

For more information about SPSS Inc. software products, please visit our Web site at <http://www.spss.com> or contact

SPSS Inc.

233 South Wacker Drive, 11th Floor

Chicago, IL 60606-6412

Tel: (312) 651-3000

Fax: (312) 651-3668

SPSS is a registered trademark and the other product names are the trademarks of SPSS Inc. for its proprietary computer software. No material describing such software may be produced or distributed without the written permission of the owners of the trademark and license rights in the software and the copyrights in the published materials.

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of The Rights in Technical Data and Computer Software clause at 52.227-7013. Contractor/manufacturer is SPSS Inc., 233 South Wacker Drive, 11th Floor, Chicago, IL 60606-6412.

Patent No. 7,023,453

General notice: Other product names mentioned herein are used for identification purposes only and may be trademarks of their respective companies.

Windows is a registered trademark of Microsoft Corporation.

Apple, Mac, and the Mac logo are trademarks of Apple Computer, Inc., registered in the U.S. and other countries.

This product uses WinWrap Basic, Copyright 1993-2007, Polar Engineering and Consulting, <http://www.winwrap.com>.

Printed in the United States of America.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

Preface

SPSS Statistics 17.0 is a comprehensive system for analyzing data. The Decision Trees optional add-on module provides the additional analytic techniques described in this manual. The Decision Trees add-on module must be used with the SPSS Statistics 17.0 Base system and is completely integrated into that system.

Installation

To install the Decision Trees add-on module, run the License Authorization Wizard using the authorization code that you received from SPSS Inc. For more information, see the installation instructions supplied with the Decision Trees add-on module.

Compatibility

SPSS Statistics is designed to run on many computer systems. See the installation instructions that came with your system for specific information on minimum and recommended requirements.

Serial Numbers

Your serial number is your identification number with SPSS Inc. You will need this serial number when you contact SPSS Inc. for information regarding support, payment, or an upgraded system. The serial number was provided with your Base system.

Customer Service

If you have any questions concerning your shipment or account, contact your local office, listed on the Web site at <http://www.spss.com/worldwide>. Please have your serial number ready for identification.

Training Seminars

SPSS Inc. provides both public and onsite training seminars. All seminars feature hands-on workshops. Seminars will be offered in major cities on a regular basis. For more information on these seminars, contact your local office, listed on the Web site at <http://www.spss.com/worldwide>.

Technical Support

Technical Support services are available to maintenance customers. Customers may contact Technical Support for assistance in using SPSS Statistics or for installation help for one of the supported hardware environments. To reach Technical Support, see the Web site at <http://www.spss.com>, or contact your local office, listed on the Web site at <http://www.spss.com/worldwide>. Be prepared to identify yourself, your organization, and the serial number of your system.

Additional Publications

The *SPSS Statistical Procedures Companion*, by Marija Norušis, has been published by Prentice Hall. A new version of this book, updated for SPSS Statistics 17.0, is planned. The *SPSS Advanced Statistical Procedures Companion*, also based on SPSS Statistics 17.0, is forthcoming. The *SPSS Guide to Data Analysis* for SPSS Statistics 17.0 is also in development. Announcements of publications available exclusively through Prentice Hall will be available on the Web site at <http://www.spss.com/estore> (select your home country, and then click Books).

Contents

Part I: User's Guide

1 Creating Decision Trees 1

Selecting Categories	7
Validation	9
Tree-Growing Criteria	10
Growth Limits	11
CHAID Criteria	12
CRT Criteria	15
QUEST Criteria	16
Pruning Trees	17
Surrogates	18
Options	19
Misclassification Costs	19
Profits	21
Prior Probabilities	23
Scores	25
Missing Values	27
Saving Model Information	29
Output	30
Tree Display	31
Statistics	34
Charts	39
Selection and Scoring Rules	45

2 Tree Editor 48

Working with Large Trees	50
Tree Map	50
Scaling the Tree Display	51
Node Summary Window	52
Controlling Information Displayed in the Tree	53
Changing Tree Colors and Text Fonts	54
Case Selection and Scoring Rules	57
Filtering Cases	57
Saving Selection and Scoring Rules	58

Part II: Examples

3 Data Assumptions and Requirements 61

Effects of Measurement Level on Tree Models	61
Permanently Assigning Measurement Level	65
Effects of Value Labels on Tree Models	66
Assigning Value Labels to All Values	68

4 Using Decision Trees to Evaluate Credit Risk 70

Creating the Model	70
Building the CHAID Tree Model	70
Selecting Target Categories	71
Specifying Tree Growing Criteria	72

Selecting Additional Output	73
Saving Predicted Values	76
Evaluating the Model	76
Model Summary Table	77
Tree Diagram	78
Tree Table	79
Gains for Nodes	80
Gains Chart	82
Index Chart	83
Risk Estimate and Classification	84
Predicted Values	85
Refining the Model	86
Selecting Cases in Nodes	86
Examining the Selected Cases	88
Assigning Costs to Outcomes	92
Summary	96

5 Building a Scoring Model 97

Building the Model	97
Evaluating the Model	100
Model Summary	100
Tree Model Diagram	102
Risk Estimate	104
Applying the Model to Another Data File	105
Summary	108

6 Missing Values in Tree Models 109

- Missing Values with CHAID 110
 - CHAID Results 113
- Missing Values with CRT 114
 - CRT Results 117
- Summary 119

Appendix

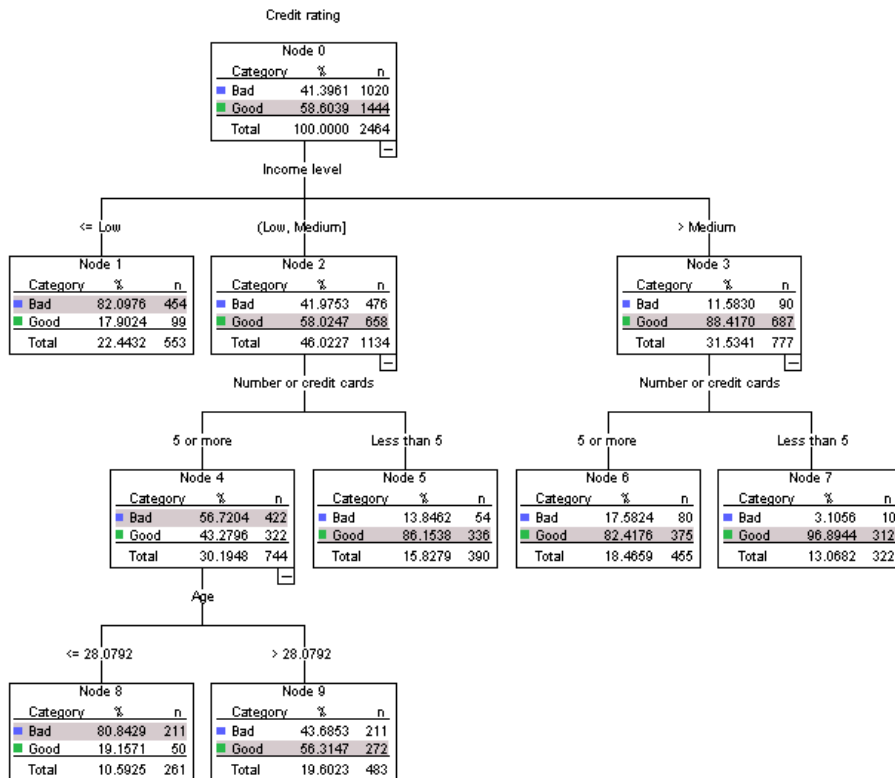
A Sample Files 120

Index 134

Part I:
User's Guide

Creating Decision Trees

Figure 1-1
Decision tree



The Decision Tree procedure creates a tree-based classification model. It classifies cases into groups or predicts values of a dependent (target) variable based on values of independent (predictor) variables. The procedure provides validation tools for exploratory and confirmatory classification analysis.

The procedure can be used for:

Segmentation. Identify persons who are likely to be members of a particular group.

Stratification. Assign cases into one of several categories, such as high-, medium-, and low-risk groups.

Prediction. Create rules and use them to predict future events, such as the likelihood that someone will default on a loan or the potential resale value of a vehicle or home.

Data reduction and variable screening. Select a useful subset of predictors from a large set of variables for use in building a formal parametric model.

Interaction identification. Identify relationships that pertain only to specific subgroups and specify these in a formal parametric model.

Category merging and discretizing continuous variables. Recode group predictor categories and continuous variables with minimal loss of information.

Example. A bank wants to categorize credit applicants according to whether or not they represent a reasonable credit risk. Based on various factors, including the known credit ratings of past customers, you can build a model to predict if future customers are likely to default on their loans.

A tree-based analysis provides some attractive features:

- It allows you to identify homogeneous groups with high or low risk.
- It makes it easy to construct rules for making predictions about individual cases.

Data Considerations

Data. The dependent and independent variables can be:

- **Nominal.** A variable can be treated as nominal when its values represent categories with no intrinsic ranking (for example, the department of the company in which an employee works). Examples of nominal variables include region, zip code, and religious affiliation.

- **Ordinal.** A variable can be treated as ordinal when its values represent categories with some intrinsic ranking (for example, levels of service satisfaction from highly dissatisfied to highly satisfied). Examples of ordinal variables include attitude scores representing degree of satisfaction or confidence and preference rating scores.
- **Scale.** A variable can be treated as scale when its values represent ordered categories with a meaningful metric, so that distance comparisons between values are appropriate. Examples of scale variables include age in years and income in thousands of dollars.

Frequency weights If weighting is in effect, fractional weights are rounded to the closest integer; so, cases with a weight value of less than 0.5 are assigned a weight of 0 and are therefore excluded from the analysis.

Assumptions. This procedure assumes that the appropriate measurement level has been assigned to all analysis variables, and some features assume that all values of the dependent variable included in the analysis have defined value labels.

- **Measurement level.** Measurement level affects the tree computations; so, all variables should be assigned the appropriate measurement level. By default, numeric variables are assumed to be scale and string variables are assumed to be nominal, which may not accurately reflect the true measurement level. An icon next to each variable in the variable list identifies the variable type.



Scale



Nominal



Ordinal

You can temporarily change the measurement level for a variable by right-clicking the variable in the source variable list and selecting a measurement level from the context menu.

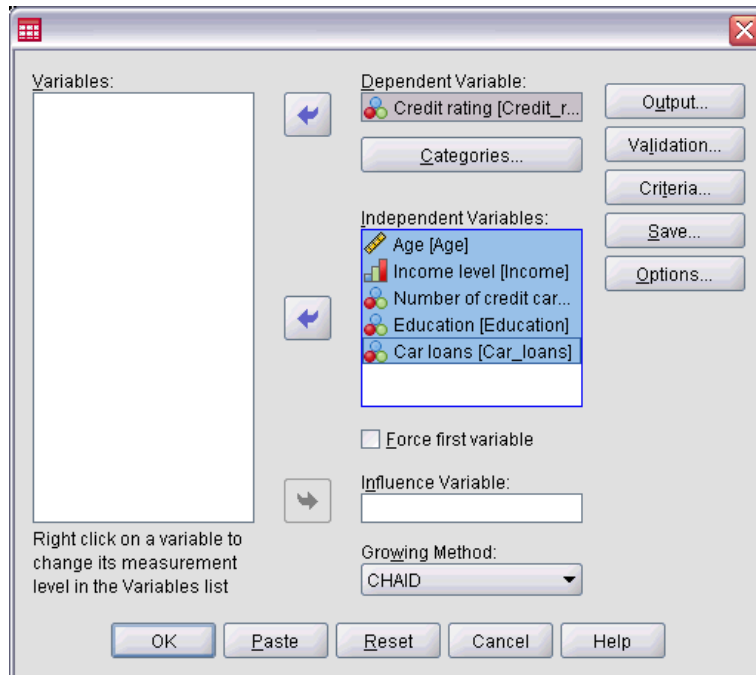
- **Value labels.** The dialog box interface for this procedure assumes that either all nonmissing values of a categorical (nominal, ordinal) dependent variable have defined value labels or none of them do. Some features are not available unless

at least two nonmissing values of the categorical dependent variable have value labels. If at least two nonmissing values have defined value labels, any cases with other values that do not have value labels will be excluded from the analysis.

To Obtain Decision Trees

- ▶ From the menus choose:
Analyze
Classify
Tree...

Figure 1-2
Decision Tree dialog box



- ▶ Select a dependent variable.
- ▶ Select one or more independent variables.
- ▶ Select a growing method.

Optionally, you can:

- Change the measurement level for any variable in the source list.
- Force the first variable in the independent variables list into the model as the first split variable.
- Select an influence variable that defines how much influence a case has on the tree-growing process. Cases with lower influence values have less influence; cases with higher values have more. Influence variable values must be positive.
- Validate the tree.
- Customize the tree-growing criteria.
- Save terminal node numbers, predicted values, and predicted probabilities as variables.
- Save the model in XML (PMML) format.

Changing Measurement Level

- ▶ Right-click the variable in the source list.
- ▶ Select a measurement level from the pop-up context menu.

This changes the measurement level temporarily for use in the Decision Tree procedure.

Growing Methods

The available growing methods are:

CHAID. Chi-squared Automatic Interaction Detection. At each step, CHAID chooses the independent (predictor) variable that has the strongest interaction with the dependent variable. Categories of each predictor are merged if they are not significantly different with respect to the dependent variable.

Exhaustive CHAID. A modification of CHAID that examines all possible splits for each predictor.

CRT. Classification and Regression Trees. CRT splits the data into segments that are as homogeneous as possible with respect to the dependent variable. A terminal node in which all cases have the same value for the dependent variable is a homogeneous, "pure" node.

QUEST. Quick, Unbiased, Efficient Statistical Tree. A method that is fast and avoids other methods' bias in favor of predictors with many categories. QUEST can be specified only if the dependent variable is nominal.

There are benefits and limitations with each method, including:

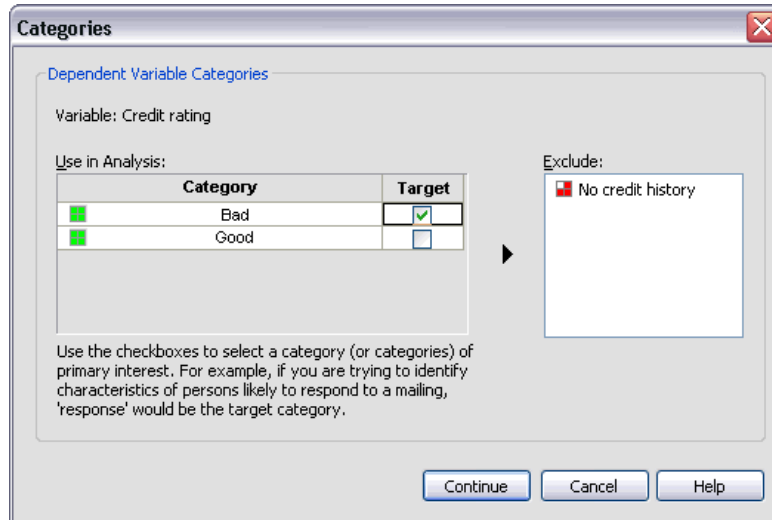
	CHAID*	CRT	QUEST
Chi-square-based**	X		
Surrogate independent (predictor) variables		X	X
Tree pruning		X	X
Multiway node splitting	X		
Binary node splitting		X	X
Influence variables	X	X	
Prior probabilities		X	X
Misclassification costs	X	X	X
Fast calculation	X		X

*Includes Exhaustive CHAID.

**QUEST also uses a chi-square measure for nominal independent variables.

Selecting Categories

Figure 1-3
Categories dialog box



For categorical (nominal, ordinal) dependent variables, you can:

- Control which categories are included in the analysis.
- Identify the target categories of interest.

Including/Excluding Categories

You can limit the analysis to specific categories of the dependent variable.

- Cases with values of the dependent variable in the Exclude list are not included in the analysis.
- For nominal dependent variables, you can also include user-missing categories in the analysis. (By default, user-missing categories are displayed in the Exclude list.)

Target Categories

Selected (checked) categories are treated as the categories of primary interest in the analysis. For example, if you are primarily interested in identifying those individuals most likely to default on a loan, you might select the “bad” credit-rating category as the target category.

- There is no default target category. If no category is selected, some classification rule options and gains-related output are not available.
- If multiple categories are selected, separate gains tables and charts are produced for each target category.
- Designating one or more categories as target categories has no effect on the tree model, risk estimate, or misclassification results.

Categories and Value Labels

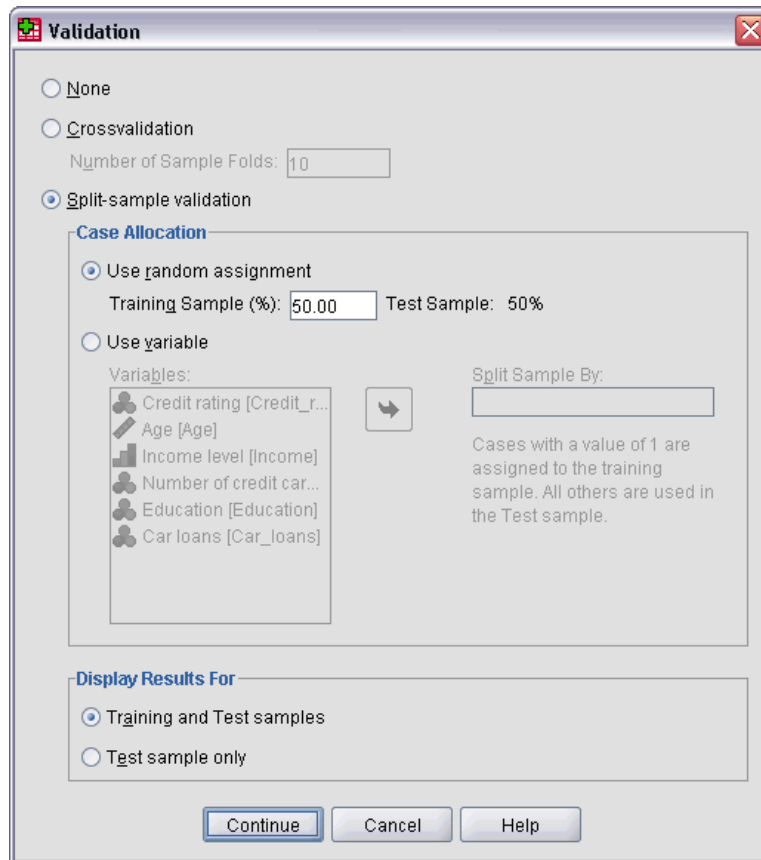
This dialog box requires defined value labels for the dependent variable. It is not available unless at least two values of the categorical dependent variable have defined value labels.

To Include/Exclude Categories and Select Target Categories

- ▶ In the main Decision Tree dialog box, select a categorical (nominal, ordinal) dependent variable with two or more defined value labels.
- ▶ Click Categories.

Validation

Figure 1-4
Validation dialog box



Validation allows you to assess how well your tree structure generalizes to a larger population. Two validation methods are available: crossvalidation and split-sample validation.

Crossvalidation

Crossvalidation divides the sample into a number of subsamples, or **folds**. Tree models are then generated, excluding the data from each subsample in turn. The first tree is based on all of the cases except those in the first sample fold, the second tree is based on all of the cases except those in the second sample fold, and so on. For each tree, misclassification risk is estimated by applying the tree to the subsample excluded in generating it.

- You can specify a maximum of 25 sample folds. The higher the value, the fewer the number of cases excluded for each tree model.
- Crossvalidation produces a single, final tree model. The crossvalidated risk estimate for the final tree is calculated as the average of the risks for all of the trees.

Split-Sample Validation

With split-sample validation, the model is generated using a training sample and tested on a hold-out sample.

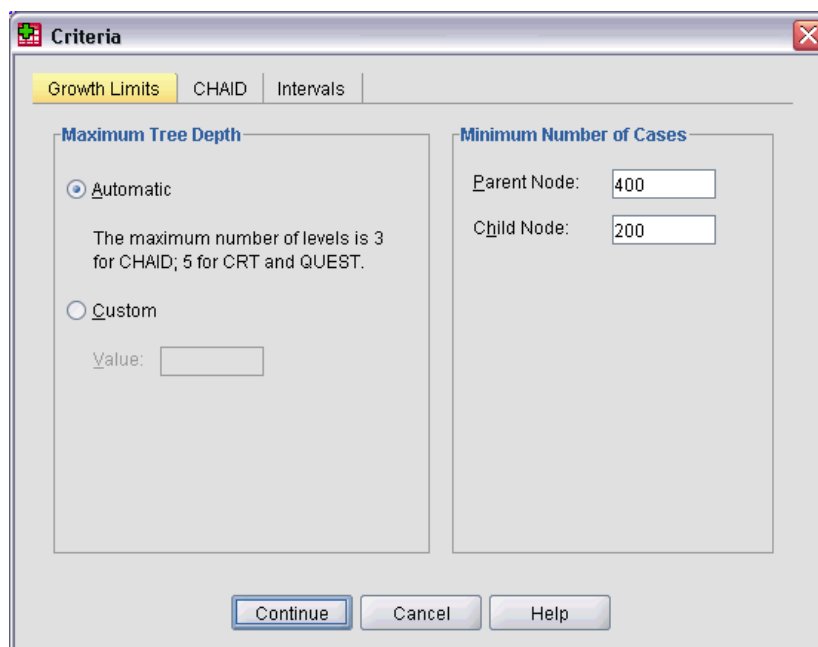
- You can specify a training sample size, expressed as a percentage of the total sample size, or a variable that splits the sample into training and testing samples.
- If you use a variable to define training and testing samples, cases with a value of 1 for the variable are assigned to the training sample, and all other cases are assigned to the testing sample. The variable cannot be the dependent variable, weight variable, influence variable, or a forced independent variable.
- You can display results for both the training and testing samples or just the testing sample.
- Split-sample validation should be used with caution on small data files (data files with a small number of cases). Small training sample sizes may yield poor models, since there may not be enough cases in some categories to adequately grow the tree.

Tree-Growing Criteria

The available growing criteria may depend on the growing method, level of measurement of the dependent variable, or a combination of the two.

Growth Limits

Figure 1-5
Criteria dialog box, Growth Limits tab



The Growth Limits tab allows you to limit the number of levels in the tree and control the minimum number of cases for parent and child nodes.

Maximum Tree Depth. Controls the maximum number of levels of growth beneath the root node. The Automatic setting limits the tree to three levels beneath the root node for the CHAID and Exhaustive CHAID methods and five levels for the CRT and QUEST methods.

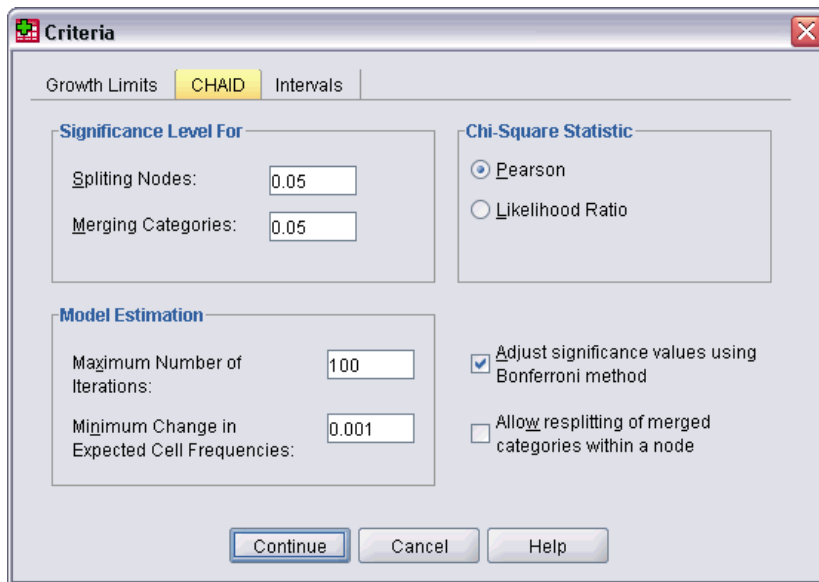
Minimum Number of Cases. Controls the minimum numbers of cases for nodes. Nodes that do not satisfy these criteria will not be split.

- Increasing the minimum values tends to produce trees with fewer nodes.
- Decreasing the minimum values produces trees with more nodes.

For data files with a small number of cases, the default values of 100 cases for parent nodes and 50 cases for child nodes may sometimes result in trees with no nodes below the root node; in this case, lowering the minimum values may produce more useful results.

CHAID Criteria

Figure 1-6
Criteria dialog box, CHAID tab



For the CHAID and Exhaustive CHAID methods, you can control:

Significance Level. You can control the significance value for splitting nodes and merging categories. For both criteria, the default significance level is 0.05.

- For splitting nodes, the value must be greater than 0 and less than 1. Lower values tend to produce trees with fewer nodes.
- For merging categories, the value must be greater than 0 and less than or equal to 1. To prevent merging of categories, specify a value of 1. For a scale independent variable, this means that the number of categories for the variable in the final tree is the specified number of intervals (the default is 10). For more information, see [Scale Intervals for CHAID Analysis](#) on p. 14.

Chi-Square Statistic. For ordinal dependent variables, chi-square for determining node splitting and category merging is calculated using the likelihood-ratio method. For nominal dependent variables, you can select the method:

- **Pearson.** This method provides faster calculations but should be used with caution on small samples. This is the default method.
- **Likelihood ratio.** This method is more robust than Pearson but takes longer to calculate. For small samples, this is the preferred method.

