions/vulnerable groups in FEWS methodology. Memorandum. Rosslyn, Va., U.S.A.

Frankenberger, T. 1992. Indicators and data collection methods for assessing household food security. In *Household food security: Concepts, indicators, and methods*, ed. S. Maxwell and T. Frankenberger. Rome: United Nations Childrens Fund/International Fund for Agricultural Development.

Riely, F. 1993. Vulnerability analysis in the FEWS project. Report to the United States Agency for International Development. Tulane University, New Orleans, La., U.S.A. Mimeo.

Seaman, J., J. Holt, and P. Allen. 1993. A new approach to vulnerability mapping for areas at risk of food crisis. Interim report on the Risk-Mapping Project. London. Mimeo.

Seyoum, S., E. Richardson, P. Webb, F. Riely, and Y. Yohannes. 1995. Analyzing and mapping food insecurity: An exploratory CART methodology applied to Ethiopia. Final report to the United States Agency for International Development. International Food Policy Research Institute, Washington, D.C. Mimeo.

Steinberg, D., and P. Colla. 1995. *CART: Tree-structured non-parametric data analysis*. San Diego, Calif., U.S.A.: Salford Systems.

Steinberg, D., P. Colla, and K. Martin. 1998. *CART Classification and regression trees: Supplementary manual for Windows*. San Diego, Calif., U.S.A.: Salford Systems.

Yohannes, Y., and P. Webb. 1998. Classification and regression trees: A user manual for identifying indicators of vulnerability to famine and chronic food insecurity. International Food Policy Research Institute, Washington, D.C. Mimeo.