Model Estimation. For nominal and ordinal dependent variables, you can specify:

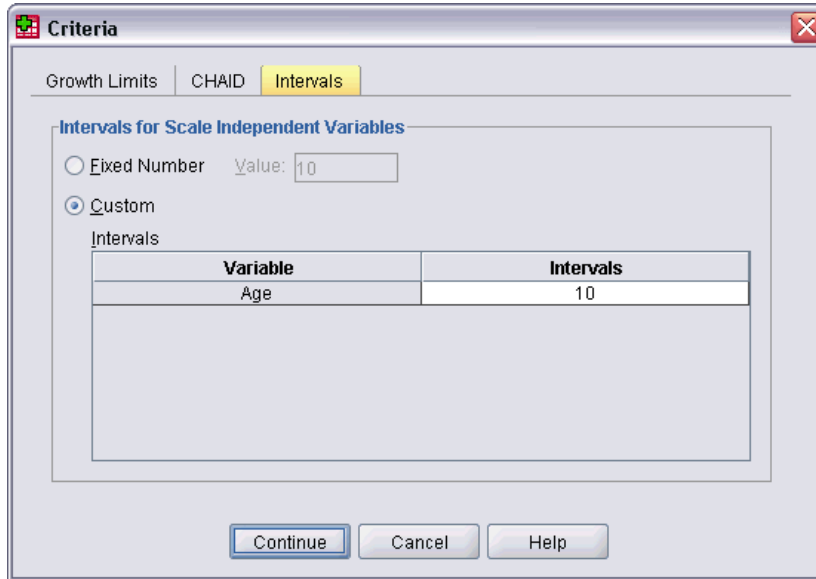
- **Maximum number of iterations.** The default is 100. If the tree stops growing because the maximum number of iterations has been reached, you may want to increase the maximum or change one or more of the other criteria that control tree growth.
- **Minimum change in expected cell frequencies.** The value must be greater than 0 and less than 1. The default is 0.05. Lower values tend to produce trees with fewer nodes.

Adjust significance values using Bonferroni method. For multiple comparisons, significance values for merging and splitting criteria are adjusted using the Bonferroni method. This is the default.

Allow resplitting of merged categories within a node. Unless you explicitly prevent category merging, the procedure will attempt to merge independent (predictor) variable categories together to produce the simplest tree that describes the model. This option allows the procedure to resplit merged categories if that provides a better solution.

Scale Intervals for CHAID Analysis

Figure 1-7
Criteria dialog box, Intervals tab



In CHAID analysis, scale independent (predictor) variables are always banded into discrete groups (for example, 0–10, 11–20, 21–30, etc.) prior to analysis. You can control the initial/maximum number of groups (although the procedure may merge contiguous groups after the initial split):

- **Fixed number.** All scale independent variables are initially banded into the same number of groups. The default is 10.
- **Custom.** Each scale independent variable is initially banded into the number of groups specified for that variable.

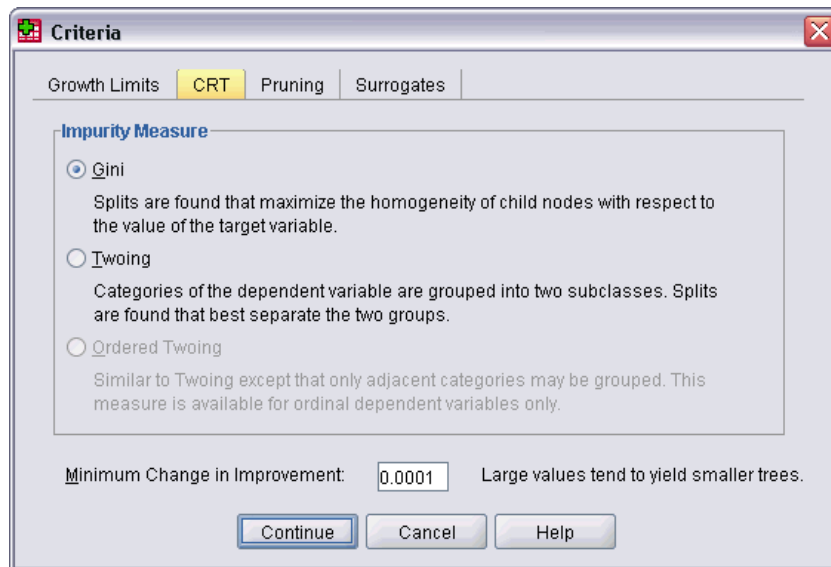
To Specify Intervals for Scale Independent Variables

- ▶ In the main Decision Tree dialog box, select one or more scale independent variables.
- ▶ For the growing method, select CHAID or Exhaustive CHAID.
- ▶ Click Criteria.
- ▶ Click the Intervals tab.

In CRT and QUEST analysis, all splits are binary and scale and ordinal independent variables are handled the same way; so, you cannot specify a number of intervals for scale independent variables.

CRT Criteria

Figure 1-8
Criteria dialog box, CRT tab



The CRT growing method attempts to maximize within-node homogeneity. The extent to which a node does not represent a homogenous subset of cases is an indication of **impurity**. For example, a terminal node in which all cases have the same value for the dependent variable is a homogenous node that requires no further splitting because it is “pure.”

You can select the method used to measure impurity and the minimum decrease in impurity required to split nodes.

Impurity Measure. For scale dependent variables, the least-squared deviation (LSD) measure of impurity is used. It is computed as the within-node variance, adjusted for any frequency weights or influence values.

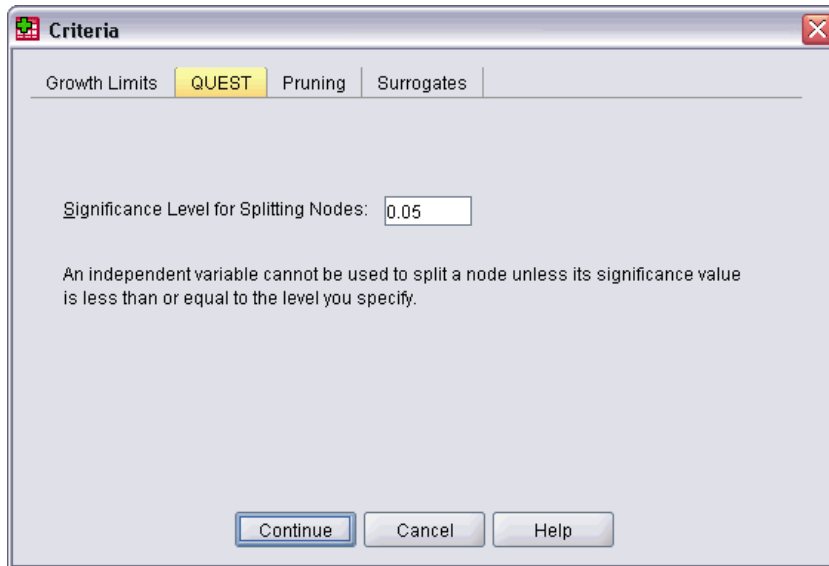
For categorical (nominal, ordinal) dependent variables, you can select the impurity measure:

- **Gini.** Splits are found that maximize the homogeneity of child nodes with respect to the value of the dependent variable. Gini is based on squared probabilities of membership for each category of the dependent variable. It reaches its minimum (zero) when all cases in a node fall into a single category. This is the default measure.
- **Twoing.** Categories of the dependent variable are grouped into two subclasses. Splits are found that best separate the two groups.
- **Ordered twoing.** Similar to twoing except that only adjacent categories can be grouped. This measure is available only for ordinal dependent variables.

Minimum change in improvement. This is the minimum decrease in impurity required to split a node. The default is 0.0001. Higher values tend to produce trees with fewer nodes.

QUEST Criteria

Figure 1-9
Criteria dialog box, QUEST tab



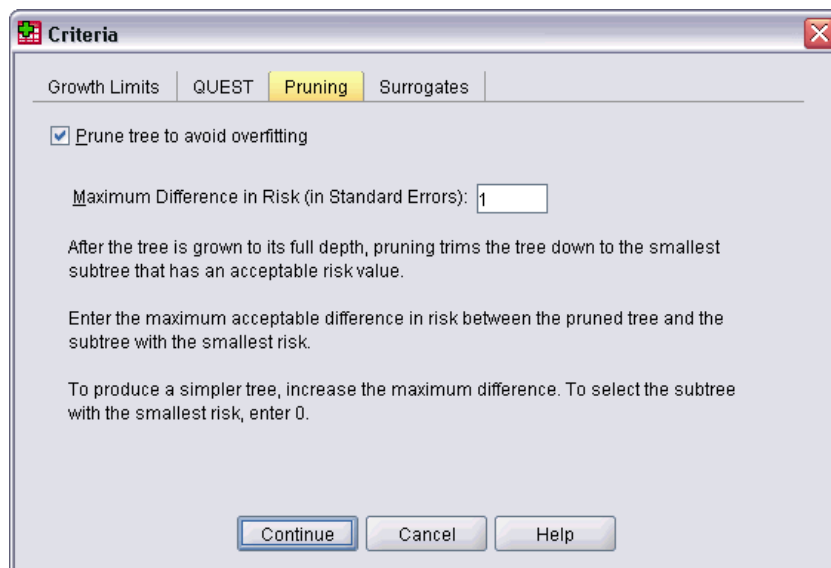
For the QUEST method, you can specify the significance level for splitting nodes. An independent variable cannot be used to split nodes unless the significance level is less than or equal to the specified value. The value must be greater than 0 and less than 1. The default is 0.05. Smaller values will tend to exclude more independent variables from the final model.

To Specify QUEST Criteria

- ▶ In the main Decision Tree dialog box, select a nominal dependent variable.
- ▶ For the growing method, select QUEST.
- ▶ Click Criteria.
- ▶ Click the QUEST tab.

Pruning Trees

Figure 1-10
Criteria dialog box, Pruning tab



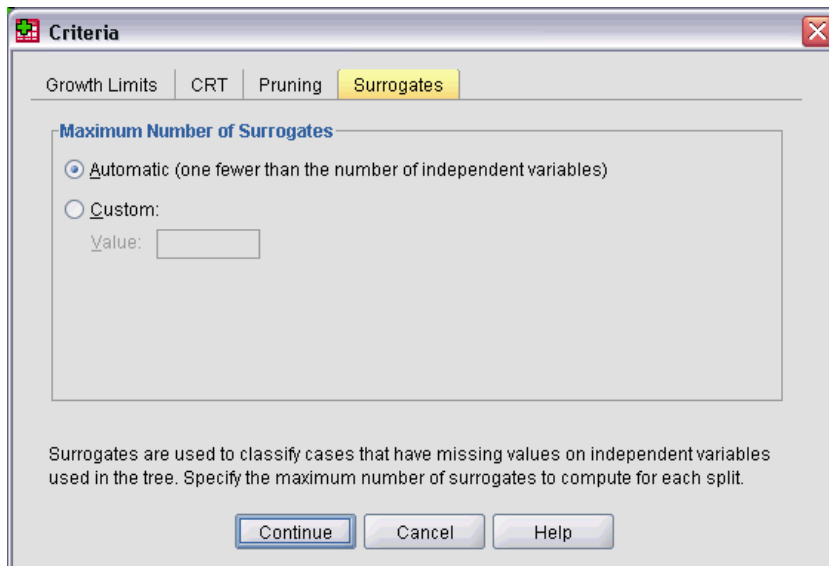
With the CRT and QUEST methods, you can avoid overfitting the model by **pruning** the tree: the tree is grown until stopping criteria are met, and then it is trimmed automatically to the smallest subtree based on the specified maximum difference in risk. The risk value is expressed in standard errors. The default is 1. The value must be non-negative. To obtain the subtree with the minimum risk, specify 0.

Pruning versus Hiding Nodes

When you create a pruned tree, any nodes pruned from the tree are not available in the final tree. You can interactively hide and show selected child nodes in the final tree, but you cannot show nodes that were pruned in the tree creation process. For more information, see [Tree Editor](#) in Chapter 2 on p. 48.

Surrogates

Figure 1-11
Criteria dialog box, Surrogates tab



CRT and QUEST can use **surrogates** for independent (predictor) variables. For cases in which the value for that variable is missing, other independent variables having high associations with the original variable are used for classification. These alternative

predictors are called surrogates. You can specify the maximum number of surrogates to use in the model.

- By default, the maximum number of surrogates is one less than the number of independent variables. In other words, for each independent variable, all other independent variables may be used as surrogates.
- If you don't want the model to use surrogates, specify 0 for the number of surrogates.

Options

Available options may depend on the growing method, the level of measurement of the dependent variable, and/or the existence of defined value labels for values of the dependent variable.

Misclassification Costs

Figure 1-12
Options dialog box, Misclassification Costs tab

The screenshot shows the 'Options' dialog box with the 'Misclassification Costs' tab selected. The 'Custom' radio button is chosen. The table below shows the misclassification costs for 'Bad' and 'Good' predicted categories against 'Bad' and 'Good' actual categories.

		Predicted Category	
		Bad	Good
Actual Category	Bad	0	2
	Good	1	0

Below the table is a 'Fill Matrix' section with three buttons: 'Duplicate Lower Triangle', 'Duplicate Upper Triangle', and 'Use Average Cell Values'. At the bottom of the dialog are 'Continue', 'Cancel', and 'Help' buttons.

For categorical (nominal, ordinal) dependent variables, misclassification costs allow you to include information about the relative penalty associated with incorrect classification. For example:

- The cost of denying credit to a creditworthy customer is likely to be different from the cost of extending credit to a customer who then defaults on the loan.
- The cost of misclassifying an individual with a high risk of heart disease as low risk is probably much higher than the cost of misclassifying a low-risk individual as high-risk.
- The cost of sending a mass mailing to someone who isn't likely to respond is probably fairly low, while the cost of not sending the mailing to someone who is likely to respond is relatively higher (in terms of lost revenue).

Misclassification Costs and Value Labels

This dialog box is not available unless at least two values of the categorical dependent variable have defined value labels.

To Specify Misclassification Costs

- ▶ In the main Decision Tree dialog box, select a categorical (nominal, ordinal) dependent variable with two or more defined value labels.
- ▶ Click Options.
- ▶ Click the Misclassification Costs tab.
- ▶ Click Custom.
- ▶ Enter one or more misclassification costs in the grid. Values must be non-negative. (Correct classifications, represented on the diagonal, are always 0.)

Fill Matrix. In many instances, you may want costs to be symmetric—that is, the cost of misclassifying A as B is the same as the cost of misclassifying B as A. The following controls can make it easier to specify a symmetric cost matrix:

- **Duplicate Lower Triangle.** Copies values in the lower triangle of the matrix (below the diagonal) into the corresponding upper-triangular cells.

- **Duplicate Upper Triangle.** Copies values in the upper triangle of the matrix (above the diagonal) into the corresponding lower-triangular cells.
- **Use Average Cell Values.** For each cell in each half of the matrix, the two values (upper- and lower-triangular) are averaged and the average replaces both values. For example, if the cost of misclassifying A as B is 1 and the cost of misclassifying B as A is 3, then this control replaces both of those values with the average $(1+3)/2 = 2$.

Profits

Figure 1-13
Options dialog box, Profits tab

Options dialog box, Profits tab

Missing Values | Missclassification Costs | Profits

None
 Custom

Revenue and Expense Values:

	Revenue	Expense	Profit
Bad	10	12	-2.0
Good	100	5	95.0

Enter revenue and expense values for each category. Profits are computed automatically.

Continue Cancel Help

For categorical dependent variables, you can assign revenue and expense values to levels of the dependent variable.

- Profit is computed as revenue minus expense.

- Profit values affect average profit and ROI (return on investment) values in gains tables. They do not affect the basic tree model structure.
- Revenue and expense values must be numeric and must be specified for all categories of the dependent variable displayed in the grid.

Profits and Value Labels

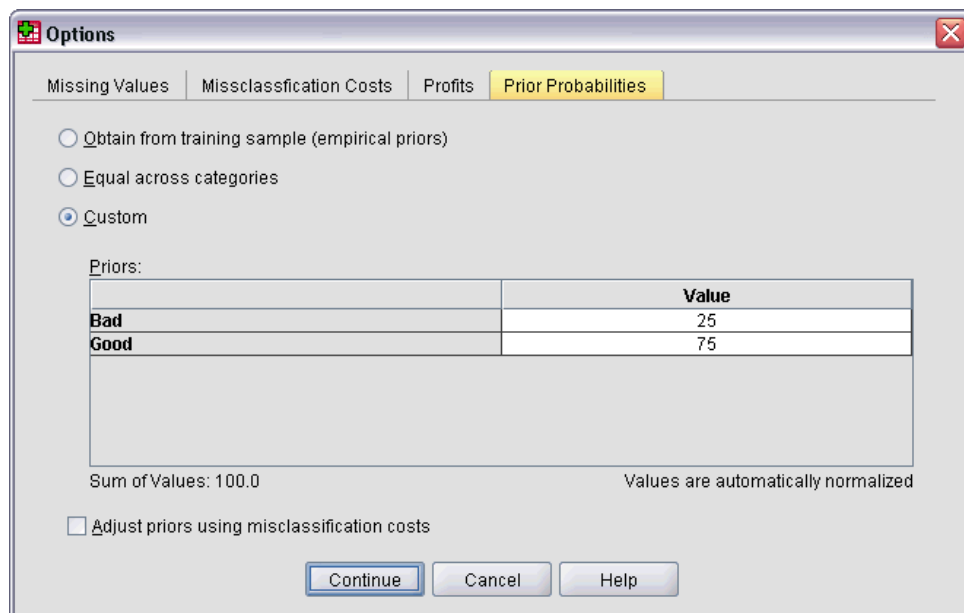
This dialog box requires defined value labels for the dependent variable. It is not available unless at least two values of the categorical dependent variable have defined value labels.

To Specify Profits

- ▶ In the main Decision Tree dialog box, select a categorical (nominal, ordinal) dependent variable with two or more defined value labels.
- ▶ Click Options.
- ▶ Click the Profits tab.
- ▶ Click Custom.
- ▶ Enter revenue and expense values for all dependent variable categories listed in the grid.

Prior Probabilities

Figure 1-14
Options dialog box, Prior Probabilities tab



For CRT and QUEST trees with categorical dependent variables, you can specify prior probabilities of group membership. **Prior probabilities** are estimates of the overall relative frequency for each category of the dependent variable prior to knowing anything about the values of the independent (predictor) variables. Using prior probabilities helps to correct any tree growth caused by data in the sample that is not representative of the entire population.

Obtain from training sample (empirical priors). Use this setting if the distribution of dependent variable values in the data file is representative of the population distribution. If you are using split-sample validation, the distribution of cases in the training sample is used.

Note: Since cases are randomly assigned to the training sample in split-sample validation, you won't know the actual distribution of cases in the training sample in advance. For more information, see [Validation](#) on p. 9.

Equal across categories. Use this setting if categories of the dependent variable are represented equally in the population. For example, if there are four categories, approximately 25% of the cases are in each category.

Custom. Enter a non-negative value for each category of the dependent variable listed in the grid. The values can be proportions, percentages, frequency counts, or any other values that represent the distribution of values across categories.

Adjust priors using misclassification costs. If you define custom misclassification costs, you can adjust prior probabilities based on those costs. For more information, see [Misclassification Costs](#) on p. 19.

Profits and Value Labels

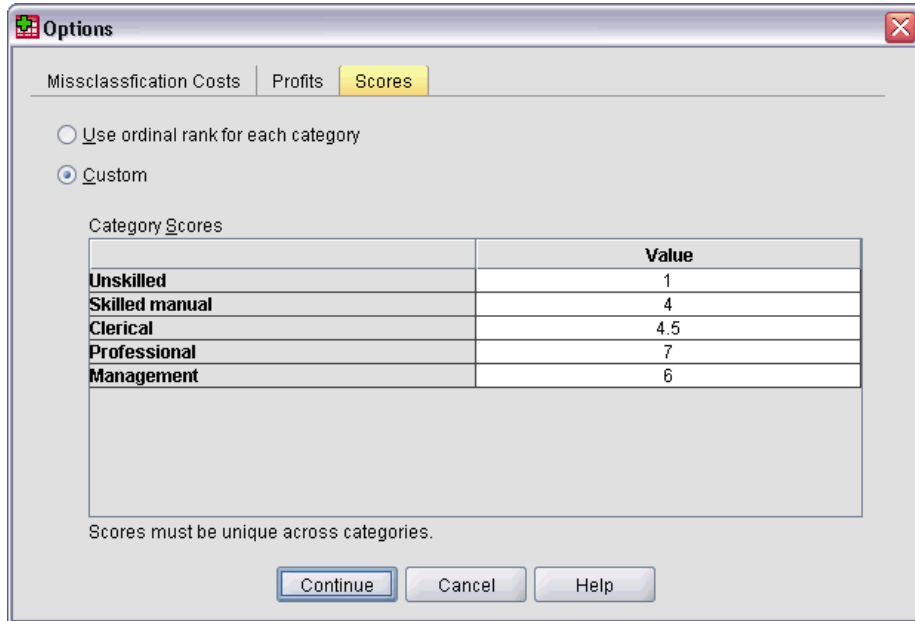
This dialog box requires defined value labels for the dependent variable. It is not available unless at least two values of the categorical dependent variable have defined value labels.

To Specify Prior Probabilities

- ▶ In the main Decision Tree dialog box, select a categorical (nominal, ordinal) dependent variable with two or more defined value labels.
- ▶ For the growing method, select CRT or QUEST.
- ▶ Click Options.
- ▶ Click the Prior Probabilities tab.

Scores

Figure 1-15
Options dialog box, Scores tab



For CHAID and Exhaustive CHAID with an ordinal dependent variable, you can assign custom scores to each category of the dependent variable. Scores define the order of and distance between categories of the dependent variable. You can use scores to increase or decrease the relative distance between ordinal values or to change the order of the values.

- **Use ordinal rank for each category.** The lowest category of the dependent variable is assigned a score of 1, the next highest category is assigned a score of 2, and so on. This is the default.
- **Custom.** Enter a numeric score value for each category of the dependent variable listed in the grid.

Example

Value Label	Original Value	Score
Unskilled	1	1

Value Label	Original Value	Score
Skilled manual	2	4
Clerical	3	4.5
Professional	4	7
Management	5	6

- The scores increase the relative distance between *Unskilled* and *Skilled manual* and decrease the relative distance between *Skilled manual* and *Clerical*.
- The scores reverse the order of *Management* and *Professional*.

Scores and Value Labels

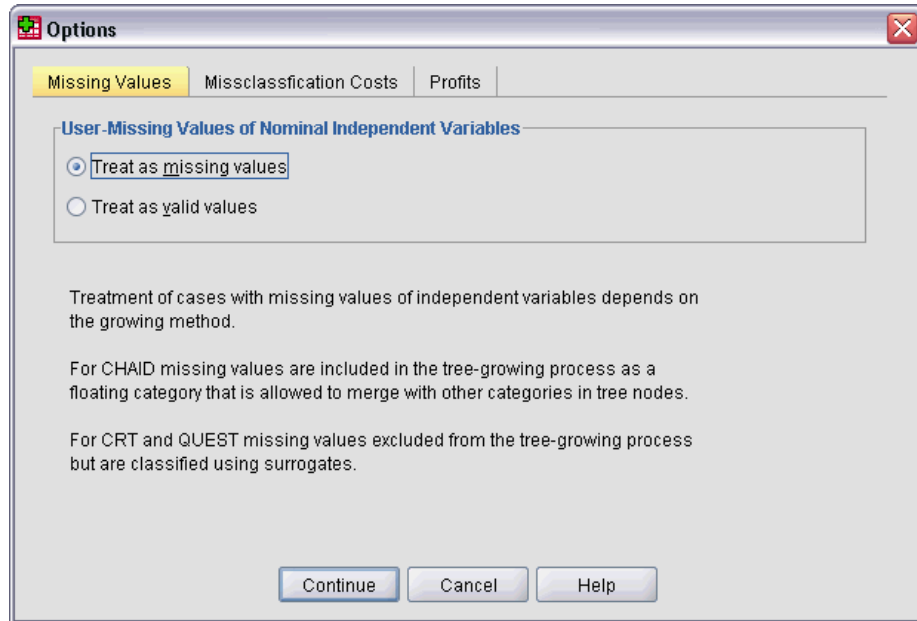
This dialog box requires defined value labels for the dependent variable. It is not available unless at least two values of the categorical dependent variable have defined value labels.

To Specify Scores

- ▶ In the main Decision Tree dialog box, select an ordinal dependent variable with two or more defined value labels.
- ▶ For the growing method, select CHAID or Exhaustive CHAID.
- ▶ Click Options.
- ▶ Click the Scores tab.

Missing Values

Figure 1-16
Options dialog box, Missing Values tab



The Missing Values tab controls the handling of nominal, user-missing, independent (predictor) variable values.

- Handling of ordinal and scale user-missing independent variable values varies between growing methods.
- Handling of nominal dependent variables is specified in the Categories dialog box. For more information, see [Selecting Categories](#) on p. 7.
- For ordinal and scale dependent variables, cases with system-missing or user-missing dependent variable values are always excluded.

Treat as missing values. User-missing values are treated like system-missing values. The handling of system-missing values varies between growing methods.

Treat as valid values. User-missing values of nominal independent variables are treated as ordinary values in tree growing and classification.

Method-Dependent Rules

If some, but not all, independent variable values are system- or user-missing:

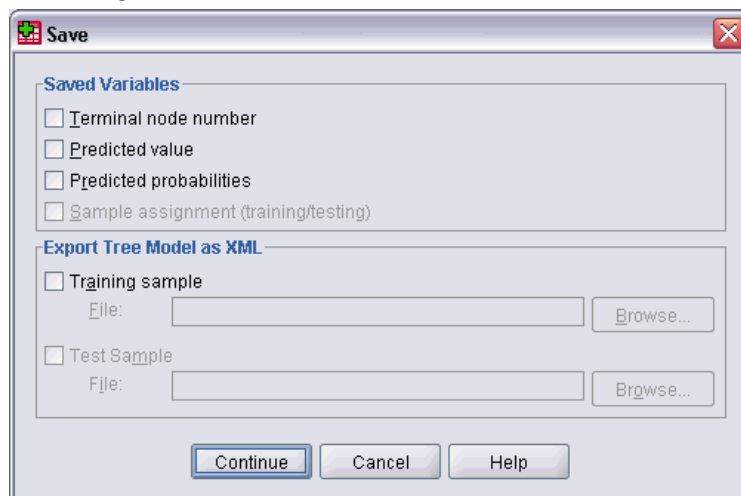
- For CHAID and Exhaustive CHAID, system- and user-missing independent variable values are included in the analysis as a single, combined category. For scale and ordinal independent variables, the algorithms first generate categories using valid values and then decide whether to merge the missing category with its most similar (valid) category or keep it as a separate category.
- For CRT and QUEST, cases with missing independent variable values are excluded from the tree-growing process but are classified using surrogates if surrogates are included in the method. If nominal user-missing values are treated as missing, they are also handled in this manner. For more information, see [Surrogates](#) on p. 18.

To Specify Nominal, Independent User-Missing Treatment

- ▶ In the main Decision Tree dialog box, select at least one nominal independent variable.
- ▶ Click Options.
- ▶ Click the Missing Values tab.

Saving Model Information

Figure 1-17
Save dialog box



You can save information from the model as variables in the working data file, and you can also save the entire model in XML (PMML) format to an external file.

Saved Variables

Terminal node number. The terminal node to which each case is assigned. The value is the tree node number.

Predicted value. The class (group) or value for the dependent variable predicted by the model.

Predicted probabilities. The probability associated with the model's prediction. One variable is saved for each category of the dependent variable. Not available for scale dependent variables.

Sample assignment (training/testing). For split-sample validation, this variable indicates whether a case was used in the training or testing sample. The value is 1 for the training sample and 0 for the testing sample. Not available unless you have selected split-sample validation. For more information, see [Validation](#) on p. 9.

Export Tree Model as XML

You can save the entire tree model in XML (PMML) format. *SmartScore* and SPSS Statistics Server (a separate product) can use this model file to apply the model information to other data files for scoring purposes.

Training sample. Writes the model to the specified file. For split-sample validated trees, this is the model for the training sample.

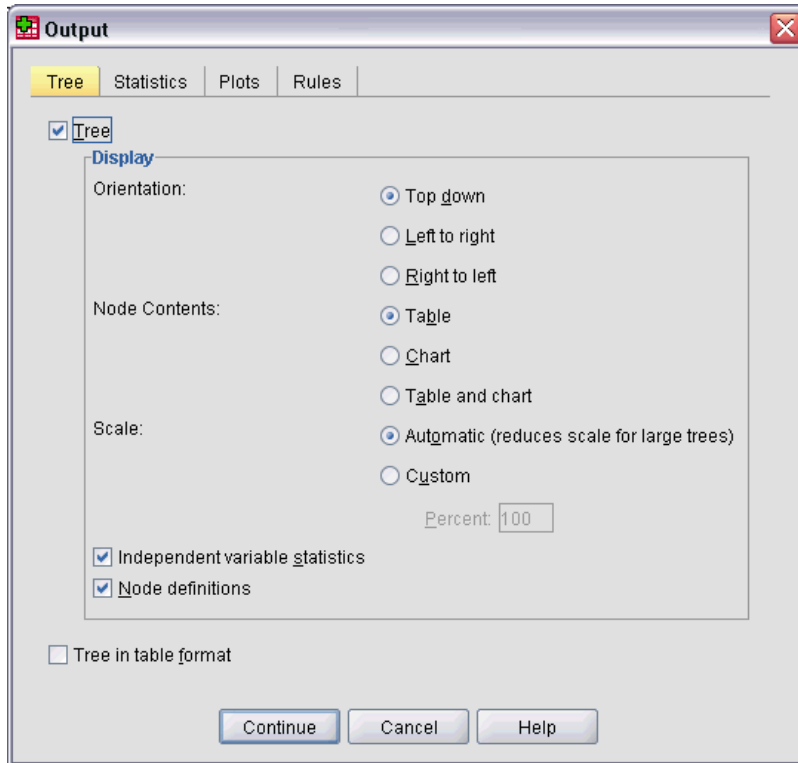
Test sample. Writes the model for the test sample to the specified file. Not available unless you have selected split-sample validation.

Output

Available output options depend on the growing method, the measurement level of the dependent variable, and other settings.

Tree Display

Figure 1-18
Output dialog box, Tree tab



You can control the initial appearance of the tree or completely suppress the tree display.

Tree. By default, the tree diagram is included in the output displayed in the Viewer. Deselect (uncheck) this option to exclude the tree diagram from the output.

Display. These options control the initial appearance of the tree diagram in the Viewer. All of these attributes can also be modified by editing the generated tree.

- **Orientation.** The tree can be displayed top down with the root node at the top, left to right, or right to left.

- **Node contents.** Nodes can display tables, charts, or both. For categorical dependent variables, tables display frequency counts and percentages, and the charts are bar charts. For scale dependent variables, tables display means, standard deviations, number of cases, and predicted values, and the charts are histograms.
- **Scale.** By default, large trees are automatically scaled down in an attempt to fit the tree on the page. You can specify a custom scale percentage of up to 200%.
- **Independent variable statistics.** For CHAID and Exhaustive CHAID, statistics include F value (for scale dependent variables) or chi-square value (for categorical dependent variables) as well as significance value and degrees of freedom. For CRT, the improvement value is shown. For QUEST, F , significance value, and degrees of freedom are shown for scale and ordinal independent variables; for nominal independent variables, chi-square, significance value, and degrees of freedom are shown.
- **Node definitions.** Node definitions display the value(s) of the independent variable used at each node split.

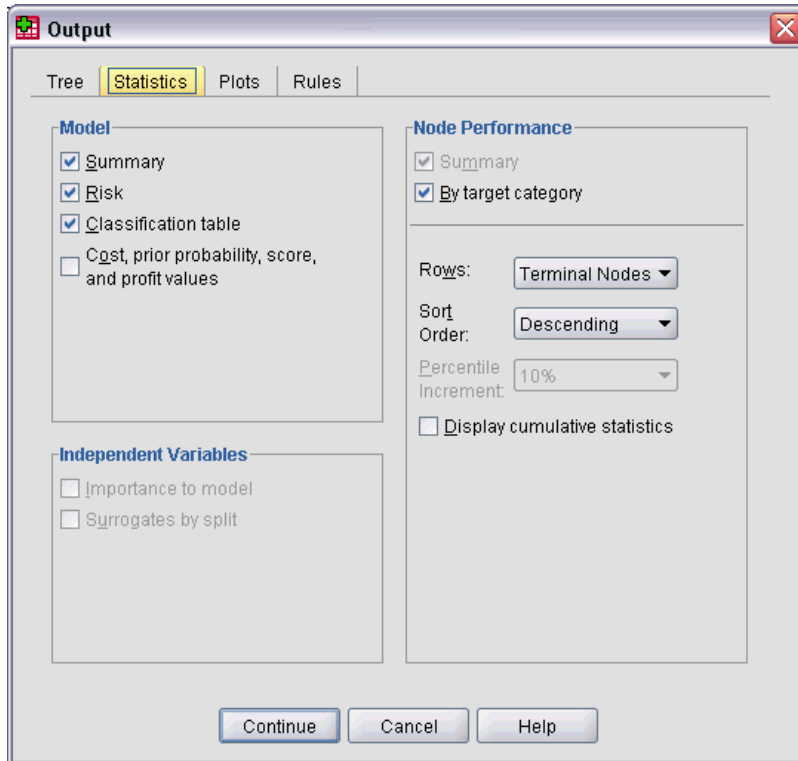
Tree in table format. Summary information for each node in the tree, including parent node number, independent variable statistics, independent variable value(s) for the node, mean and standard deviation for scale dependent variables, or counts and percentages for categorical dependent variables.

Figure 1-19
Tree in table format

Node	Bad		Good		Total		Predicted Category	Parent Node	Primary Independent Variable				
	N	Percent	N	Percent	N	Percent			Variable	Sig.	Chi-Square	df	Split Values
0	1020	41.4%	1444	58.6%	2464	100.0%	Good						
1	454	82.1%	99	17.9%	553	22.4%	Bad	0	Income level	.000	662.457	2	<= Low
2	476	42.0%	658	58.0%	1134	46.0%	Good	0	Income level	.000	662.457	2	(Low, Medium]
3	90	11.6%	687	88.4%	777	31.5%	Good	0	Income level	.000	662.457	2	> Medium
4	422	56.7%	322	43.3%	744	30.2%	Bad	2	Number of credit cards	.000	193.113	1	5 or more
5	54	13.8%	336	86.2%	390	15.8%	Good	2	Number of credit cards	.000	193.113	1	Less than 5
6	80	17.6%	375	82.4%	455	18.5%	Good	3	Number of credit cards	.000	38.587	1	5 or more
7	10	3.1%	312	96.9%	322	13.1%	Good	3	Number of credit cards	.000	38.587	1	Less than 5
8	211	80.8%	50	19.2%	261	10.6%	Bad	4	Age	.000	95.299	1	<= 28.079
9	211	43.7%	272	56.3%	483	19.6%	Good	4	Age	.000	95.299	1	> 28.079

Statistics

Figure 1-20
Output dialog box, Statistics tab



Available statistics tables depend on the measurement level of the dependent variable, the growing method, and other settings.

Model

Summary. The summary includes the method used, the variables included in the model, and the variables specified but not included in the model.

Figure 1-21
Model summary table

Specifications	Growing Method	CHAID
	Dependent Variable	Credit rating
	Independent Variables	Age, Income, Credit cards, Education, Car loans
	Validation	NONE
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	400
	Minimum Cases in Child Node	200
Results	Independent Variables Included	Age, Income, Credit cards
	Number of Nodes	10
	Number of Terminal Nodes	6
	Depth	3

Risk. Risk estimate and its standard error. A measure of the tree's predictive accuracy.

- For categorical dependent variables, the risk estimate is the proportion of cases incorrectly classified after adjustment for prior probabilities and misclassification costs.
- For scale dependent variables, the risk estimate is within-node variance.

Classification table. For categorical (nominal, ordinal) dependent variables, this table shows the number of cases classified correctly and incorrectly for each category of the dependent variable. Not available for scale dependent variables.

Figure 1-22
Risk and classification tables

Risk

Estimate	Std. Error
.205	.008

Growing Method: CHAID
Dependent Variable: Credit rating

Classification

Observed	Predicted		
	Bad	Good	Percent Correct
Bad	665	355	65.2%
Good	149	1295	89.7%
Overall Percentage	33.0%	67.0%	79.5%

Growing Method: CHAID
Dependent Variable: Credit rating

Cost, prior probability, score, and profit values. For categorical dependent variables, this table shows the cost, prior probability, score, and profit values used in the analysis. Not available for scale dependent variables.

Independent Variables

Importance to model. For the CRT growing method, ranks each independent (predictor) variable according to its importance to the model. Not available for QUEST or CHAID methods.

Surrogates by split. For the CRT and QUEST growing methods, if the model includes surrogates, lists surrogates for each split in the tree. Not available for CHAID methods. For more information, see [Surrogates](#) on p. 18.

Node Performance

Summary. For scale dependent variables, the table includes the node number, the number of cases, and the mean value of the dependent variable. For categorical dependent variables with defined profits, the table includes the node number, the number of cases, the average profit, and the ROI (return on investment) values. Not available for categorical dependent variables without defined profits. For more information, see [Profits](#) on p. 21.

Figure 1-23
Gain summary tables for nodes and percentiles

Gain Summary for Nodes

Node	N	Percent	Profit	ROI
7	322	13.1%	77.826	377.4%
5	390	15.8%	70.308	308.8%
6	455	18.5%	67.692	287.9%
9	483	19.6%	49.420	172.0%
8	261	10.6%	23.410	64.7%
1	553	22.4%	22.532	61.9%

Gain Summary for Percentiles

Percentile	Nodes	N	Profit	ROI
10	7	246	77.826	377.4%
20	7 ; 5	493	75.218	352.0%
30	5 ; 6	739	73.488	336.2%
40	6	986	72.036	323.4%
50	6 ; 9	1232	70.205	307.9%
60	9	1478	66.745	280.6%
70	9 ; 8	1725	63.134	254.4%
80	8 ; 1	1971	58.149	221.6%
90	1	2218	54.183	197.9%
100	1	2464	51.023	180.4%

By target category. For categorical dependent variables with defined target categories, the table includes the percentage gain, the response percentage, and the index percentage (lift) by node or percentile group. A separate table is produced for each target category. Not available for scale dependent variables or categorical dependent variables without defined target categories. For more information, see [Selecting Categories](#) on p. 7.

Figure 1-24
Target category gains for nodes and percentiles

Target Category: Bad

Gains for Nodes

Node	Node		Gain		Response	Index
	N	Percent	N	Percent		
1	553	22.4%	454	44.5%	82.1%	198.3%
8	261	10.6%	211	20.7%	80.8%	195.3%
9	483	19.6%	211	20.7%	43.7%	105.5%
6	455	18.5%	80	7.8%	17.6%	42.5%
5	390	15.8%	54	5.3%	13.8%	33.4%
7	322	13.1%	10	1.0%	3.1%	7.5%

Gains for Percentiles

Percentile	Nodes	N	Gain		Response	Index
			N	Percent		
10	1	246	202	19.8%	82.1%	198.3%
20	1	493	405	39.7%	82.1%	198.3%
30	1 ; 8	739	604	59.3%	81.8%	197.6%
40	8 ; 9	986	740	72.6%	75.1%	181.3%
50	9	1232	848	83.1%	68.8%	166.2%
60	9 ; 6	1478	908	89.0%	61.4%	148.4%
70	6	1725	951	93.3%	55.1%	133.2%
80	6 ; 5	1971	986	96.7%	50.0%	120.9%
90	5 ; 7	2218	1012	99.3%	45.6%	110.3%
100	7	2464	1020	100.0%	41.4%	100.0%

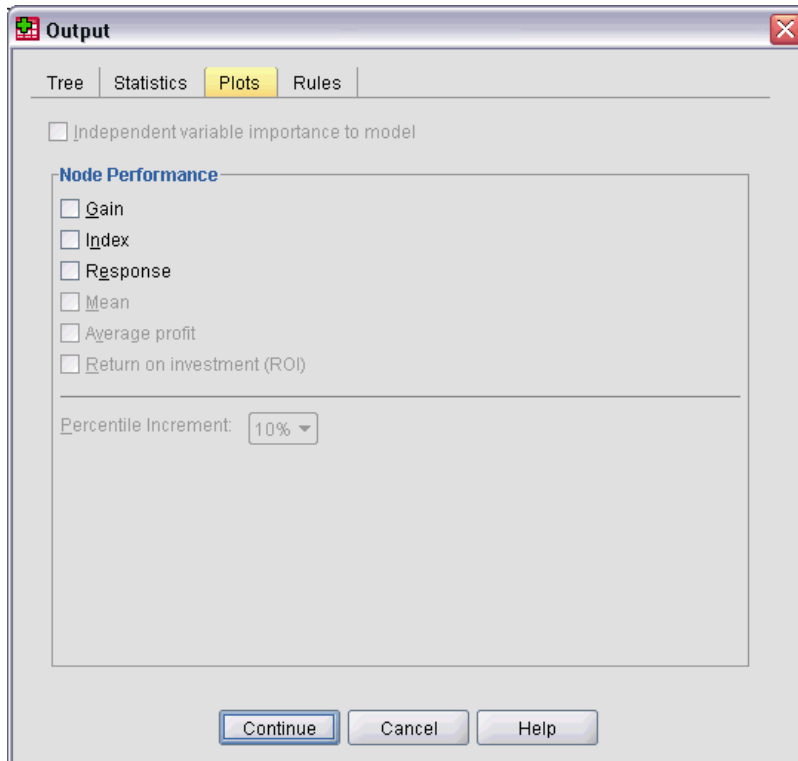
Rows. The node performance tables can display results by terminal nodes, percentiles, or both. If you select both, two tables are produced for each target category. Percentile tables display cumulative values for each percentile, based on sort order.

Percentile increment. For percentile tables, you can select the percentile increment: 1, 2, 5, 10, 20, or 25.

Display cumulative statistics. For terminal node tables, displays additional columns in each table with cumulative results.

Charts

Figure 1-25
Output dialog box, Plots tab



Available charts depend on the measurement level of the dependent variable, the growing method, and other settings.

Independent variable importance to model. Bar chart of model importance by independent variable (predictor). Available only with the CRT growing method.

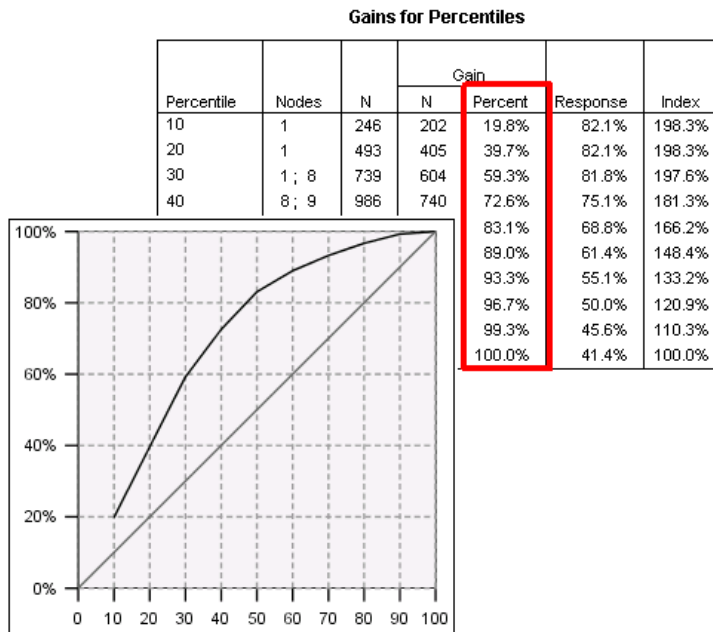
Node Performance

Gain. Gain is the percentage of total cases in the target category in each node, computed as: $(\text{node target } n / \text{total target } n) \times 100$. The gains chart is a line chart of cumulative percentile gains, computed as: $(\text{cumulative percentile target } n / \text{total target } n) \times 100$. A separate line chart is produced for each target category. Available only for

categorical dependent variables with defined target categories. For more information, see [Selecting Categories](#) on p. 7.

The gains chart plots the same values that you would see in the *Gain Percent* column in the gains for percentiles table, which also reports cumulative values.

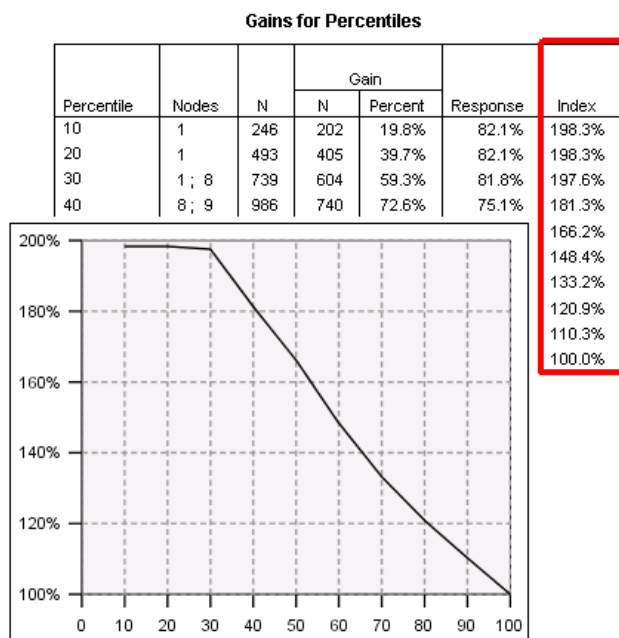
Figure 1-26
Gains for percentiles table and gains chart



Index. Index is the ratio of the node response percentage for the target category compared to the overall target category response percentage for the entire sample. The index chart is a line chart of cumulative percentile index values. Available only for categorical dependent variables. Cumulative percentile index is computed as: $(\text{cumulative percentile response percent} / \text{total response percent}) \times 100$. A separate chart is produced for each target category, and target categories must be defined.

The index chart plots the same values that you would see in the *Index* column in the gains for percentiles table.

Figure 1-27
Gains for percentiles table and index chart

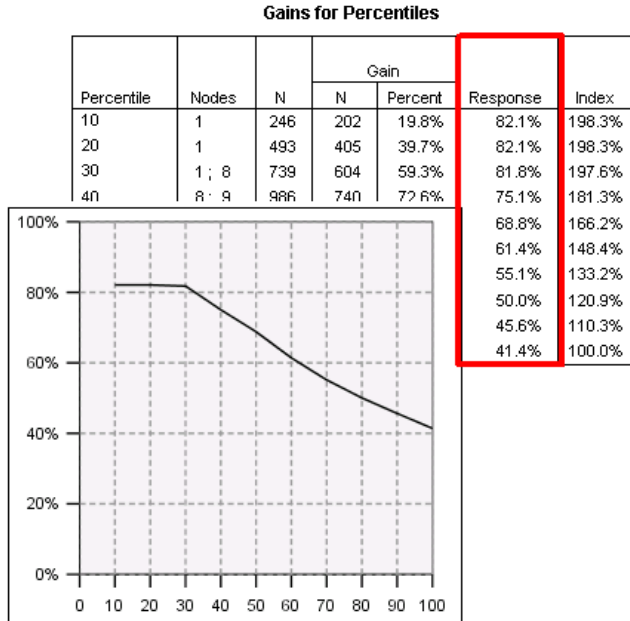


Response. The percentage of cases in the node in the specified target category.

The response chart is a line chart of cumulative percentile response, computed as: $(\text{cumulative percentile target } n / \text{cumulative percentile total } n) \times 100$. Available only for categorical dependent variables with defined target categories.

The response chart plots the same values that you would see in the *Response* column in the gains for percentiles table.

Figure 1-28
Gains for percentiles table and response chart

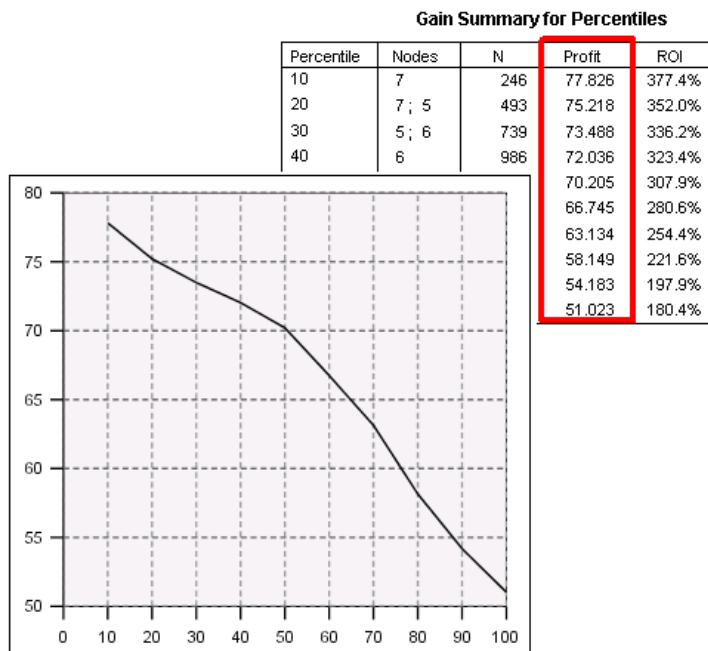


Mean. Line chart of cumulative percentile mean values for the dependent variable. Available only for scale dependent variables.

Average profit. Line chart of cumulative average profit. Available only for categorical dependent variables with defined profits. For more information, see [Profits](#) on p. 21.

The average profit chart plots the same values that you would see in the *Profit* column in the gain summary for percentiles table.

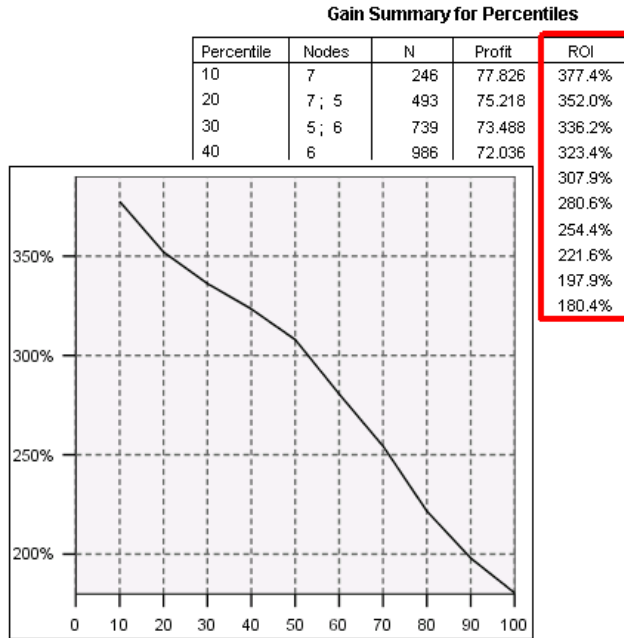
Figure 1-29
Gain summary for percentiles table and average profit chart



Return on investment (ROI). Line chart of cumulative ROI (return on investment). ROI is computed as the ratio of profits to expenses. Available only for categorical dependent variables with defined profits.

The ROI chart plots the same values that you would see in the *ROI* column in the gain summary for percentiles table.

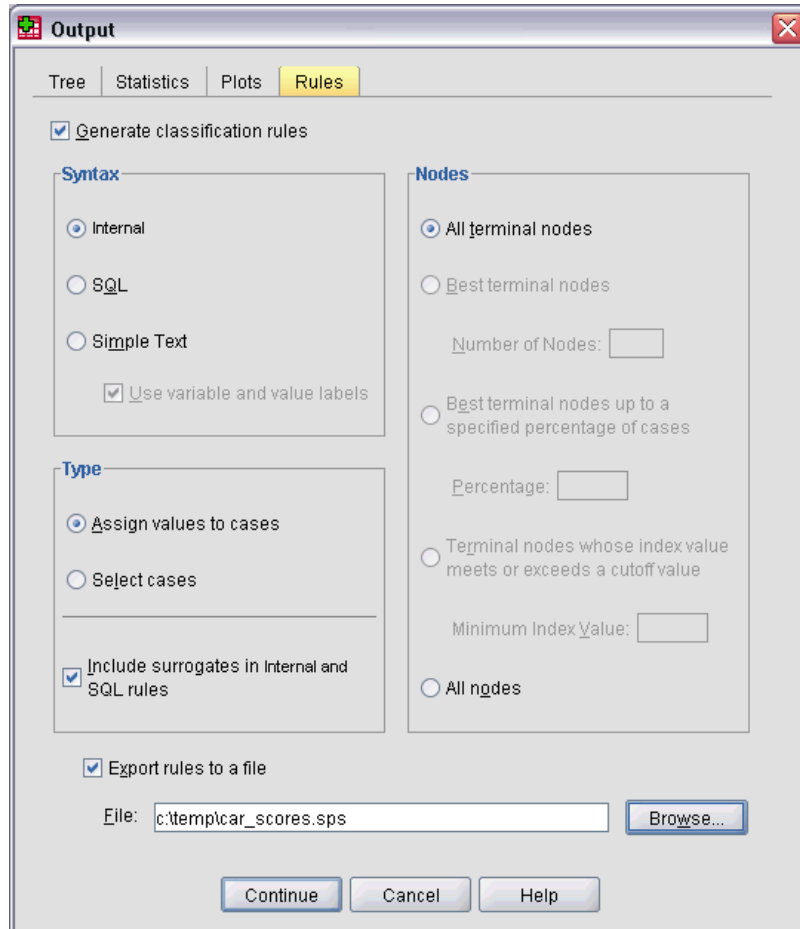
Figure 1-30
Gain summary for percentiles table and ROI chart



Percentile increment. For all percentile charts, this setting controls the percentile increments displayed on the chart: 1, 2, 5, 10, 20, or 25.

Selection and Scoring Rules

Figure 1-31
Output dialog box, Rules tab



The Rules tab provides the ability to generate selection or classification/prediction rules in the form of command syntax, SQL, or simple (plain English) text. You can display these rules in the Viewer and/or save the rules to an external file.

Syntax. Controls the form of the selection rules in both output displayed in the Viewer and selection rules saved to an external file.

- **Internal.** Command syntax language. Rules are expressed as a set of commands that define a filter condition that can be used to select subsets of cases or as COMPUTE statements that can be used to score cases.
- **SQL.** Standard SQL rules are generated to select or extract records from a database or assign values to those records. The generated SQL rules do not include any table names or other data source information.
- **Simple text.** Plain English pseudo-code. Rules are expressed as a set of logical “if...then” statements that describe the model’s classifications or predictions for each node. Rules in this form can use defined variable and value labels or variable names and data values.

Type. For Internal and SQL rules, controls the type of rules generated: selection or scoring rules.

- **Assign values to cases.** The rules can be used to assign the model’s predictions to cases that meet node membership criteria. A separate rule is generated for each node that meets the node membership criteria.
- **Select cases.** The rules can be used to select cases that meet node membership criteria. For Internal and SQL rules, a single rule is generated to select all cases that meet the selection criteria.

Include surrogates in Internal and SQL rules. For CRT and QUEST, you can include surrogate predictors from the model in the rules. Rules that include surrogates can be quite complex. In general, if you just want to derive conceptual information about your tree, exclude surrogates. If some cases have incomplete independent variable (predictor) data and you want rules that mimic your tree, include surrogates. For more information, see [Surrogates](#) on p. 18.

Nodes. Controls the scope of the generated rules. A separate rule is generated for each node included in the scope.

- **All terminal nodes.** Generates rules for each terminal node.
- **Best terminal nodes.** Generates rules for the top n terminal nodes based on index values. If the number exceeds the number of terminal nodes in the tree, rules are generated for all terminal nodes. (See note below.)
- **Best terminal nodes up to a specified percentage of cases.** Generates rules for terminal nodes for the top n percentage of cases based on index values. (See note below.)

- **Terminal nodes whose index value meets or exceeds a cutoff value.** Generates rules for all terminal nodes with an index value greater than or equal to the specified value. An index value greater than 100 means that the percentage of cases in the target category in that node exceeds the percentage in the root node. (See note below.)
- **All nodes.** Generates rules for all nodes.

Note 1: Node selection based on index values is available only for categorical dependent variables with defined target categories. If you have specified multiple target categories, a separate set of rules is generated for each target category.

Note 2: For Internal and SQL rules for selecting cases (not rules for assigning values), All nodes and All terminal nodes will effectively generate a rule that selects all cases used in the analysis.

Export rules to a file. Saves the rules in an external text file.

You can also generate and save selection or scoring rules interactively, based on selected nodes in the final tree model. For more information, see [Case Selection and Scoring Rules](#) in Chapter 2 on p. 57.

Note: If you apply rules in the form of command syntax to another data file, that data file must contain variables with the same names as the independent variables included in the final model, measured in the same metric, with the same user-defined missing values (if any).

Tree Editor

With the Tree Editor, you can:

- Hide and show selected tree branches.
- Control display of node content, statistics displayed at node splits, and other information.
- Change node, background, border, chart, and font colors.
- Change font style and size.
- Change tree alignment.
- Select subsets of cases for further analysis based on selected nodes.
- Create and save rules for selecting or scoring cases based on selected nodes.

To edit a tree model:

- ▶ Double-click the tree model in the Viewer window.

or

- ▶ From the Edit menu or the right-click context menu choose:
 - Edit Content
 - In Separate Window

Hiding and Showing Nodes

To hide (collapse) all the child nodes in a branch beneath a parent node:

- ▶ Click the minus sign (–) in the small box below the lower right corner of the parent node.

All nodes beneath the parent node on that branch will be hidden.

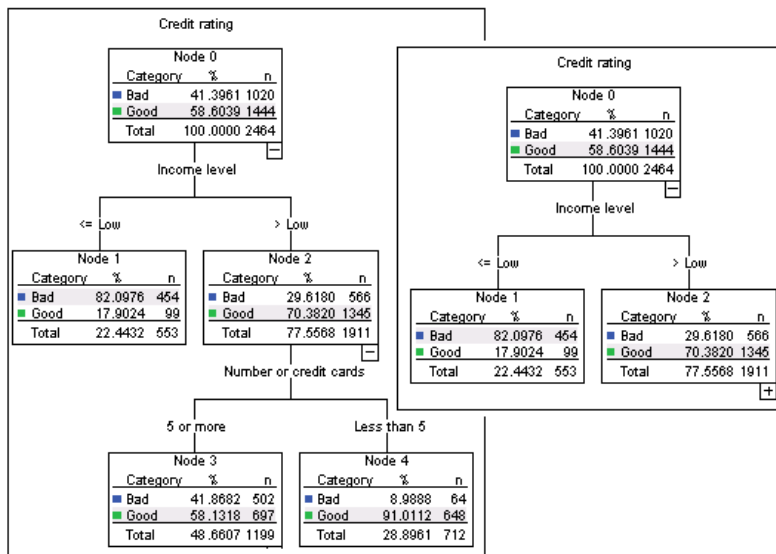
To show (expand) the child nodes in a branch beneath a parent node:

- Click the plus sign (+) in the small box below the lower right corner of the parent node.

Note: Hiding the child nodes on a branch is not the same as pruning a tree. If you want a pruned tree, you must request pruning before you create the tree, and pruned branches are not included in the final tree. For more information, see [Pruning Trees](#) in Chapter 1 on p. 17.

Figure 2-1

Expanded and collapsed tree



Selecting Multiple Nodes

You can select cases, generate scoring and selections rules, and perform other actions based on the currently selected node(s). To select multiple nodes:

- Click a node you want to select.
- Ctrl-click the other nodes you want to select.

You can multiple-select sibling nodes and/or parent nodes in one branch and child nodes in another branch. You cannot, however, use multiple selection on a parent node and a child/descendant of the same node branch.

Working with Large Trees

Tree models may sometimes contain so many nodes and branches that it is difficult or impossible to view the entire tree at full size. There are a number of features that you may find useful when working with large trees:

- **Tree map.** You can use the tree map, a much smaller, simplified version of the tree, to navigate the tree and select nodes. For more information, see [Tree Map](#) on p. 50.
- **Scaling.** You can zoom out and zoom in by changing the scale percentage for the tree display. For more information, see [Scaling the Tree Display](#) on p. 51.
- **Node and branch display.** You can make a tree more compact by displaying only tables or only charts in the nodes and/or suppressing the display of node labels or independent variable information. For more information, see [Controlling Information Displayed in the Tree](#) on p. 53.

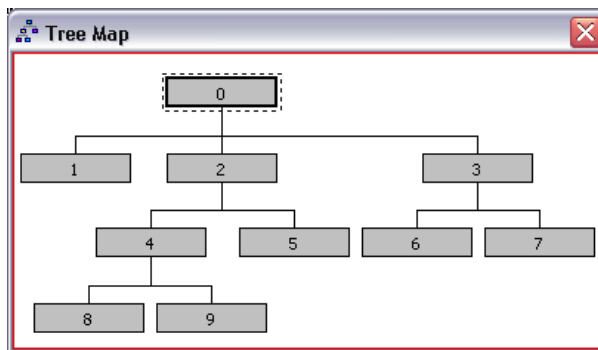
Tree Map

The tree map provides a compact, simplified view of the tree that you can use to navigate the tree and select nodes.

To use the tree map window:

- ▶ From the Tree Editor menus choose:
View
Tree Map

Figure 2-2
Tree map window



- The currently selected node is highlighted in both the Tree Model Editor and the tree map window.
- The portion of the tree that is currently in the Tree Model Editor view area is indicated with a red rectangle in the tree map. Right-click and drag the rectangle to change the section of the tree displayed in the view area.
- If you select a node in the tree map that isn't currently in the Tree Editor view area, the view shifts to include the selected node.
- Multiple node selection works the same in the tree map as in the Tree Editor: Ctrl-click to select multiple nodes. You cannot use multiple selection on a parent node and a child/descendant of the same node branch.

Scaling the Tree Display

By default, trees are automatically scaled to fit in the Viewer window, which can result in some trees that are initially very difficult to read. You can select a preset scale setting or enter your own custom scale value of between 5% and 200%.

To change the scale of the tree:

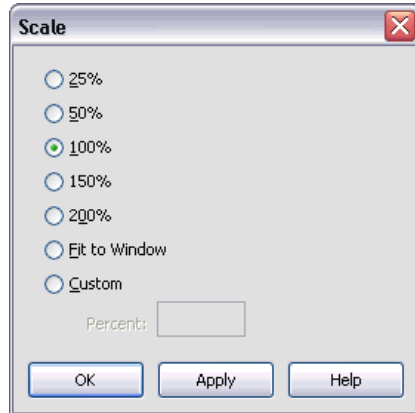
- ▶ Select a scale percentage from the drop-down list on the toolbar, or enter a custom percentage value.

or

- ▶ From the Tree Editor menus choose:

View
Scale...

Figure 2-3
Scale dialog box



You can also specify a scale value before you create the tree model. For more information, see [Output](#) in Chapter 1 on p. 30.

Node Summary Window

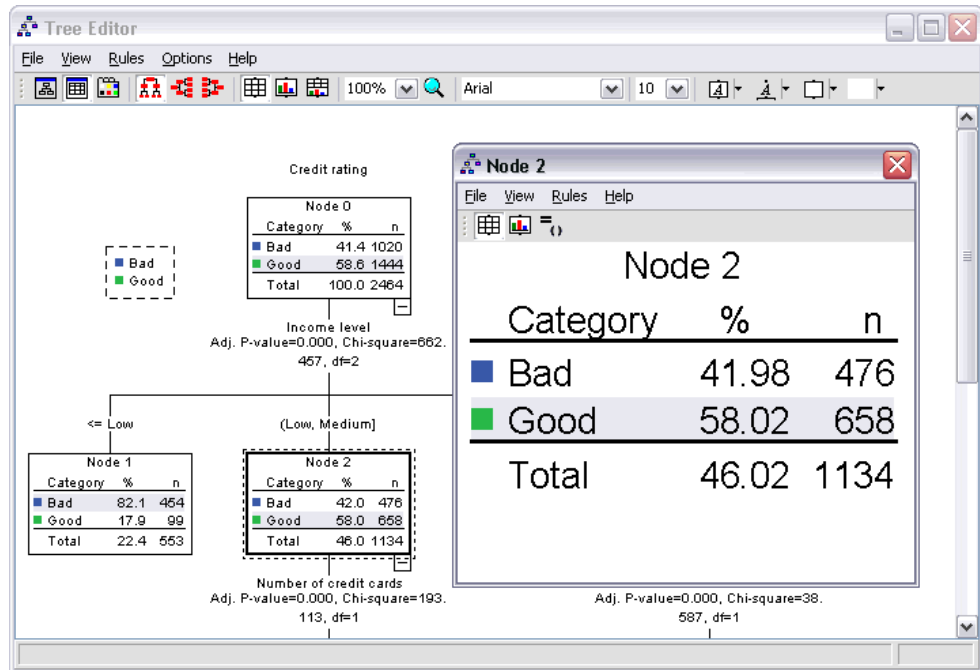
The node summary window provides a larger view of the selected nodes. You can also use the summary window to view, apply, or save selection or scoring rules based on the selected nodes.

- Use the View menu in the node summary window to switch between views of a summary table, chart, or rules.
- Use the Rules menu in the node summary window to select the type of rules you want to see. For more information, see [Case Selection and Scoring Rules](#) on p. 57.
- All views in the node summary window reflect a combined summary for all selected nodes.

To use the node summary window:

- ▶ Select the nodes in the Tree Editor. To select multiple nodes, use Ctrl-click.
- ▶ From the menus choose:
View
Summary

Figure 2-4
Summary window

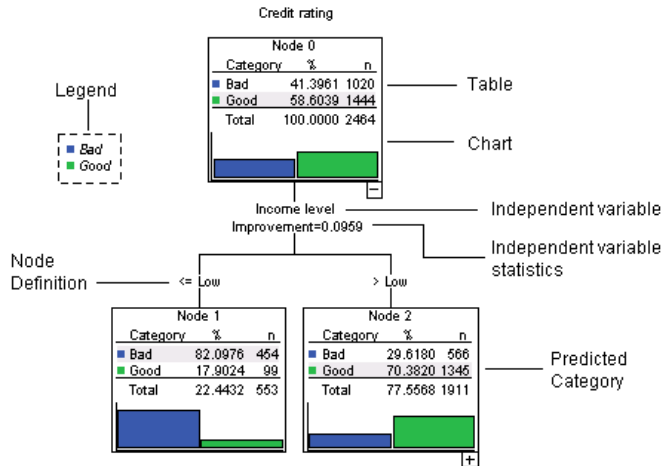


Controlling Information Displayed in the Tree

The Options menu in the Tree Editor allows you to control the display of node contents, independent variable (predictor) names and statistics, node definitions, and other settings. Many of these settings can be also be controlled from the toolbar.

Setting	Options Menu Selection
Highlight predicted category (categorical dependent variable)	Highlight Predicted
Tables and/or charts in node	Node Contents
Significance test values and p values	Independent Variable Statistics
Independent (predictor) variable names	Independent Variables
Independent (predictor) value(s) for nodes	Node Definitions
Alignment (top-down, left-right, right-left)	Orientation
Chart legend	Legend

Figure 2-5
Tree elements



Changing Tree Colors and Text Fonts

You can change the following colors in the tree:

- Node border, background, and text color
- Branch color and branch text color
- Tree background color
- Predicted category highlight color (categorical dependent variables)
- Node chart colors

You can also change the type font, style, and size for all text in the tree.

Note: You cannot change color or font attributes for individual nodes or branches. Color changes apply to all elements of the same type, and font changes (other than color) apply to all chart elements.

To change colors and text font attributes:

- ▶ Use the toolbar to change font attributes for the entire tree or colors for different tree elements. (ToolTips describe each control on the toolbar when you put the mouse cursor on the control.)

or

- ▶ Double-click anywhere in the Tree Editor to open the Properties window, or from the menus choose:

View
Properties

- ▶ For border, branch, node background, predicted category, and tree background, click the Color tab.
- ▶ For font colors and attributes, click the Text tab.
- ▶ For node chart colors, click the Node Charts tab.

Figure 2-6

Properties window, Color tab

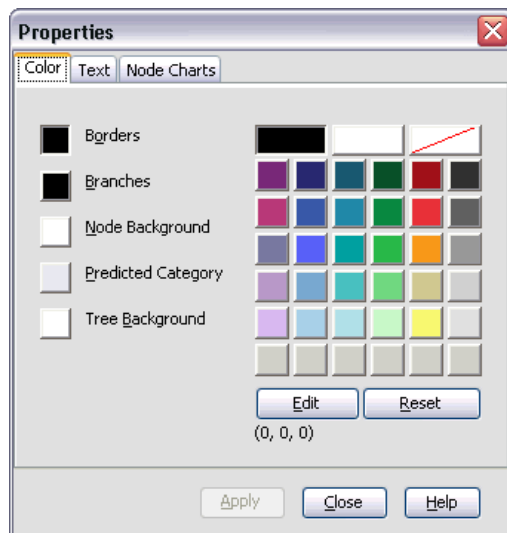


Figure 2-7
Properties window, Text tab

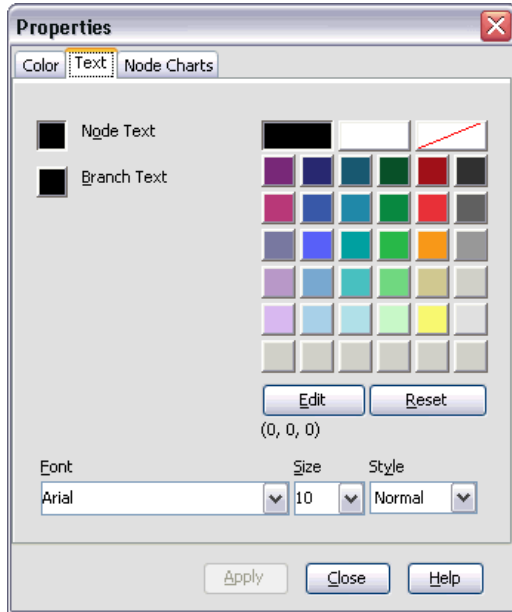
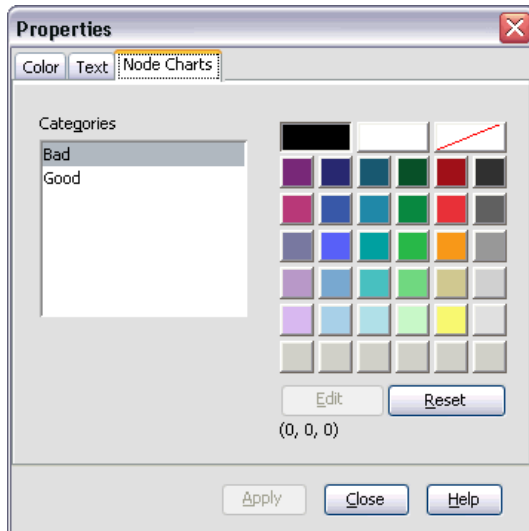


Figure 2-8
Properties window, Node Charts tab



Case Selection and Scoring Rules

You can use the Tree Editor to:

- Select subsets of cases based on the selected node(s). For more information, see [Filtering Cases](#) on p. 57.
- Generate case selection rules or scoring rules in SPSS Statistics command syntax or SQL format. For more information, see [Saving Selection and Scoring Rules](#) on p. 58.

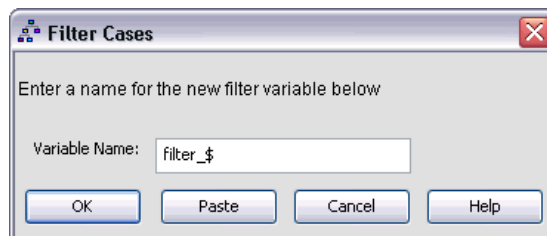
You can also automatically save rules based on various criteria when you run the Decision Tree procedure to create the tree model. For more information, see [Selection and Scoring Rules](#) in Chapter 1 on p. 45.

Filtering Cases

If you want to know more about the cases in a particular node or group of nodes, you can select a subset of cases for further analysis based on the selected nodes.

- ▶ Select the nodes in the Tree Editor. To select multiple nodes, use Ctrl-click.
- ▶ From the menus choose:
Rules
Filter Cases...
- ▶ Enter a filter variable name. Cases from the selected nodes will receive a value of 1 for this variable. All other cases will receive a value of 0 and will be excluded from subsequent analysis until you change the filter status.
- ▶ Click OK.

Figure 2-9
Filter Cases dialog box



Saving Selection and Scoring Rules

You can save case selection or scoring rules in an external file and then apply those rules to a different data source. The rules are based on the selected nodes in the Tree Editor.

Syntax. Controls the form of the selection rules in both output displayed in the Viewer and selection rules saved to an external file.

- **Internal.** Command syntax language. Rules are expressed as a set of commands that define a filter condition that can be used to select subsets of cases or as COMPUTE statements that can be used to score cases.
- **SQL.** Standard SQL rules are generated to select/extract records from a database or assign values to those records. The generated SQL rules do not include any table names or other data source information.

Type. You can create selection or scoring rules.

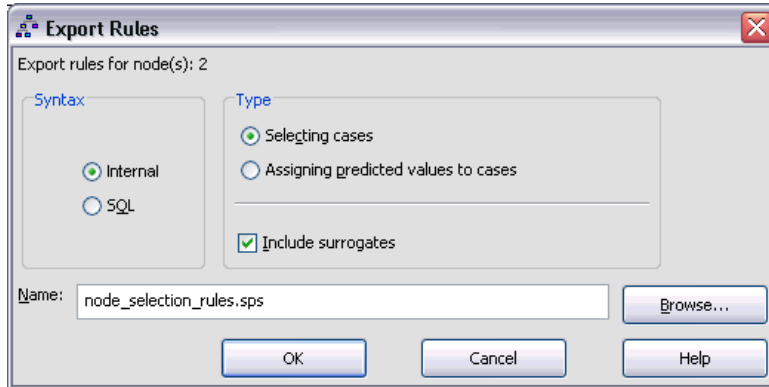
- **Select cases.** The rules can be used to select cases that meet node membership criteria. For Internal and SQL rules, a single rule is generated to select all cases that meet the selection criteria.
- **Assign values to cases.** The rules can be used to assign the model's predictions to cases that meet node membership criteria. A separate rule is generated for each node that meets the node membership criteria.

Include surrogates. For CRT and QUEST, you can include surrogate predictors from the model in the rules. Rules that include surrogates can be quite complex. In general, if you just want to derive conceptual information about your tree, exclude surrogates. If some cases have incomplete independent variable (predictor) data and you want rules that mimic your tree, include surrogates. For more information, see [Surrogates](#) in Chapter 1 on p. 18.

To save case selection or scoring rules:

- ▶ Select the nodes in the Tree Editor. To select multiple nodes, use Ctrl-click.
- ▶ From the menus choose:
 - Rules
 - Export...
- ▶ Select the type of rules you want and enter a filename.

Figure 2-10
Export Rules dialog box



Note: If you apply rules in the form of command syntax to another data file, that data file must contain variables with the same names as the independent variables included in the final model, measured in the same metric, with the same user-defined missing values (if any).

Part II: Examples

Data Assumptions and Requirements

The Decision Tree procedure assumes that:

- The appropriate measurement level has been assigned to all analysis variables.
- For categorical (**nominal, ordinal**) dependent variables, value labels have been defined for all categories that should be included in the analysis.

We'll use the file *tree_textdata.sav* to illustrate the importance of both of these requirements. This data file reflects the default state of data read or entered before defining any attributes, such as measurement level or value labels. For more information, see [Sample Files](#) in Appendix A on p. 120.

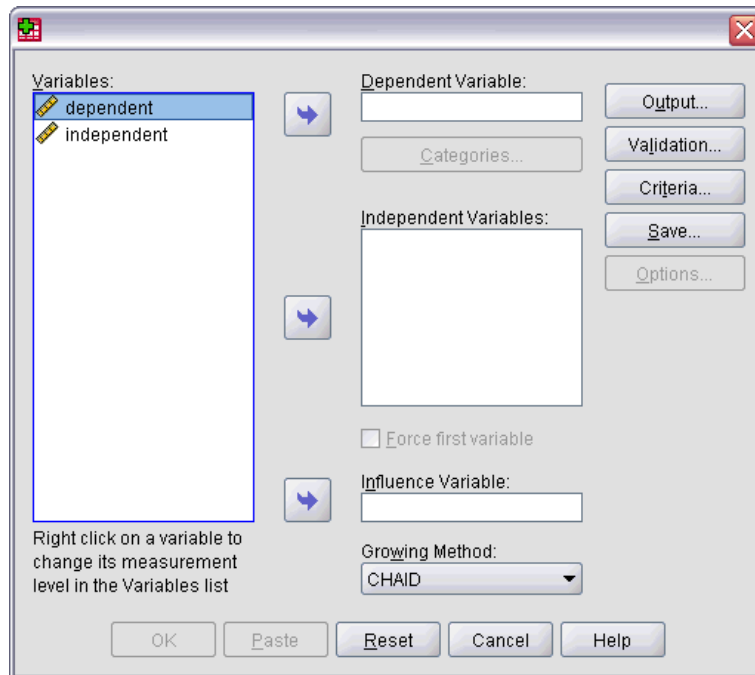
Effects of Measurement Level on Tree Models

Both variables in this data file are numeric. By default, numeric variables are assumed to have a **scale** measurement level. But (as we will see later) both variables are really categorical variables that rely on numeric codes to stand for category values.

- ▶ To run a Decision Tree analysis, from the menus choose:
Analyze
Classify
Tree...

The icons next to the two variables in the source variable list indicate that they will be treated as scale variables.

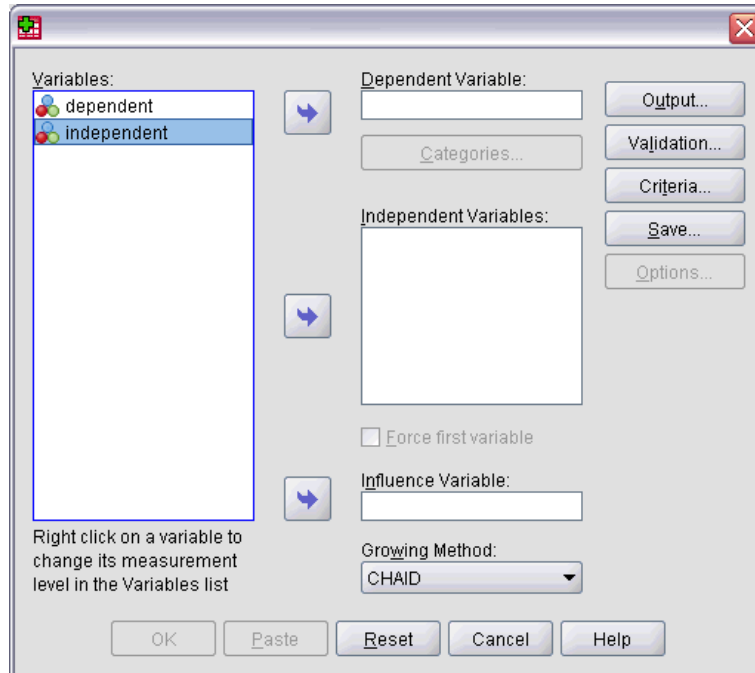
Figure 3-1
Decision Tree main dialog box with two scale variables



- ▶ Select *dependent* as the dependent variable.
- ▶ Select *independent* as the independent variable.
- ▶ Click OK to run the procedure.
- ▶ Open the Decision Tree dialog box again and click Reset.
- ▶ Right-click *dependent* in the source list and select Nominal from the context menu.
- ▶ Do the same for the variable *independent* in the source list.

Now the icons next to each variable indicate that they will be treated as nominal variables.

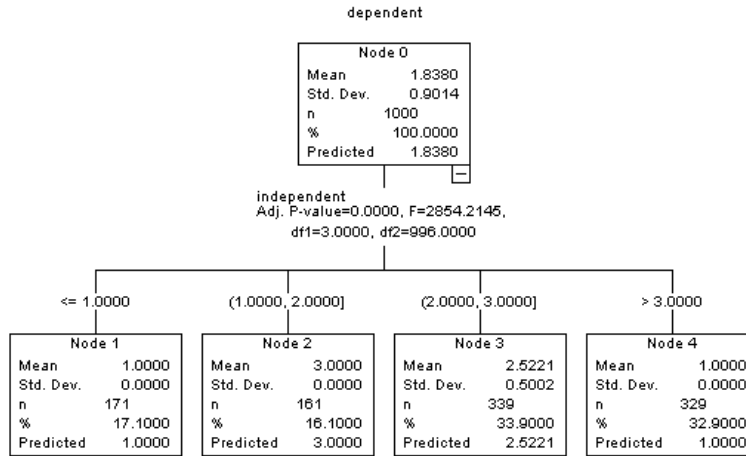
Figure 3-2
Nominal icons in source list



- ▶ Select *dependent* as the dependent variable and *independent* as the independent variable, and click OK to run the procedure again.

Now let's compare the two trees. First, we'll look at the tree in which both numeric variables are treated as scale variables.

Figure 3-3
Tree with both variables treated as scale

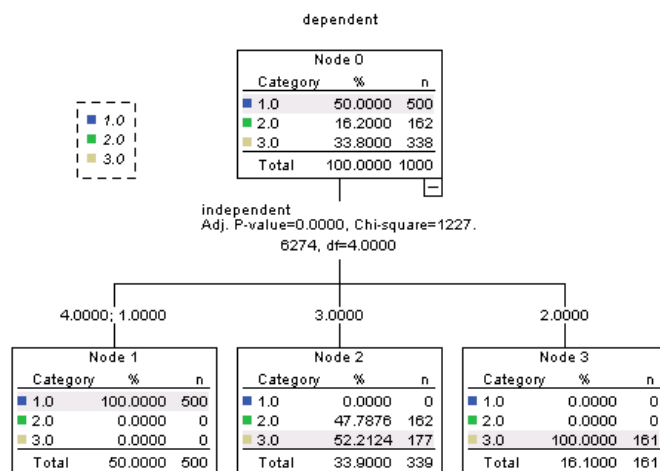


- Each node of tree shows the “predicted” value, which is the mean value for the dependent variable at that node. For a variable that is actually categorical, the mean may not be a meaningful statistic.
- The tree has four child nodes, one for each value of the independent variable.

Tree models will often merge similar nodes, but for a scale variable, only contiguous values can be merged. In this example, no contiguous values were considered similar enough to merge any nodes together.

The tree in which both variables are treated as nominal is somewhat different in several respects.

Figure 3-4
Tree with both variables treated as nominal



- Instead of a predicted value, each node contains a frequency table that shows the number of cases (count and percentage) for each category of the dependent variable.
- The “predicted” category—the category with the highest count in each node—is highlighted. For example, the predicted category for node 2 is category 3.
- Instead of four child nodes, there are only three, with two values of the independent variable merged into a single node.

The two independent values merged into the same node are 1 and 4. Since, by definition, there is no inherent order to nominal values, merging of noncontiguous values is allowed.

Permanently Assigning Measurement Level

When you change the measurement level for a variable in the Decision Tree dialog box, the change is only temporary; it is not saved with the data file. Furthermore, you may not always know what the correct measurement level should be for all variables.

Define Variable Properties can help you determine the correct measurement level for each variable and permanently change the assigned measurement level. To use Define Variable Properties:

- ▶ From the menus choose:
 - Data
 - Define Variable Properties...

Effects of Value Labels on Tree Models

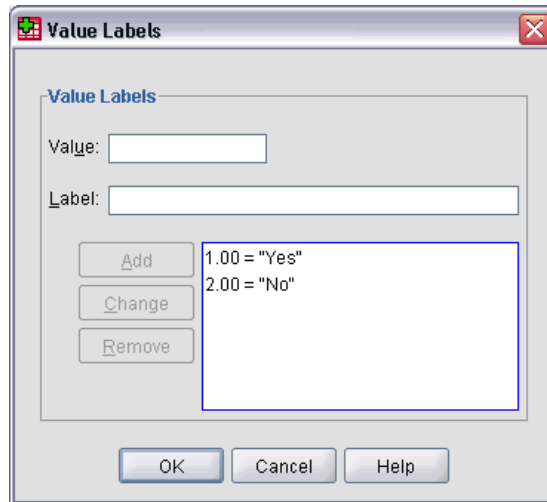
The Decision Tree dialog box interface assumes that either *all* nonmissing values of a categorical (nominal, ordinal) dependent variable have defined value labels or *none* of them do. Some features are not available unless at least two nonmissing values of the categorical dependent variable have value labels. If at least two nonmissing values have defined value labels, any cases with other values that do not have value labels will be excluded from the analysis.

The original data file in this example contains no defined value labels, and when the dependent variable is treated as nominal, the tree model uses all nonmissing values in the analysis. In this example, those values are 1, 2, and 3.

But what happens when we define value labels for some, but not all, values of the dependent variable?

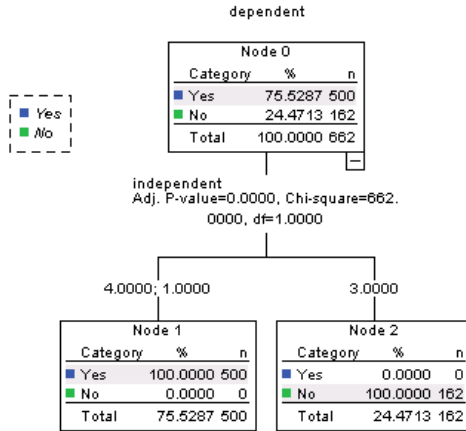
- ▶ In the Data Editor window, click the Variable View tab.
- ▶ Click the Values cell for the variable *dependent*.

Figure 3-5
Defining value labels for dependent variable



- ▶ First, enter 1 for Value and Yes for Value Label, and then click Add.
- ▶ Next, enter 2 for Value and No for Value Label, and then click Add again.
- ▶ Then click OK.
- ▶ Open the Decision Tree dialog box again. The dialog box should still have *dependent* selected as the dependent variable, with a nominal measurement level.
- ▶ Click OK to run the procedure again.

Figure 3-6
Tree for nominal dependent variable with partial value labels



Now only the two dependent variable values with defined value labels are included in the tree model. All cases with a value of 3 for the dependent variable have been excluded, which might not be readily apparent if you aren't familiar with the data.

Assigning Value Labels to All Values

To avoid accidental omission of valid categorical values from the analysis, use Define Variable Properties to assign value labels to all dependent variable values found in the data.

When the data dictionary information for the variable *name* is displayed in the Define Variable Properties dialog box, you can see that although there are over 300 cases with a value of 3 for that variable, no value label has been defined for that value.

Figure 3-7

Variable with partial value labels in Define Variable Properties dialog box

Define Variable Properties

Scanned Variable List: U... M... Variable
 dependent

Current Variable: dependent Label:

Measurement Level: Nominal Type: Numeric

Unlabeled values: 1 Width: 8 Decimals: 2

Value Label grid: Enter or edit labels in the grid. You can enter additional values at the bottom.

	Changed	Missing	Count	Value	Label
1	<input type="checkbox"/>	<input type="checkbox"/>	500	1.00	Yes
2	<input type="checkbox"/>	<input type="checkbox"/>	162	2.00	No
3	<input type="checkbox"/>	<input type="checkbox"/>	338	3.00	
4	<input type="checkbox"/>	<input type="checkbox"/>			

Cases scanned: 4

Value list limit: 200

Copy Properties

Unlabeled Values

Using Decision Trees to Evaluate Credit Risk

A bank maintains a database of historic information on customers who have taken out loans from the bank, including whether or not they repaid the loans or defaulted. Using a tree model, you can analyze the characteristics of the two groups of customers and build models to predict the likelihood that loan applicants will default on their loans.

The credit data are stored in *tree_credit.sav*. For more information, see [Sample Files](#) in Appendix A on p. 120.

Creating the Model

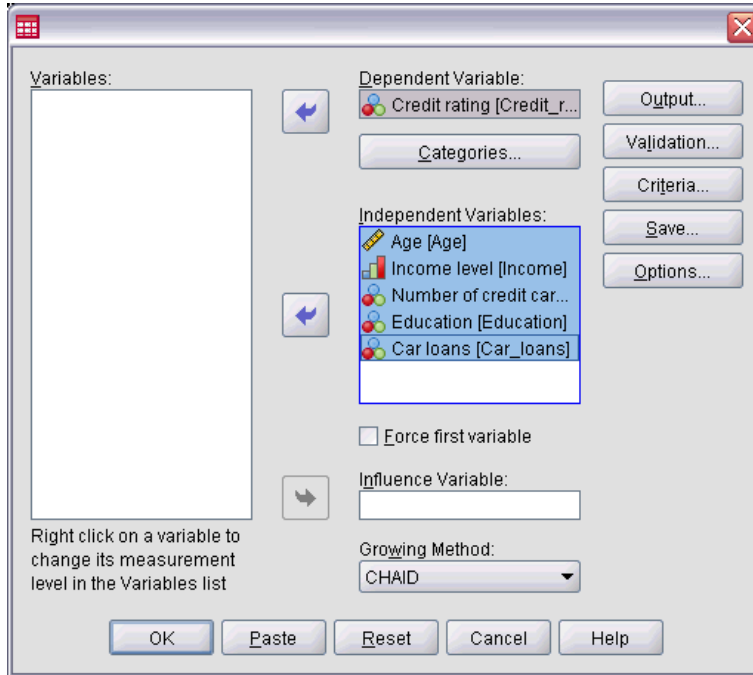
The Decision Tree Procedure offers several different methods for creating tree models. For this example, we'll use the default method:

CHAID. Chi-squared Automatic Interaction Detection. At each step, CHAID chooses the independent (predictor) variable that has the strongest interaction with the dependent variable. Categories of each predictor are merged if they are not significantly different with respect to the dependent variable.

Building the CHAID Tree Model

- ▶ To run a Decision Tree analysis, from the menus choose:
 - Analyze
 - Classify
 - Tree...

Figure 4-1
Decision Tree dialog box



- ▶ Select *Credit rating* as the dependent variable.
- ▶ Select all the remaining variables as independent variables. (The procedure will automatically exclude any variables that don't make a significant contribution to the final model.)

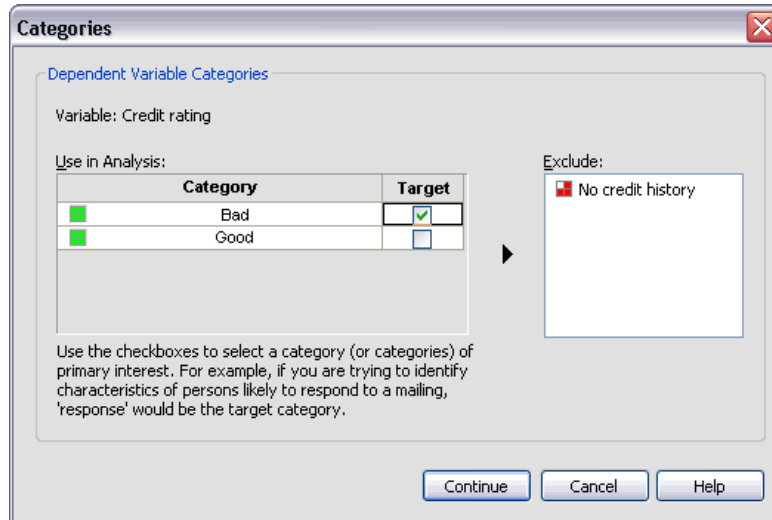
At this point, you could run the procedure and produce a basic tree model, but we're going to select some additional output and make a few minor adjustments to the criteria used to generate the model.

Selecting Target Categories

- ▶ Click the Categories button right below the selected dependent variable.

This opens the Categories dialog box, where you can specify the dependent variable target categories of interest. Target categories do not affect the tree model itself, but some output and options are available only if you have selected target categories.

Figure 4-2
Categories dialog box



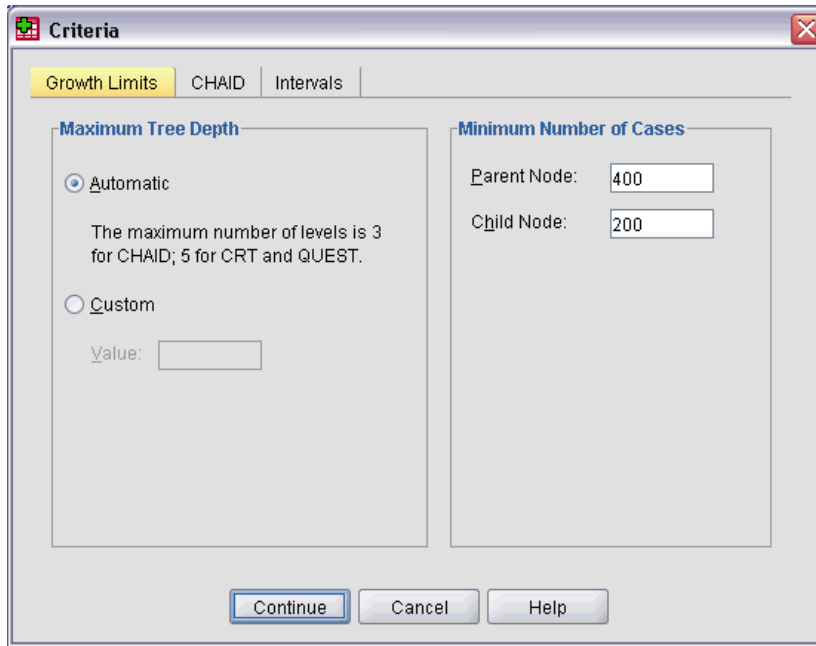
- ▶ Select (check) the Target check box for the *Bad* category. Customers with a bad credit rating (defaulted on a loan) will be treated as the target category of interest.
- ▶ Click Continue.

Specifying Tree Growing Criteria

For this example, we want to keep the tree fairly simple, so we'll limit the tree growth by raising the minimum number of cases for parent and child nodes.

- ▶ In the main Decision Tree dialog box, click Criteria.

Figure 4-3
Criteria dialog box, Growth Limits tab



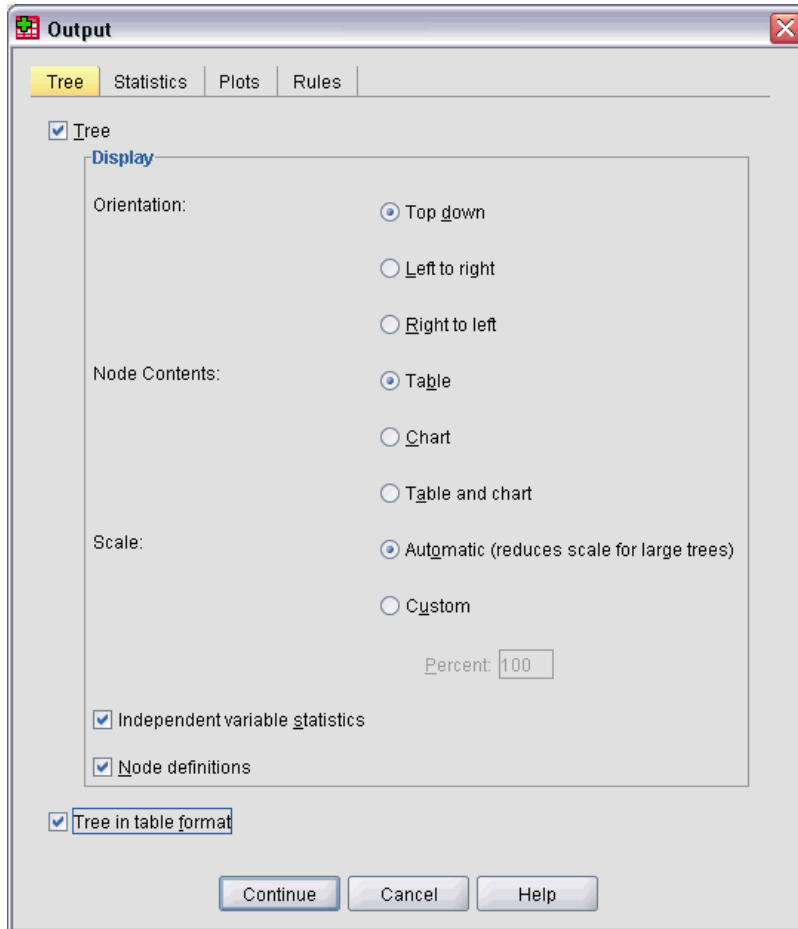
- ▶ In the Minimum Number of Cases group, type 400 for Parent Node and 200 for Child Node.
- ▶ Click Continue.

Selecting Additional Output

- ▶ In the main Decision Tree dialog box, click Output.

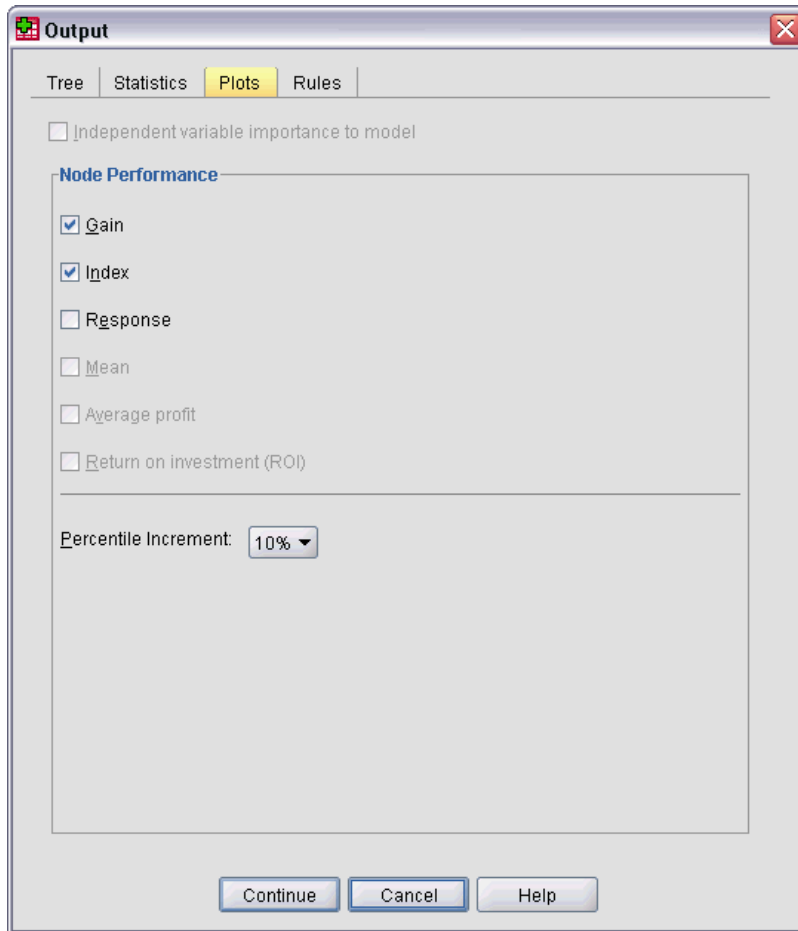
This opens a tabbed dialog box, where you can select various types of additional output.

Figure 4-4
Output dialog box, Tree tab



- ▶ On the Tree tab, select (check) Tree in table format.
- ▶ Then click the Plots tab.

Figure 4-5
Output dialog box, Plots tab



- ▶ Select (check) Gain and Index.

Note: These charts require a target category for the dependent variable. In this example, the Plots tab isn't accessible until after you have specified one or more target categories.

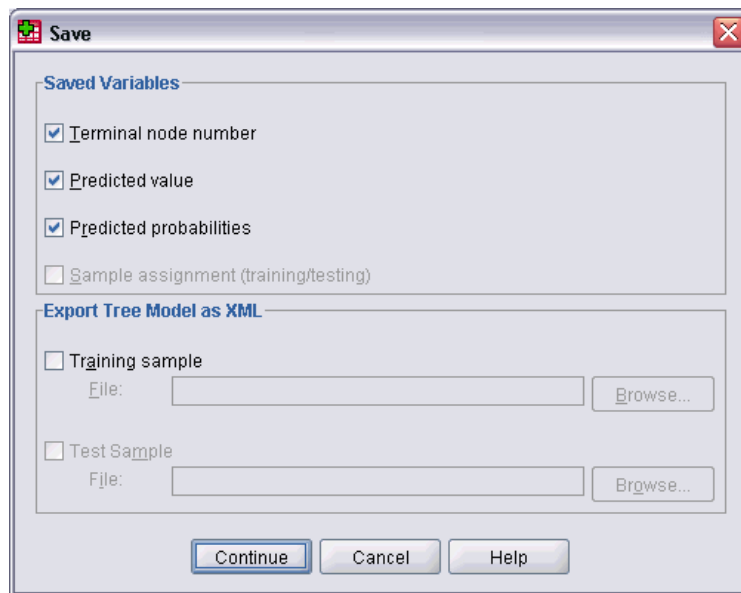
- ▶ Click Continue.

Saving Predicted Values

You can save variables that contain information about model predictions. For example, you can save the credit rating predicted for each case and then compare those predictions to the actual credit ratings.

- ▶ In the main Decision Tree dialog box, click Save.

Figure 4-6
Save dialog box



- ▶ Select (check) Terminal node number, Predicted value, and Predicted probabilities.
- ▶ Click Continue.
- ▶ In the main Decision Tree dialog box, click OK to run the procedure.

Evaluating the Model

For this example, the model results include:

- Tables that provide information about the model.

- Tree diagram.
- Charts that provide an indication of model performance.
- Model prediction variables added to the active dataset.

Model Summary Table

Figure 4-7
Model summary

Specifications	Growing Method	CHAID
	Dependent Variable	Credit rating
	Independent Variables	Age, Income, Credit cards, Education, Car loans
	Validation	NONE
	Maximum Tree Depth	3
	Minimum Cases in Parent Node	400
	Minimum Cases in Child Node	200
Results	Independent Variables Included	Age, Income, Credit cards
	Number of Nodes	10
	Number of Terminal Nodes	6
	Depth	3

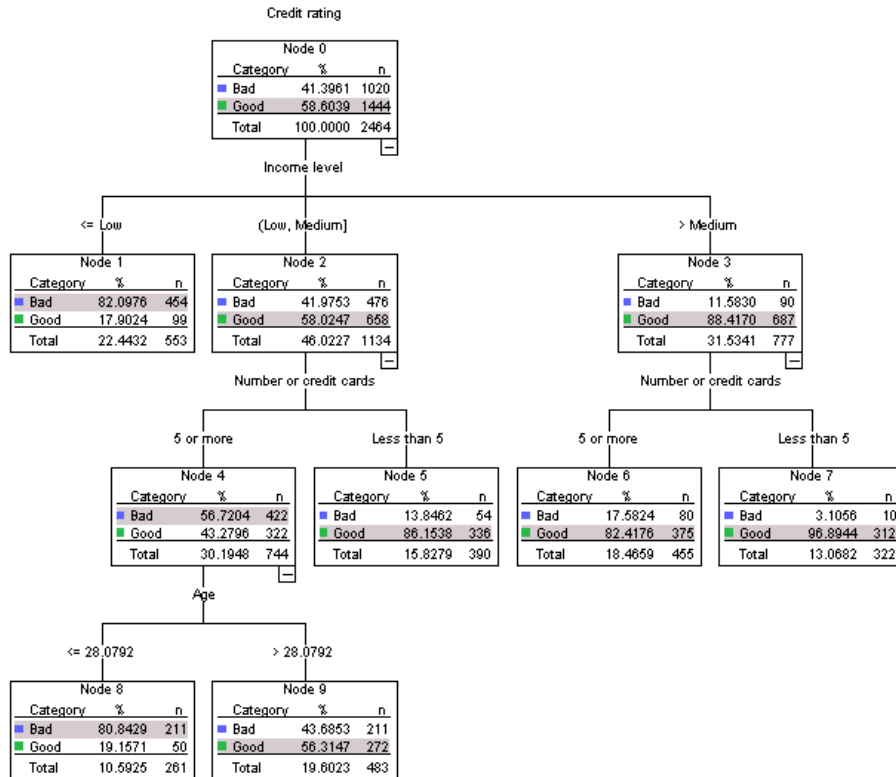
The model summary table provides some very broad information about the specifications used to build the model and the resulting model.

- The Specifications section provides information on the settings used to generate the tree model, including the variables used in the analysis.
- The Results section displays information on the number of total and terminal nodes, depth of the tree (number of levels below the root node), and independent variables included in the final model.

Five independent variables were specified, but only three were included in the final model. The variables for *education* and number of current *car loans* did not make a significant contribution to the model, so they were automatically dropped from the final model.

Tree Diagram

Figure 4-8
Tree diagram for credit rating model



The tree diagram is a graphic representation of the tree model. This tree diagram shows that:

- Using the CHAID method, *income level* is the best predictor of *credit rating*.
- For the low income category, *income level* is the only significant predictor of *credit rating*. Of the bank customers in this category, 82% have defaulted on loans. Since there are no child nodes below it, this is considered a **terminal** node.
- For the medium and high income categories, the next best predictor is *number of credit cards*.

- For medium income customers with five or more credit cards, the model includes one more predictor: *age*. Over 80% of those customers 28 or younger have a bad credit rating, while slightly less than half of those over 28 have a bad credit rating.

You can use the Tree Editor to hide and show selected branches, change colors and fonts, and select subsets of cases based on selected nodes. For more information, see [Selecting Cases in Nodes](#) on p. 86.

Tree Table

Figure 4-9
Tree table for credit rating

Node	Bad		Good		Total		Predicted Category	Parent Node
	N	Percent	N	Percent	N	Percent		
0	1020	41.4%	1444	58.6%	2464	100.0%	Good	
1	454	82.1%	99	17.9%	553	22.4%	Bad	0
2	476	42.0%	658	58.0%	1134	46.0%	Good	0
3	90	11.6%	687	88.4%	777	31.5%	Good	0
4	422	56.7%	322	43.3%	744	30.2%	Bad	2
5	54	13.8%	336	86.2%	390	15.8%	Good	2
6	80	17.6%	375	82.4%	455	18.5%	Good	3
7	10	3.1%	312	96.9%	322	13.1%	Good	3
8	211	80.8%	50	19.2%	261	10.6%	Bad	4
9	211	43.7%	272	56.3%	483	19.6%	Good	4

The tree table, as the name suggests, provides most of the essential tree diagram information in the form of a table. For each node, the table displays:

- The number and percentage of cases in each category of the dependent variable.
- The predicted category for the dependent variable. In this example, the predicted category is the *credit rating* category with more than 50% of cases in that node, since there are only two possible credit ratings.
- The parent node for each node in the tree. Note that node 1—the low income level node—is not the parent node of any node. Since it is a terminal node, it has no child nodes.

Figure 4-10
Tree table for credit rating (continued)

Primary Independent Variable				
Variable	Sig.	Chi-Square	df	Split Values
Income level	.000	662.457	2	<= Low
Income level	.000	662.457	2	(Low, Medium]
Income level	.000	662.457	2	> Medium
Number or credit cards	.000	193.113	1	5 or more
Number or credit cards	.000	193.113	1	Less than 5
Number or credit cards	.000	38.587	1	5 or more
Number or credit cards	.000	38.587	1	Less than 5
Age	.000	95.299	1	<= 28.0792
Age	.000	95.299	1	> 28.0792

- The independent variable used to split the node.
- The chi-square value (since the tree was generated with the CHAID method), degrees of freedom (*df*), and significance level (*Sig.*) for the split. For most practical purposes, you will probably be interested only in the significance level, which is less than 0.0001 for all splits in this model.
- The value(s) of the independent variable for that node.

Note: For ordinal and scale independent variables, you may see ranges in the tree and tree table expressed in the general form (*value1, value2*), which basically means “greater than value1 and less than or equal to value2.” In this example, income level has only three possible values—*Low*, *Medium*, and *High*—and (*Low, Medium*) simply means *Medium*. In a similar fashion, *>Medium* means *High*.

Gains for Nodes

Figure 4-11
Gains for nodes

Node	Node		Gain		Response	Index
	N	Percent	N	Percent		
1	553	22.4%	454	44.5%	82.1%	198.3%
8	261	10.6%	211	20.7%	80.8%	195.3%
9	483	19.6%	211	20.7%	43.7%	105.5%
6	455	18.5%	80	7.8%	17.6%	42.5%
5	390	15.8%	54	5.3%	13.8%	33.4%
7	322	13.1%	10	1.0%	3.1%	7.5%

Growing Method: CHAID
Dependent Variable: Credit rating

The gains for nodes table provides a summary of information about the terminal nodes in the model.

- Only the terminal nodes—nodes at which the tree stops growing—are listed in this table. Frequently, you will be interested only in the terminal nodes, since they represent the best classification predictions for the model.
- Since gain values provide information about target categories, this table is available only if you specified one or more target categories. In this example, there is only one target category, so there is only one gains for nodes table.
- *Node N* is the number of cases in each terminal node, and *Node Percent* is the percentage of the total number of cases in each node.
- *Gain N* is the number of cases in each terminal node in the target category, and *Gain Percent* is the percentage of cases in the target category with respect to the overall number of cases in the target category—in this example, the number and percentage of cases with a bad credit rating.
- For categorical dependent variables, *Response* is the percentage of cases in the node in the specified target category. In this example, these are the same percentages displayed for the *Bad* category in the tree diagram.
- For categorical dependent variables, *Index* is the ratio of the response percentage for the target category compared to the response percentage for the entire sample.

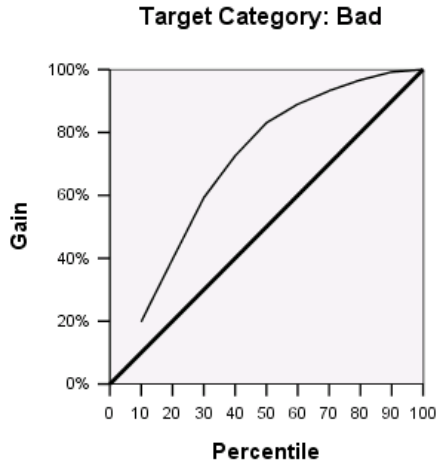
Index Values

The index value is basically an indication of how far the *observed* target category percentage for that node differs from the *expected* percentage for the target category. The target category percentage in the root node represents the expected percentage before the effects of any of the independent variables are considered.

An index value of greater than 100% means that there are more cases in the target category than the overall percentage in the target category. Conversely, an index value of less than 100% means there are fewer cases in the target category than the overall percentage.

Gains Chart

Figure 4-12
Gains chart for bad credit rating target category

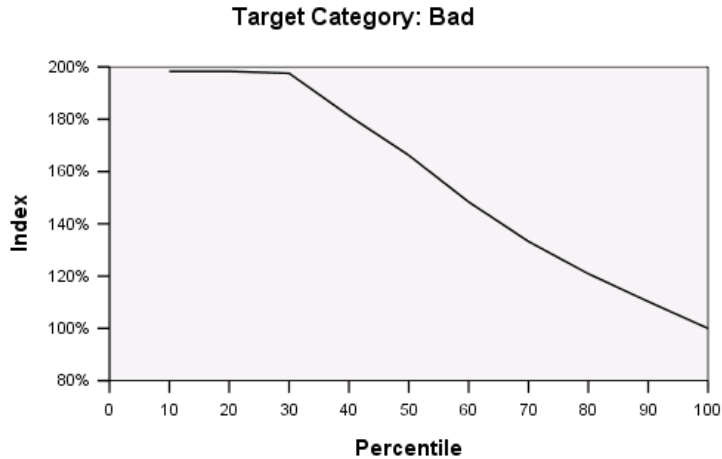


This gains chart indicates that the model is a fairly good one.

Cumulative gains charts always start at 0% and end at 100% as you go from one end to the other. For a good model, the gains chart will rise steeply toward 100% and then level off. A model that provides no information will follow the diagonal reference line.

Index Chart

Figure 4-13
Index chart for bad credit rating target category



The index chart also indicates that the model is a good one. Cumulative index charts tend to start above 100% and gradually descend until they reach 100%.

For a good model, the index value should start well above 100%, remain on a high plateau as you move along, and then trail off sharply toward 100%. For a model that provides no information, the line will hover around 100% for the entire chart.

Risk Estimate and Classification

Figure 4-14
Risk and classification tables

Risk			
Estimate	Std. Error		
.205	.008		
Growing Method: CHAID			
Dependent Variable: Credit rating			

Classification			
Observed	Predicted		
	Bad	Good	Percent Correct
Bad	665	355	65.2%
Good	149	1295	89.7%
Overall Percentage	33.0%	67.0%	79.5%

Growing Method: CHAID
Dependent Variable: Credit rating

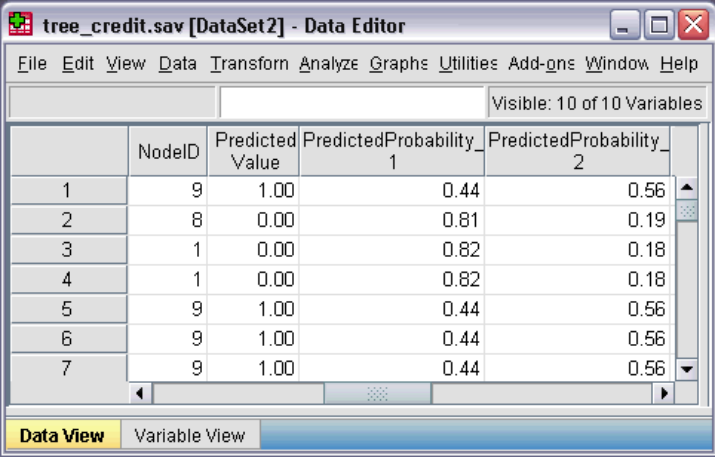
The risk and classification tables provide a quick evaluation of how well the model works.

- The risk estimate of 0.205 indicates that the category predicted by the model (good or bad credit rating) is wrong for 20.5% of the cases. So the “risk” of misclassifying a customer is approximately 21%.
- The results in the classification table are consistent with the risk estimate. The table shows that the model classifies approximately 79.5% of the customers correctly.

The classification table does, however, reveal one potential problem with this model: for those customers with a bad credit rating, it predicts a bad rating for only 65% of them, which means that 35% of customers with a bad credit rating are inaccurately classified with the “good” customers.

Predicted Values

Figure 4-15
New variables for predicted values and probabilities



	NodeID	Predicted Value	PredictedProbability_1	PredictedProbability_2
1	9	1.00	0.44	0.56
2	8	0.00	0.81	0.19
3	1	0.00	0.82	0.18
4	1	0.00	0.82	0.18
5	9	1.00	0.44	0.56
6	9	1.00	0.44	0.56
7	9	1.00	0.44	0.56

Four new variables have been created in the active dataset:

NodeID. The terminal node number for each case.

PredictedValue. The predicted value of the dependent variable for each case. Since the dependent variable is coded 0 = *Bad* and 1 = *Good*, a predicted value of 0 means that the case is predicted to have a bad credit rating.

PredictedProbability. The probability that the case belongs in each category of the dependent variable. Since there are only two possible values for the dependent variable, two variables are created:

- **PredictedProbability_1.** The probability that the case belongs in the bad credit rating category.
- **PredictedProbability_2.** The probability that the case belongs in the good credit rating category.

The predicted probability is simply the proportion of cases in each category of the dependent variable for the terminal node that contains each case. For example, in node 1, 82% of the cases are in the bad category and 18% are in the good category, resulting in predicted probabilities of 0.82 and 0.18, respectively.

For a categorical dependent variable, the predicted value is the category with the highest proportion of cases in the terminal node for each case. For example, for the first case, the predicted value is 1 (good credit rating), since approximately 56% of the cases in its terminal node have a good credit rating. Conversely, for the second case, the predicted value is 0 (bad credit rating), since approximately 81% of cases in its terminal node have a bad credit rating.

If you have defined costs, however, the relationship between predicted category and predicted probabilities may not be quite so straightforward. For more information, see [Assigning Costs to Outcomes](#) on p. 92.

Refining the Model

Overall, the model has a correct classification rate of just under 80%. This is reflected in most of the terminal nodes, where the predicted category—the highlighted category in the node—is the same as the actual category for 80% or more of the cases.

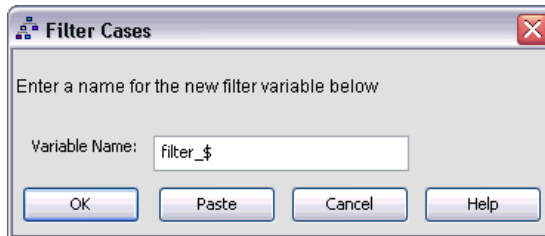
There is, however, one terminal node where cases are fairly evenly split between good and bad credit ratings. In node 9, the predicted credit rating is “good,” but only 56% of the cases in that node actually have a good credit rating. That means that almost half of the cases in that node (44%) will have the wrong predicted category. And if the primary concern is identifying bad credit risks, this node doesn’t perform very well.

Selecting Cases in Nodes

Let’s look at the cases in node 9 to see if the data reveal any useful additional information.

- ▶ Double-click the tree in the Viewer to open the Tree Editor.
- ▶ Click node 9 to select it. (If you want to select multiple nodes, use Ctrl-click).
- ▶ From the Tree Editor menus choose:
 - Rules
 - Filter Cases...

Figure 4-16
Filter Cases dialog box



The Filter Cases dialog box will create a filter variable and apply a filter setting based on the values of that variable. The default filter variable name is *filter_\$*.

- Cases from the selected nodes will receive a value of 1 for the filter variable.
- All other cases will receive a value of 0 and will be excluded from subsequent analyses until you change the filter status.

In this example, that means cases that aren't in node 9 will be filtered out (but not deleted) for now.

- ▶ Click OK to create the filter variable and apply the filter condition.

Figure 4-17
Filtered cases in Data Editor

	Income	Credit_cards	Education	Car_loans	NodeID	P
1	2.00	2.00	2.00	2.00	9	
2	2.00	2.00	2.00	2.00	8	
3	1.00	2.00	1.00	2.00	1	
4	1.00	2.00	2.00	1.00	1	
5	2.00	2.00	2.00	2.00	9	
6	2.00	2.00	2.00	2.00	9	
7	2.00	2.00	2.00	2.00	9	
8	1.00	2.00	1.00	2.00	1	
9	1.00	2.00	1.00	2.00	1	
10	2.00	2.00	2.00	2.00	8	

In the Data Editor, cases that have been filtered out are indicated with a diagonal slash through the row number. Cases that are not in node 9 are filtered out. Cases in node 9 are not filtered; so subsequent analyses will include only cases from node 9.

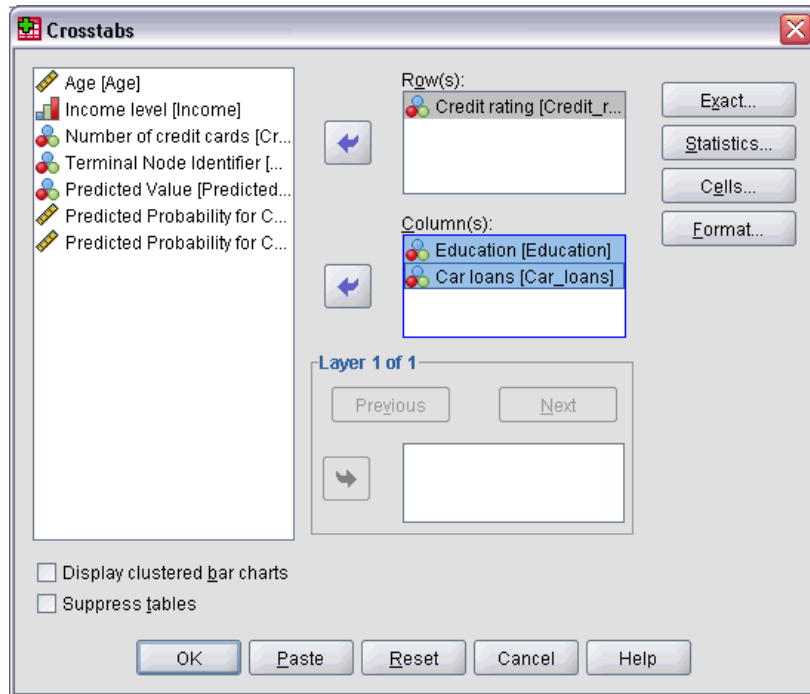
Examining the Selected Cases

As a first step in examining the cases in node 9, you might want to look at the variables not used in the model. In this example, all variables in the data file were included in the analysis, but two of them were not included in the final model: *education* and *car loans*. Since there's probably a good reason why the procedure omitted them from the final model, they probably won't tell us much, but let's take a look anyway.

- ▶ From the menus choose:

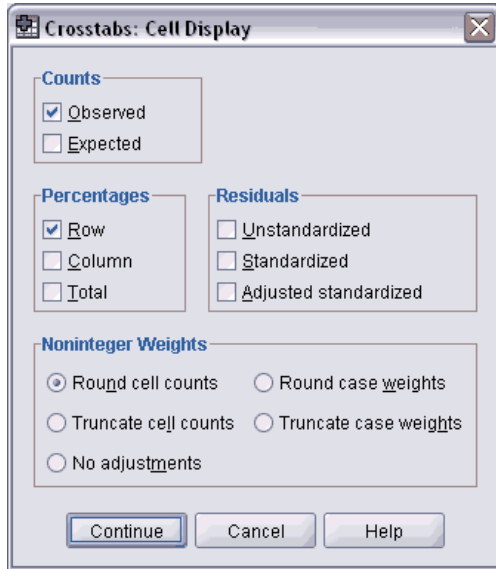
- Analyze
 - Descriptive Statistics
 - Crosstabs...

Figure 4-18
Crosstabs dialog box



- ▶ Select *Credit rating* for the row variable.
- ▶ Select *Education* and *Car loans* for the column variables.
- ▶ Click Cells.

Figure 4-19
Crosstabs Cell Display dialog box



- ▶ In the Percentages group, select (check) Row.
- ▶ Then click Continue, and in the main Crosstabs dialog box, click OK to run the procedure.

Examining the crosstabulations, you can see that for the two variables not included in the model, there isn't a great deal of difference between cases in the good and bad credit rating categories.

Figure 4-20
Crosstabulations for cases in selected node

Credit rating * Education Crosstabulation

			Education		Total
			High school	College	
Credit rating	Bad	Count	110	101	211
		% within Credit rating	52.1%	47.9%	100.0%
	Good	Count	128	144	272
		% within Credit rating	47.1%	52.9%	100.0%
Total		Count	238	245	483
		% within Credit rating	49.3%	50.7%	100.0%

Credit rating * Car loans Crosstabulation

			Car loans		Total
			None or 1	2 or More	
Credit rating	Bad	Count	18	193	211
		% within Credit rating	8.5%	91.5%	100.0%
	Good	Count	39	233	272
		% within Credit rating	14.3%	85.7%	100.0%
Total		Count	57	426	483
		% within Credit rating	11.8%	88.2%	100.0%

- For *education*, slightly more than half of the cases with a bad credit rating have only a high school education, while slightly more than half with a good credit rating have a college education—but this difference is not statistically significant.
- For *car loans*, the percentage of good credit cases with only one or no car loans is higher than the corresponding percentage for bad credit cases, but the vast majority of cases in both groups has two or more car loans.

So, although you now get some idea of why these variables were not included in the final model, you unfortunately haven't gained any insight into how to get better prediction for node 9. If there were other variables not specified for the analysis, you might want to examine some of them before proceeding.

Assigning Costs to Outcomes

As noted earlier, aside from the fact that almost half of the cases in node 9 fall in each credit rating category, the fact that the predicted category is “good” is problematic if your main objective is to build a model that correctly identifies bad credit risks. Although you may not be able to improve the performance of node 9, you can still refine the model to improve the rate of correct classification for bad credit rating cases—although this will also result in a higher rate of misclassification for good credit rating cases.

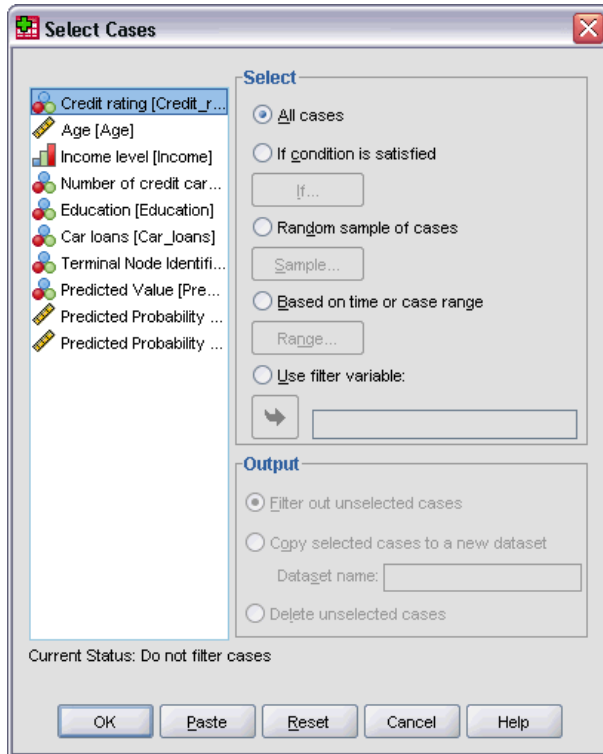
First, you need to turn off case filtering so that all cases will be used in the analysis again.

- ▶ From the menus choose:

- Data
 - Select Cases...

- ▶ In the Select Cases dialog box, select All cases, and then click OK.

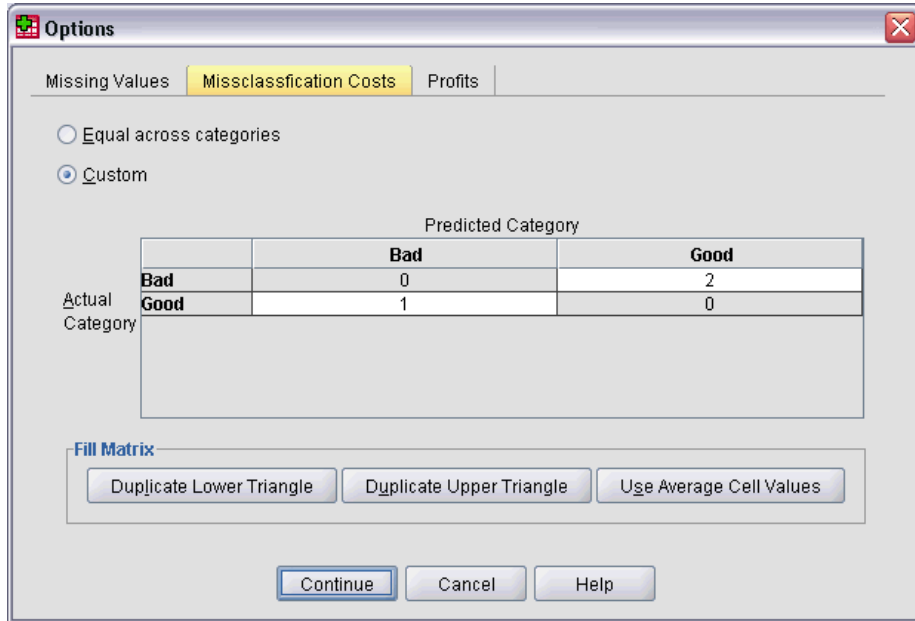
Figure 4-21
Select Cases dialog box



- ▶ Open the Decision Tree dialog box again, and click Options.

- ▶ Click the Misclassification Costs tab.

Figure 4-22
Options dialog box, Misclassification Costs tab

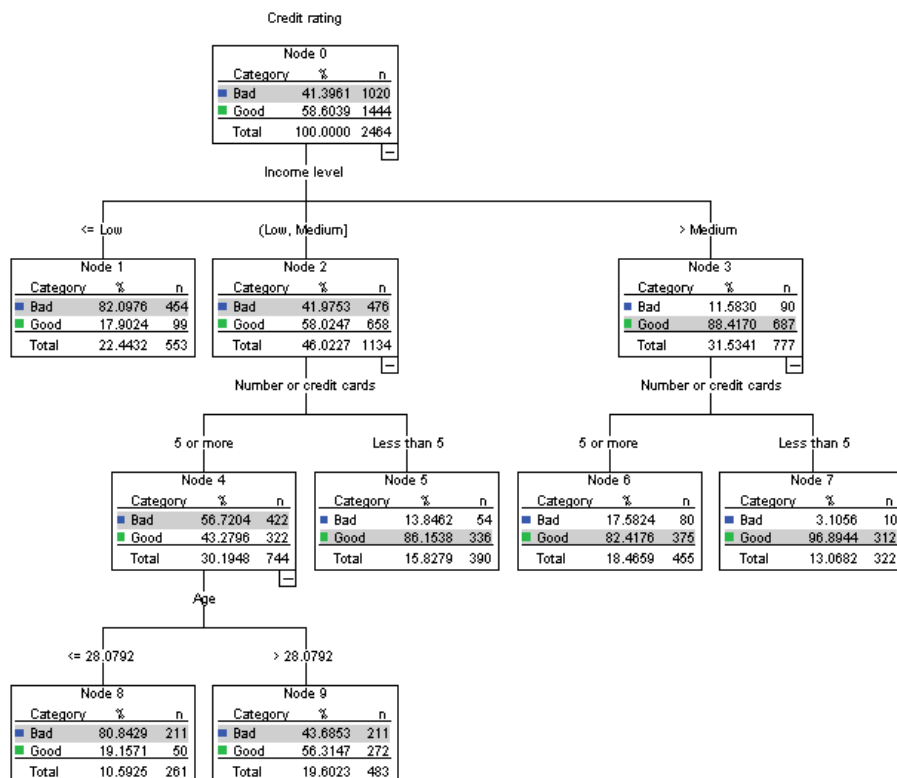


- ▶ Select Custom, and for *Bad* Actual Category / *Good* Predicted Category, enter a value of 2.

This tells the procedure that the “cost” of incorrectly classifying a bad credit risk as good is twice as high as the “cost” of incorrectly classifying a good credit risk as bad.

- ▶ Click Continue, and then in the main dialog box, click OK to run the procedure.

Figure 4-23
Tree model with adjusted cost values



At first glance, the tree generated by the procedure looks essentially the same as the original tree. Closer inspection, however, reveals that although the distribution of cases in each node hasn't changed, some predicted categories have changed.

For the terminal nodes, the predicted category remains the same in all nodes except one: node 9. The predicted category is now *Bad* even though slightly more than half of the cases are in the *Good* category.

Since we told the procedure that there was a higher cost for misclassifying bad credit risks as good, any node where the cases are fairly evenly distributed between the two categories now has a predicted category of *Bad* even if a slight majority of cases is in the *Good* category.

This change in predicted category is reflected in the classification table.

Figure 4-24
Risk and classification tables based on adjusted costs

Risk

Estimate	Std. Error
.288	.011

Growing Method: CHAID
Dependent Variable: Credit rating

Classification

Observed	Predicted		
	Bad	Good	Percent Correct
Bad	876	144	85.9%
Good	421	1023	70.8%
Overall Percentage	52.6%	47.4%	77.1%

Growing Method: CHAID
Dependent Variable: Credit rating

- Almost 86% of the bad credit risks are now correctly classified, compared to only 65% before.
- On the other hand, correct classification of good credit risks has decreased from 90% to 71%, and overall correct classification has decreased from 79.5% to 77.1%.

Note also that the risk estimate and the overall correct classification rate are no longer consistent with each other. You would expect a risk estimate of 0.229 if the overall correct classification rate is 77.1%. Increasing the cost of misclassification for bad credit cases has, in this example, inflated the risk value, making its interpretation less straightforward.

Summary

You can use tree models to classify cases into groups identified by certain characteristics, such as the characteristics associated with bank customers with good and bad credit records. If a particular predicted outcome is more important than other possible outcomes, you can refine the model to associate a higher misclassification cost for that outcome—but reducing misclassification rates for one outcome will increase misclassification rates for other outcomes.

Building a Scoring Model

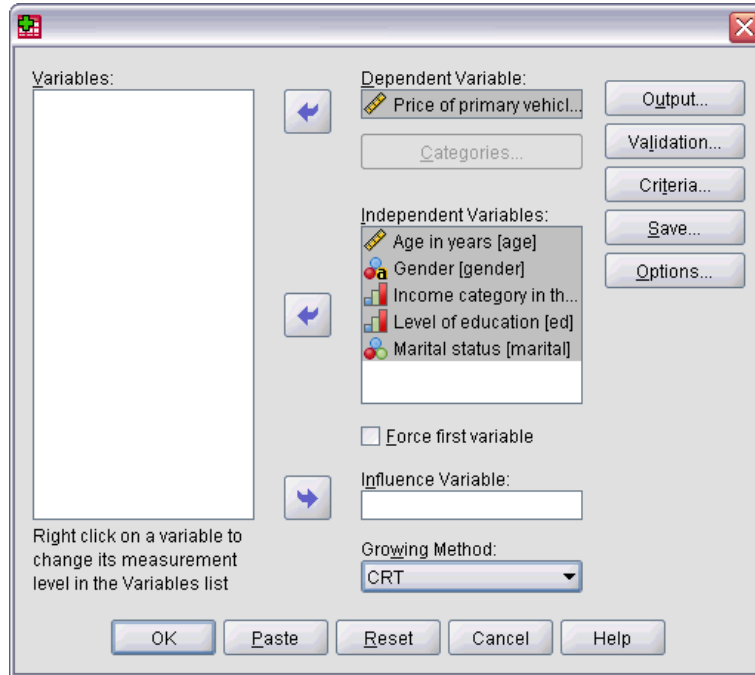
One of the most powerful and useful features of the Decision Tree procedure is the ability to build models that can then be applied to other data files to predict outcomes. For example, based on a data file that contains both demographic information and information on vehicle purchase price, we can build a model that can be used to predict how much people with similar demographic characteristics are likely to spend on a new car—and then apply that model to other data files where demographic information is available but information on previous vehicle purchasing is not.

For this example, we'll use the data file *tree_car.sav*. For more information, see [Sample Files](#) in Appendix A on p. 120.

Building the Model

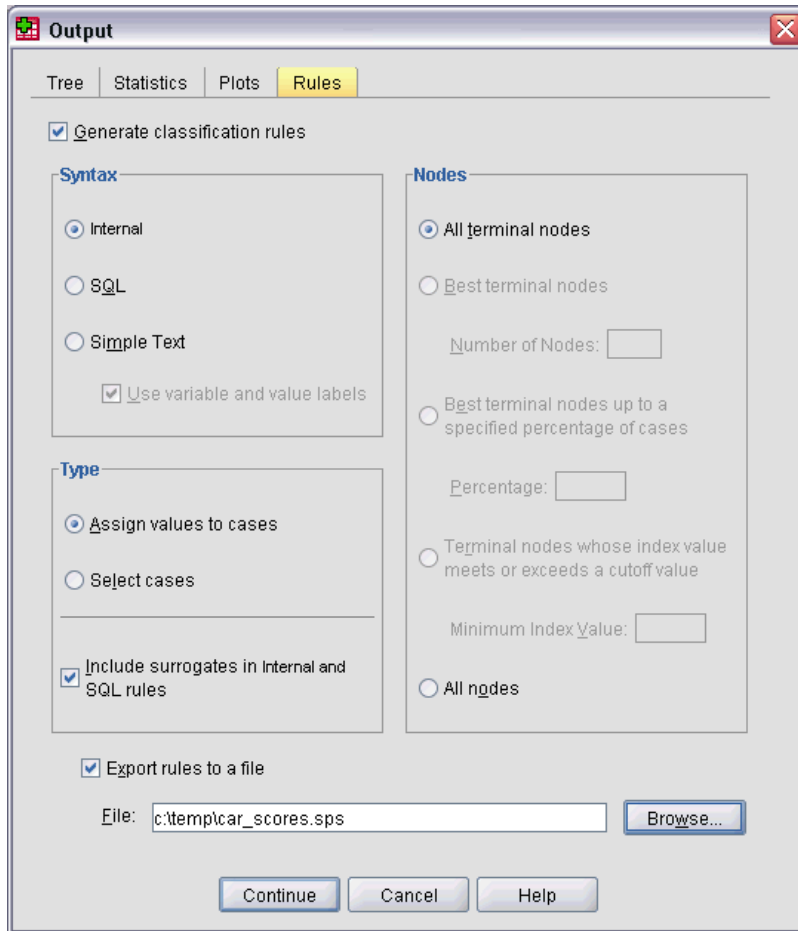
- ▶ To run a Decision Tree analysis, from the menus choose:
 - Analyze
 - Classify
 - Tree...

Figure 5-1
Decision Tree dialog box



- ▶ Select *Price of primary vehicle* as the dependent variable.
- ▶ Select all the remaining variables as independent variables. (The procedure will automatically exclude any variables that don't make a significant contribution to the final model.)
- ▶ For the growing method, select CRT.
- ▶ Click Output.

Figure 5-2
Output dialog box, Rules tab



- ▶ Click the Rules tab.
- ▶ Select (check) Generate classification rules.
- ▶ For Syntax, select Internal.
- ▶ For Type, select Assign values to cases.
- ▶ Select (check) Export rules to a file and enter a filename and directory location.

Remember the filename and location or write it down because you'll need it a little later. If you don't include a directory path, you may not know where the file has been saved. You can use the Browse button to navigate to a specific (and valid) directory location.

- ▶ Click Continue, and then click OK to run the procedure and build the tree model.

Evaluating the Model

Before applying the model to other data files, you probably want to make sure that the model works reasonably well with the original data used to build it.

Model Summary

Figure 5-3
Model summary table

Specifications	Growing Method	CRT
	Dependent Variable	Price of primary vehicle
	Independent Variables	Age in years , Gender , Income category in thousands , Level of education , Marital status
	Validation	NONE
	Maximum Tree Depth	5
	Minimum Cases in Parent Node	100
	Minimum Cases in Child Node	50
Results	Independent Variables Included	Income category in thousands , Age in years , Level of education
	Number of Nodes	29
	Number of Terminal Nodes	15
	Depth	5

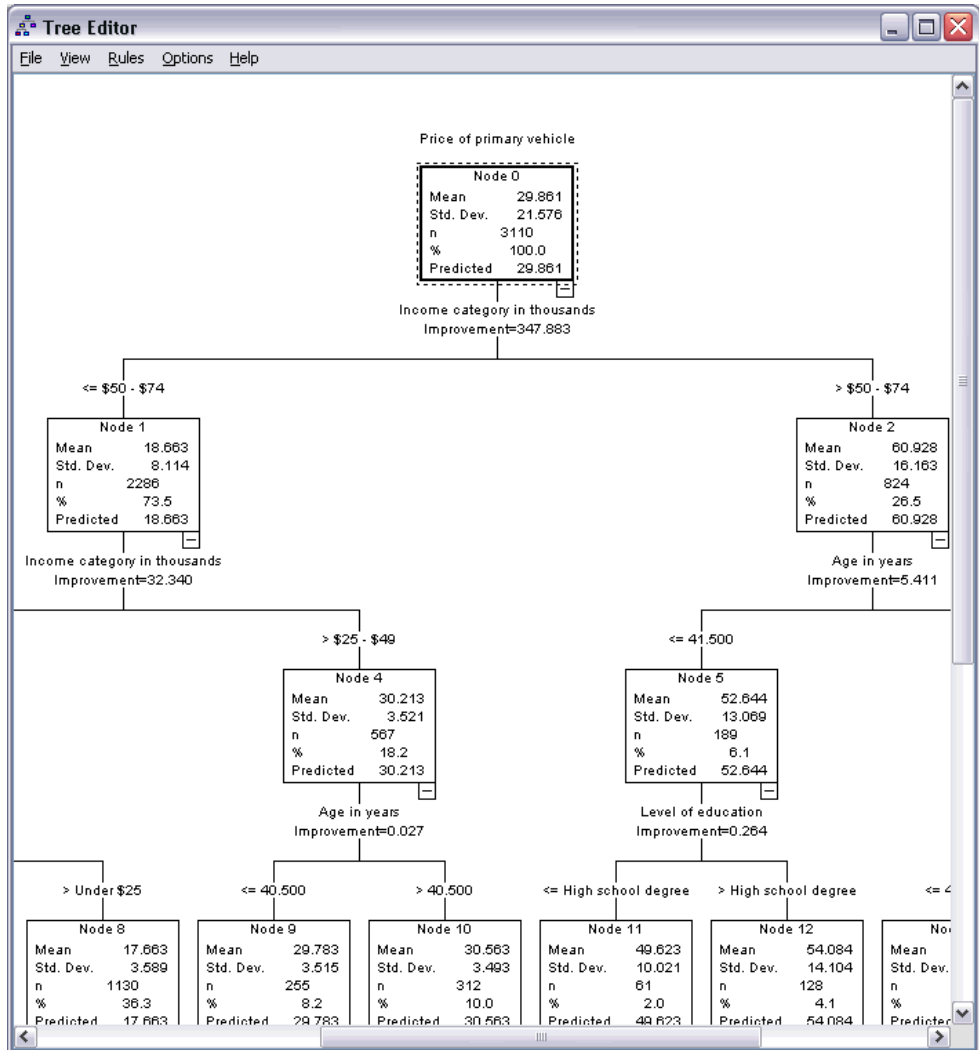
The model summary table indicates that only three of the selected independent variables made a significant enough contribution to be included in the final model: *income*, *age*, and *education*. This is important information to know if you want to apply this model to other data files, since the independent variables used in the model must be present in any data file to which you want to apply the model.

The summary table also indicates that the tree model itself is probably not a particularly simple one since it has 29 nodes and 15 terminal nodes. This may not be an issue if you want a reliable model that can be applied in a practical fashion rather than a simple model that is easy to describe or explain. Of course, for practical purposes, you probably also want a model that doesn't rely on too many independent

(predictor) variables. In this case, that's not a problem since only three independent variables are included in the final model.

Tree Model Diagram

Figure 5-4
Tree model diagram in Tree Editor



The tree model diagram has so many nodes that it may be difficult to see the whole model at once at a size where you can still read the node content information. You can use the tree map to see the entire tree:

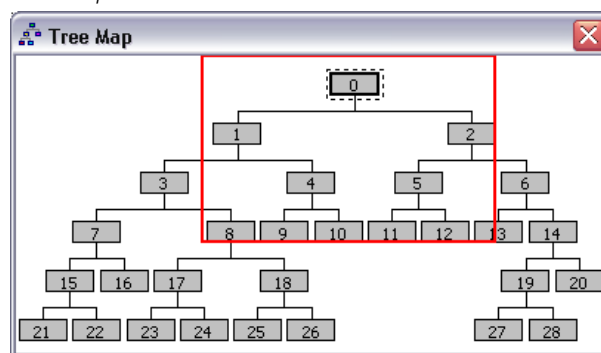
► Double-click the tree in the Viewer to open the Tree Editor.

► From the Tree Editor menus choose:

View
Tree Map

Figure 5-5

Tree map



- The tree map shows the entire tree. You can change the size of the tree map window, and it will grow or shrink the map display of the tree to fit the window size.
- The highlighted area in the tree map is the area of the tree currently displayed in the Tree Editor.
- You can use the tree map to navigate the tree and select nodes.

For more information, see [Tree Map](#) in Chapter 2 on p. 50.

For scale dependent variables, each node shows the mean and standard deviation of the dependent variable. Node 0 displays an overall mean vehicle purchase price of about 29.9 (in thousands), with a standard deviation of about 21.6.

- Node 1, which represents cases with an income of less than 75 (also in thousands) has a mean vehicle price of only 18.7.
- In contrast, node 2, which represents cases with an income of 75 or more, has a mean vehicle price of 60.9.

Further investigation of the tree would show that *age* and *education* also display a relationship with vehicle purchase price, but right now we're primarily interested in the practical application of the model rather than a detailed examination of its components.

Risk Estimate

Figure 5-6
Risk table

Risk	
Estimate	Std. Error
68.485	2.985

Growing Method: CRT
Dependent Variable: Price of primary vehicle

None of the results we've examined so far tell us if this is a particularly good model. One indicator of the model's performance is the risk estimate. For a scale dependent variable, the risk estimate is a measure of the within-node variance, which by itself may not tell you a great deal. A lower variance indicates a better model, but the variance is relative to the unit of measurement. If, for example, price was recorded in ones instead of thousands, the risk estimate would be a thousand times larger.

To provide a meaningful interpretation for the risk estimate with a scale dependent variable requires a little work:

- Total variance equals the within-node (error) variance plus the between-node (explained) variance.
- The within-node variance is the risk estimate value: 68.485.
- The total variance is the variance for the dependent variables before consideration of any independent variables, which is the variance at the root node.
- The standard deviation displayed at the root node is 21.576; so the total variance is that value squared: 465.524.
- The proportion of variance due to error (unexplained variance) is $68.485 / 465.524 = 0.147$.
- The proportion of variance explained by the model is $1 - 0.147 = 0.853$, or 85.3%, which indicates that this is a fairly good model. (This has a similar interpretation to the overall correct classification rate for a categorical dependent variable.)

Applying the Model to Another Data File

Having determined that the model is reasonably good, we can now apply that model to other data files containing similar *age*, *income*, and *education* variables and generate a new variable that represents the predicted vehicle purchase price for each case in that file. This process is often referred to as **scoring**.

When we generated the model, we specified that “rules” for assigning values to cases should be saved in a text file—in the form of command syntax. We will now use the commands in that file to generate scores in another data file.

- ▶ Open the data file *tree_score_car.sav*. For more information, see [Sample Files](#) in Appendix A on p. 120.

- ▶ Next, from the menus choose:

- File
 - New
 - Syntax

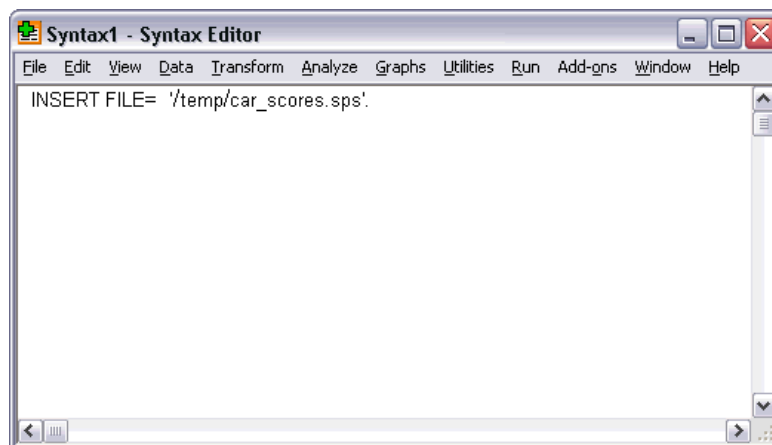
- ▶ In the command syntax window, type:

```
INSERT FILE=  
'/temp/car_scores.sps'.
```

If you used a different filename or location, make the appropriate changes.

Figure 5-7

Syntax window with INSERT command to run a command file



The INSERT command will run the commands in the specified file, which is the “rules” file that was generated when we created the model.

- ▶ From the command syntax window menus choose:
Run
All

Figure 5-8
Predicted values added to data file

	inccat	ed	marital	nod_001	pre_001
1	3.00	1	1	10.00	30.56
2	4.00	1	0	27.00	61.08
3	2.00	3	1	24.00	17.13
4	2.00	4	1	23.00	15.58
5	1.00	2	0	21.00	9.39
6	3.00	2	0	9.00	29.78
7	1.00	1	0	22.00	10.22
8	4.00	3	1	12.00	54.08
9	3.00	3	1	10.00	30.56
10	4.00	4	1	20.00	66.79

This adds two new variables to the data file:

- *nod_001* contains the terminal node number predicted by the model for each case.
- *pre_001* contains the predicted value for vehicle purchase price for each case.

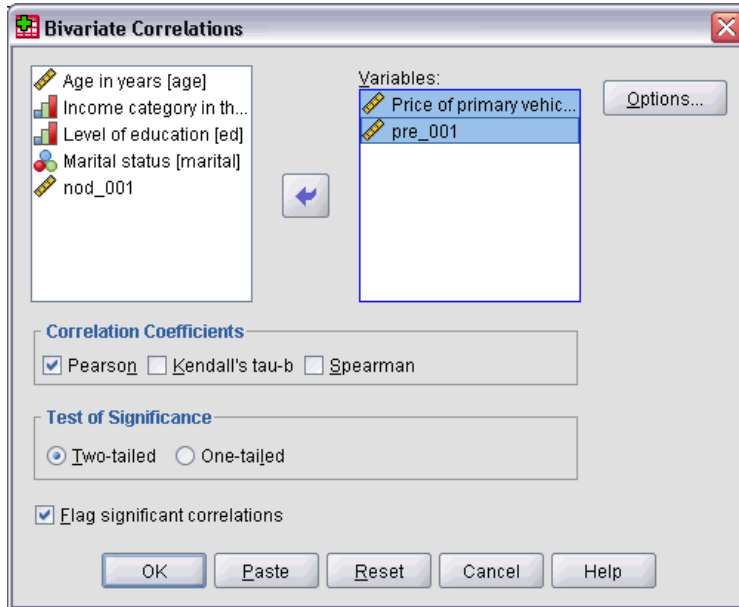
Since we requested rules for assigning values for terminal nodes, the number of possible predicted values is the same as the number of terminal nodes, which in this case is 15. For example, every case with a predicted node number of 10 will have the same predicted vehicle purchase price: 30.56. This is, not coincidentally, the mean value reported for terminal node 10 in the original model.

Although you would typically apply the model to data for which the value of the dependent variable is not known, in this example the data file to which we applied the model actually contains that information—and you can compare the model predictions to the actual values.

- ▶ From the menus choose:
 - Analyze
 - Correlate
 - Bivariate...

- ▶ Select *Price of primary vehicle* and *pre_001*.

Figure 5-9
Bivariate Correlations dialog box



- ▶ Click OK to run the procedure.

Figure 5-10
Correlation of actual and predicted vehicle price

		Price of primary vehicle	pre_001
Price of primary vehicle	Pearson Correlation	1	.919**
	Sig. (2-tailed)		.000
	N	3290	3290
pre_001	Pearson Correlation	.919**	1
	Sig. (2-tailed)		.000
	N	3290	3290

** . Correlation is significant at the 0.01 level (2-tailed).

The correlation of 0.92 indicates a very high positive correlation between actual and predicted vehicle price, which indicates that the model works well.

Summary

You can use the Decision Tree procedure to build models that can then be applied to other data files to predict outcomes. The target data file must contain variables with the same names as the independent variables included in the final model, measured in the same metric and with the same user-defined missing values (if any). However, neither the dependent variable nor independent variables excluded from the final model need to be present in the target data file.

Missing Values in Tree Models

The different growing methods deal with missing values for independent (predictor) variables in different ways:

- CHAID and Exhaustive CHAID treat all system- and user-missing values for each independent variable as a single category. For scale and ordinal independent variables, that category may or may not subsequently get merged with other categories of that independent variable, depending on the growing criteria.
- CRT and QUEST attempt to use **surrogates** for independent (predictor) variables. For cases in which the value for that variable is missing, other independent variables having high associations with the original variable are used for classification. These alternative predictors are called surrogates.

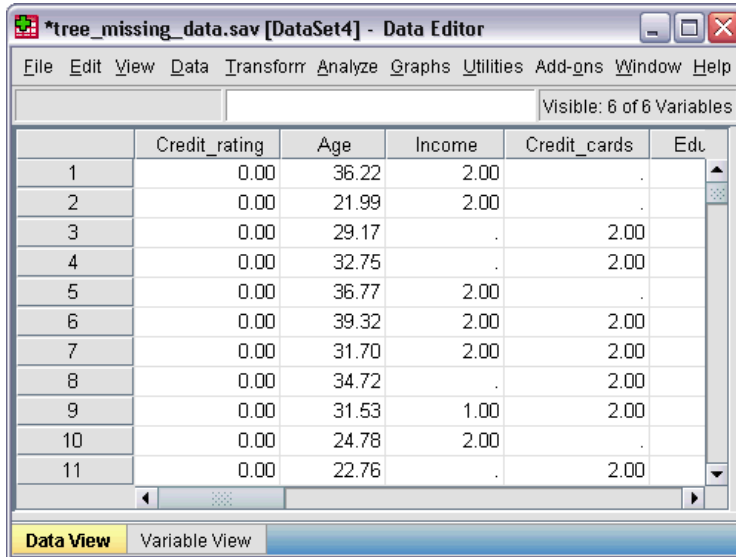
This example shows the difference between CHAID and CRT when there are missing values for independent variables used in the model.

For this example, we'll use the data file *tree_missing_data.sav*. For more information, see [Sample Files](#) in Appendix A on p. 120.

Note: For nominal independent variables and nominal dependent variables, you can choose to treat **user-missing** values as valid values, in which case those values are treated like any other nonmissing values. For more information, see [Missing Values](#) in Chapter 1 on p. 27.

Missing Values with CHAID

Figure 6-1
Credit data with missing values



The screenshot shows the SPSS Data Editor window for a file named *tree_missing_data.sav [DataSet4]. The window displays a table with 11 rows and 6 columns. The columns are labeled Credit_rating, Age, Income, Credit_cards, and Edu. The data shows various values, including missing values (represented by dots) for Credit_cards and Edu in several rows. The interface includes a menu bar (File, Edit, View, Data, Transform, Analyze, Graphs, Utilities, Add-ons, Window, Help) and a status bar at the bottom with tabs for Data View and Variable View.

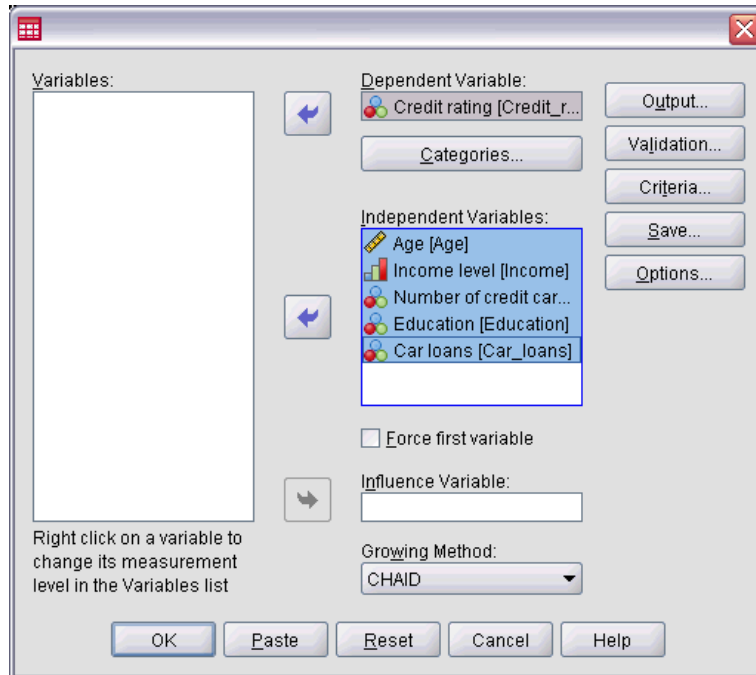
	Credit_rating	Age	Income	Credit_cards	Edu
1	0.00	36.22	2.00	.	.
2	0.00	21.99	2.00	.	.
3	0.00	29.17	.	2.00	.
4	0.00	32.75	.	2.00	.
5	0.00	36.77	2.00	.	.
6	0.00	39.32	2.00	2.00	.
7	0.00	31.70	2.00	2.00	.
8	0.00	34.72	.	2.00	.
9	0.00	31.53	1.00	2.00	.
10	0.00	24.78	2.00	.	.
11	0.00	22.76	.	2.00	.

Like the credit risk example (for more information, see [Chapter 4](#)), this example will try to build a model to classify good and bad credit risks. The main difference is that this data file contains missing values for some independent variables used in the model.

- ▶ To run a Decision Tree analysis, from the menus choose:

- Analyze
- Classify
- Tree...

Figure 6-2
Decision Tree dialog box

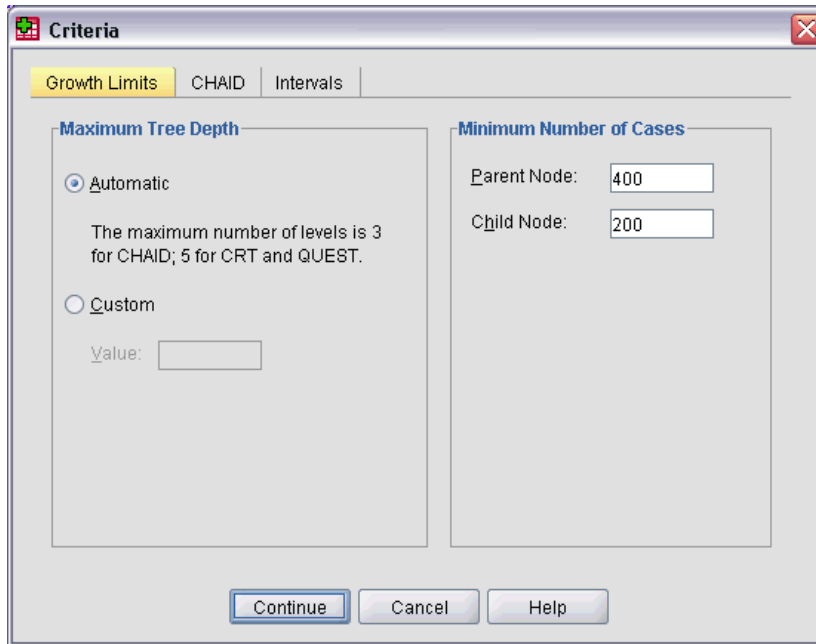


- ▶ Select *Credit rating* as the dependent variable.
- ▶ Select all of the remaining variables as independent variables. (The procedure will automatically exclude any variables that don't make a significant contribution to the final model.)
- ▶ For the growing method, select CHAID.

For this example, we want to keep the tree fairly simple; so, we'll limit the tree growth by raising the minimum number of cases for the parent and child nodes.

- ▶ In the main Decision Tree dialog box, click Criteria.

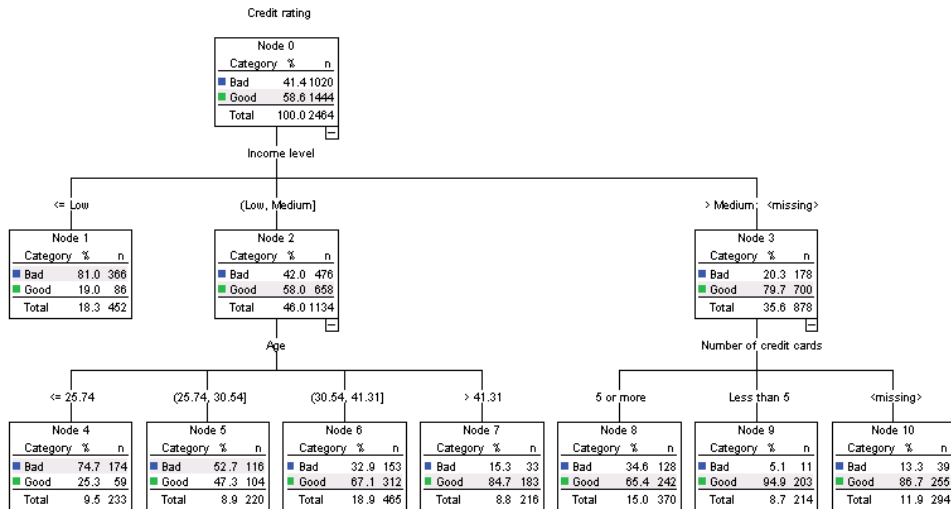
Figure 6-3
Criteria dialog box, Growth Limits tab



- ▶ For Minimum Number of Cases, type 400 for Parent Node and 200 for Child Node.
- ▶ Click Continue, and then click OK to run the procedure.

CHAID Results

Figure 6-4
CHAID tree with missing independent variable values



For node 3, the value of *income level* is displayed as *>Medium;<missing>*. This means that the node contains cases in the high-income category plus any cases with missing values for *income level*.

Terminal node 10 contains cases with missing values for *number of credit cards*. If you're interested in identifying good credit risks, this is actually the second best terminal node, which might be problematic if you want to use this model for predicting good credit risks. You probably wouldn't want a model that predicts a good credit rating simply because you don't know anything about how many credit cards a case has, and some of those cases may also be missing income-level information.

Figure 6-5
Risk and classification tables for CHAID model

Risk

Estimate	Std. Error
.249	.009

Growing Method: CHAID
 Dependent Variable: Credit rating

Classification

Observed	Predicted		
	Bad	Good	Percent Correct
Bad	656	364	64.3%
Good	249	1195	82.8%
Overall Percentage	36.7%	63.3%	75.1%

Growing Method: CHAID
 Dependent Variable: Credit rating

The risk and classification tables indicate that the CHAID model correctly classifies about 75% of the cases. This isn't bad, but it's not great. Furthermore, we may have reason to suspect that the correct classification rate for good credit cases may be overly optimistic, since it's partly based on the assumption that lack of information about two independent variables (*income level* and *number of credit cards*) is an indication of good credit.

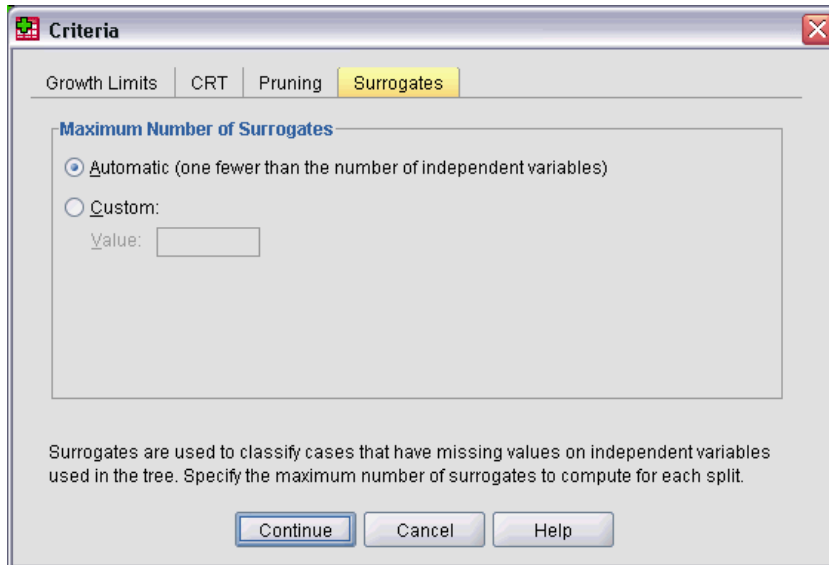
Missing Values with CRT

Now let's try the same basic analysis, except we'll use CRT as the growing method.

- ▶ In the main Decision Tree dialog box, for the growing method, select CRT.
- ▶ Click Criteria.
- ▶ Make sure that the minimum number of cases is still set at 400 for parent nodes and 200 for child nodes.
- ▶ Click the Surrogates tab.

Note: You will not see the Surrogates tab unless you have selected CRT or QUEST as the growing method.

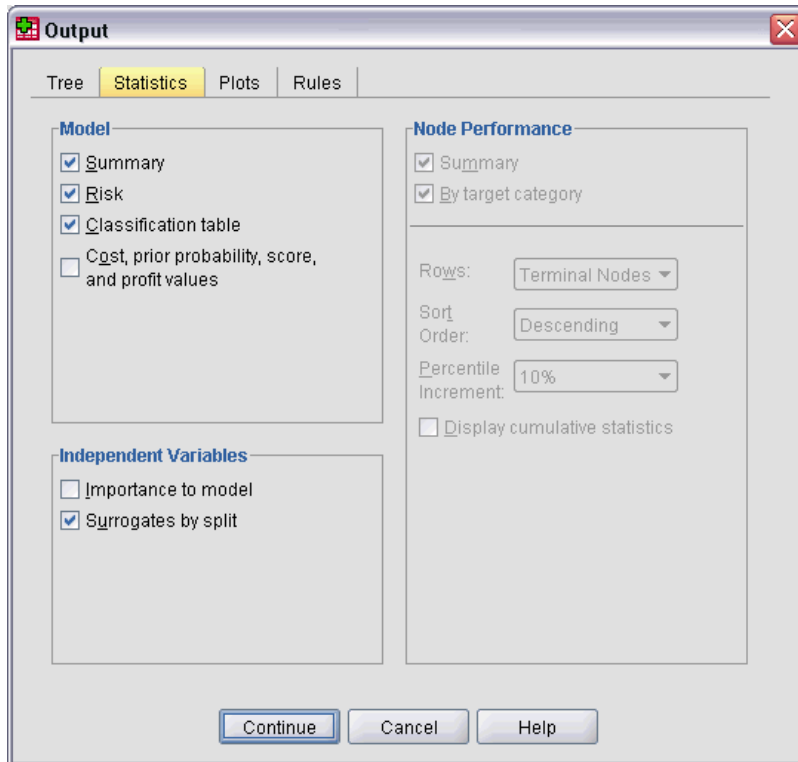
Figure 6-6
Criteria dialog box, Surrogates tab



For each independent variable node split, the Automatic setting will consider every other independent variable specified for the model as a possible surrogate. Since there aren't very many independent variables in this example, the Automatic setting is fine.

- ▶ Click Continue.
- ▶ In the main Decision Tree dialog box, click Output.

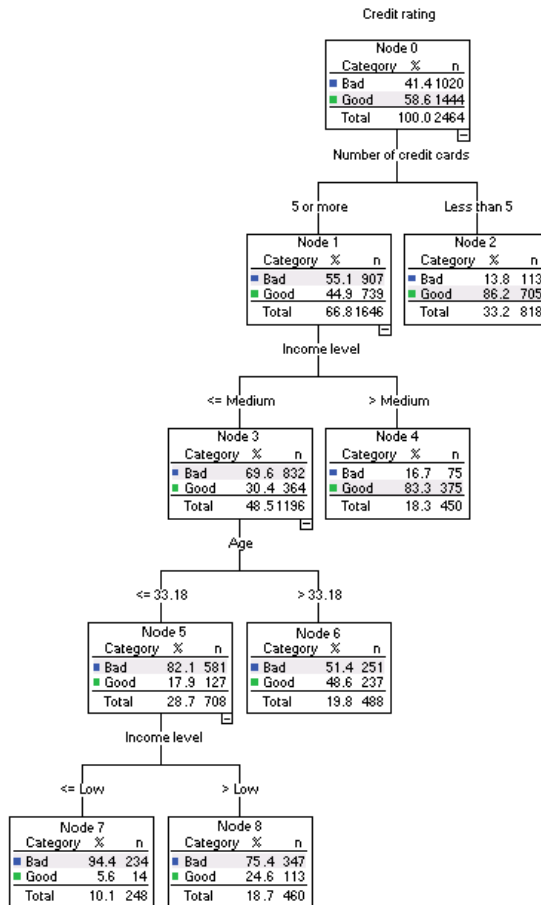
Figure 6-7
Output dialog box, Statistics tab



- ▶ Click the Statistics tab.
- ▶ Select Surrogates by split.
- ▶ Click Continue, and then click OK to run the procedure.

CRT Results

Figure 6-8
CRT tree with missing independent variable values



You may immediately notice that this tree doesn't look much like the CHAID tree. That, by itself, doesn't necessarily mean much. In a CRT tree model, all splits are binary; that is, each parent node is split into only two child nodes. In a CHAID model, parent nodes can be split into many child nodes. So, the trees will often look different even if they represent the same underlying model.

There are, however, a number of important differences:

- The most important independent (predictor) variable in the CRT model is *number of credit cards*, while in the CHAID model, the most important predictor was *income level*.
- For cases with less than five credit cards, *number of credit cards* is the only significant predictor of credit rating, and node 2 is a terminal node.
- As with the CHAID model, *income level* and *age* are also included in the model, although *income level* is now the second predictor rather than the first.
- There aren't any nodes that contain a <missing> category, because CRT uses surrogate predictors rather than missing values in the model.

Figure 6-9

Risk and classification tables for CRT model

Risk

Estimate	Std. Error
.224	.008

Growing Method: CRT
Dependent Variable: Credit rating

Classification

Observed	Predicted		
	Bad	Good	Percent Correct
Bad	832	188	81.6%
Good	364	1080	74.8%
Overall Percentage	48.5%	51.5%	77.6%

Growing Method: CRT
Dependent Variable: Credit rating

- The risk and classification tables show an overall correct classification rate of almost 78%, a slight increase over the CHAID model (75%).
- The correct classification rate for bad credit cases is much higher for the CRT model—81.6% compared to only 64.3% for the CHAID model.
- The correct classification rate for good credit cases, however, has declined from 82.8% with CHAID to 74.8% with CRT.

Surrogates

The differences between the CHAID and CRT models are due, in part, to the use of surrogates in the CRT model. The surrogates table indicates how surrogates were used in the model.

Figure 6-10
Surrogates table

Parent Node	Independent Variable	Improvement	Association
0	Primary Number of credit cards	.090	
	Surrogate Car loans	.052	.643
	Age	.001	.004
1	Primary Income level	.071	
	Surrogate Age	.001	.004
3	Primary Age	.022	
5	Primary Income level	.006	
	Surrogate Age	.000	.009

Growing Method: CRT
Dependent Variable: Credit_rating

- At the root node (node 0), the best independent (predictor) variable is *number of credit cards*.
- For any cases with missing values for *number of credit cards*, *car loans* is used as the surrogate predictor, since this variable has a fairly high association (0.643) with *number of credit cards*.
- If a case also has a missing value for *car loans*, then *age* is used as the surrogate (although it has a fairly low association value of only 0.004).
- *Age* is also used as a surrogate for *income level* at nodes 1 and 5.

Summary

Different growing methods handle missing data in different ways. If the data used to create the model contain many missing values—or if you want to apply that model to other data files that contain many missing values—you should evaluate the effect of missing values on the various models. If you want to use surrogates in the model to compensate for missing values, use the CRT or QUEST methods.

Sample Files

The sample files installed with the product can be found in the *Samples* subdirectory of the installation directory. There is a separate folder within the Samples subdirectory for each of the following languages: English, French, German, Italian, Japanese, Korean, Polish, Russian, Simplified Chinese, Spanish, and Traditional Chinese.

Not all sample files are available in all languages. If a sample file is not available in a language, that language folder contains an English version of the sample file.

Descriptions

Following are brief descriptions of the sample files used in various examples throughout the documentation.

- **accidents.sav.** This is a hypothetical data file that concerns an insurance company that is studying age and gender risk factors for automobile accidents in a given region. Each case corresponds to a cross-classification of age category and gender.
- **adl.sav.** This is a hypothetical data file that concerns efforts to determine the benefits of a proposed type of therapy for stroke patients. Physicians randomly assigned female stroke patients to one of two groups. The first received the standard physical therapy, and the second received an additional emotional therapy. Three months following the treatments, each patient's abilities to perform common activities of daily life were scored as ordinal variables.
- **advert.sav.** This is a hypothetical data file that concerns a retailer's efforts to examine the relationship between money spent on advertising and the resulting sales. To this end, they have collected past sales figures and the associated advertising costs..

- **aflatoxin.sav.** This is a hypothetical data file that concerns the testing of corn crops for aflatoxin, a poison whose concentration varies widely between and within crop yields. A grain processor has received 16 samples from each of 8 crop yields and measured the aflatoxin levels in parts per billion (PPB).
- **aflatoxin20.sav.** This data file contains the aflatoxin measurements from each of the 16 samples from yields 4 and 8 from the *aflatoxin.sav* data file.
- **anorectic.sav.** While working toward a standardized symptomatology of anorectic/bulimic behavior, researchers made a study of 55 adolescents with known eating disorders. Each patient was seen four times over four years, for a total of 220 observations. At each observation, the patients were scored for each of 16 symptoms. Symptom scores are missing for patient 71 at time 2, patient 76 at time 2, and patient 47 at time 3, leaving 217 valid observations.
- **autoaccidents.sav.** This is a hypothetical data file that concerns the efforts of an insurance analyst to model the number of automobile accidents per driver while also accounting for driver age and gender. Each case represents a separate driver and records the driver's gender, age in years, and number of automobile accidents in the last five years.
- **band.sav.** This data file contains hypothetical weekly sales figures of music CDs for a band. Data for three possible predictor variables are also included.
- **bankloan.sav.** This is a hypothetical data file that concerns a bank's efforts to reduce the rate of loan defaults. The file contains financial and demographic information on 850 past and prospective customers. The first 700 cases are customers who were previously given loans. The last 150 cases are prospective customers that the bank needs to classify as good or bad credit risks.
- **bankloan_binning.sav.** This is a hypothetical data file containing financial and demographic information on 5,000 past customers.
- **behavior.sav.** In a classic example, 52 students were asked to rate the combinations of 15 situations and 15 behaviors on a 10-point scale ranging from 0="extremely appropriate" to 9="extremely inappropriate." Averaged over individuals, the values are taken as dissimilarities.
- **behavior_ini.sav.** This data file contains an initial configuration for a two-dimensional solution for *behavior.sav*.
- **brakes.sav.** This is a hypothetical data file that concerns quality control at a factory that produces disc brakes for high-performance automobiles. The data file contains diameter measurements of 16 discs from each of 8 production machines. The target diameter for the brakes is 322 millimeters.

- **breakfast.sav.** In a classic study, 21 Wharton School MBA students and their spouses were asked to rank 15 breakfast items in order of preference with 1="most preferred" to 15="least preferred." Their preferences were recorded under six different scenarios, from "Overall preference" to "Snack, with beverage only."
- **breakfast-overall.sav.** This data file contains the breakfast item preferences for the first scenario, "Overall preference," only.
- **broadband_1.sav.** This is a hypothetical data file containing the number of subscribers, by region, to a national broadband service. The data file contains monthly subscriber numbers for 85 regions over a four-year period.
- **broadband_2.sav.** This data file is identical to *broadband_1.sav* but contains data for three additional months.
- **car_insurance_claims.sav.** A dataset presented and analyzed elsewhere concerns damage claims for cars. The average claim amount can be modeled as having a gamma distribution, using an inverse link function to relate the mean of the dependent variable to a linear combination of the policyholder age, vehicle type, and vehicle age. The number of claims filed can be used as a scaling weight.
- **car_sales.sav.** This data file contains hypothetical sales estimates, list prices, and physical specifications for various makes and models of vehicles. The list prices and physical specifications were obtained alternately from *edmunds.com* and manufacturer sites.
- **carpet.sav.** In a popular example, a company interested in marketing a new carpet cleaner wants to examine the influence of five factors on consumer preference—package design, brand name, price, a *Good Housekeeping* seal, and a money-back guarantee. There are three factor levels for package design, each one differing in the location of the applicator brush; three brand names (*K2R*, *Glory*, and *Bissell*); three price levels; and two levels (either no or yes) for each of the last two factors. Ten consumers rank 22 profiles defined by these factors. The variable *Preference* contains the rank of the average rankings for each profile. Low rankings correspond to high preference. This variable reflects an overall measure of preference for each profile.
- **carpet_prefs.sav.** This data file is based on the same example as described for *carpet.sav*, but it contains the actual rankings collected from each of the 10 consumers. The consumers were asked to rank the 22 product profiles from the most to the least preferred. The variables *PREF1* through *PREF22* contain the identifiers of the associated profiles, as defined in *carpet_plan.sav*.

-
- **catalog.sav.** This data file contains hypothetical monthly sales figures for three products sold by a catalog company. Data for five possible predictor variables are also included.
 - **catalog_seasfac.sav.** This data file is the same as *catalog.sav* except for the addition of a set of seasonal factors calculated from the Seasonal Decomposition procedure along with the accompanying date variables.
 - **cellular.sav.** This is a hypothetical data file that concerns a cellular phone company's efforts to reduce churn. Churn propensity scores are applied to accounts, ranging from 0 to 100. Accounts scoring 50 or above may be looking to change providers.
 - **ceramics.sav.** This is a hypothetical data file that concerns a manufacturer's efforts to determine whether a new premium alloy has a greater heat resistance than a standard alloy. Each case represents a separate test of one of the alloys; the heat at which the bearing failed is recorded.
 - **cereal.sav.** This is a hypothetical data file that concerns a poll of 880 people about their breakfast preferences, also noting their age, gender, marital status, and whether or not they have an active lifestyle (based on whether they exercise at least twice a week). Each case represents a separate respondent.
 - **clothing_defects.sav.** This is a hypothetical data file that concerns the quality control process at a clothing factory. From each lot produced at the factory, the inspectors take a sample of clothes and count the number of clothes that are unacceptable.
 - **coffee.sav.** This data file pertains to perceived images of six iced-coffee brands . For each of 23 iced-coffee image attributes, people selected all brands that were described by the attribute. The six brands are denoted AA, BB, CC, DD, EE, and FF to preserve confidentiality.
 - **contacts.sav.** This is a hypothetical data file that concerns the contact lists for a group of corporate computer sales representatives. Each contact is categorized by the department of the company in which they work and their company ranks. Also recorded are the amount of the last sale made, the time since the last sale, and the size of the contact's company.
 - **creditpromo.sav.** This is a hypothetical data file that concerns a department store's efforts to evaluate the effectiveness of a recent credit card promotion. To this end, 500 cardholders were randomly selected. Half received an ad promoting a reduced interest rate on purchases made over the next three months. Half received a standard seasonal ad.

- **customer_dbase.sav.** This is a hypothetical data file that concerns a company's efforts to use the information in its data warehouse to make special offers to customers who are most likely to reply. A subset of the customer base was selected at random and given the special offers, and their responses were recorded.
- **customer_information.sav.** A hypothetical data file containing customer mailing information, such as name and address.
- **customers_model.sav.** This file contains hypothetical data on individuals targeted by a marketing campaign. These data include demographic information, a summary of purchasing history, and whether or not each individual responded to the campaign. Each case represents a separate individual.
- **customers_new.sav.** This file contains hypothetical data on individuals who are potential candidates for a marketing campaign. These data include demographic information and a summary of purchasing history for each individual. Each case represents a separate individual.
- **debate.sav.** This is a hypothetical data file that concerns paired responses to a survey from attendees of a political debate before and after the debate. Each case corresponds to a separate respondent.
- **debate_aggregate.sav.** This is a hypothetical data file that aggregates the responses in *debate.sav*. Each case corresponds to a cross-classification of preference before and after the debate.
- **demo.sav.** This is a hypothetical data file that concerns a purchased customer database, for the purpose of mailing monthly offers. Whether or not the customer responded to the offer is recorded, along with various demographic information.
- **demo_cs_1.sav.** This is a hypothetical data file that concerns the first step of a company's efforts to compile a database of survey information. Each case corresponds to a different city, and the region, province, district, and city identification are recorded.
- **demo_cs_2.sav.** This is a hypothetical data file that concerns the second step of a company's efforts to compile a database of survey information. Each case corresponds to a different household unit from cities selected in the first step, and the region, province, district, city, subdivision, and unit identification are recorded. The sampling information from the first two stages of the design is also included.
- **demo_cs.sav.** This is a hypothetical data file that contains survey information collected using a complex sampling design. Each case corresponds to a different household unit, and various demographic and sampling information is recorded.

- **dietstudy.sav.** This hypothetical data file contains the results of a study of the “Stillman diet” . Each case corresponds to a separate subject and records his or her pre- and post-diet weights in pounds and triglyceride levels in mg/100 ml.
- **dischargedata.sav.** This is a data file concerning *Seasonal Patterns of Winnipeg Hospital Use*, from the Manitoba Centre for Health Policy.
- **dvdplayer.sav.** This is a hypothetical data file that concerns the development of a new DVD player. Using a prototype, the marketing team has collected focus group data. Each case corresponds to a separate surveyed user and records some demographic information about them and their responses to questions about the prototype.
- **flying.sav.** This data file contains the flying mileages between 10 American cities.
- **german_credit.sav.** This data file is taken from the “German credit” dataset in the Repository of Machine Learning Databases at the University of California, Irvine.
- **grocery_1month.sav.** This hypothetical data file is the *grocery_coupons.sav* data file with the weekly purchases “rolled-up” so that each case corresponds to a separate customer. Some of the variables that changed weekly disappear as a result, and the amount spent recorded is now the sum of the amounts spent during the four weeks of the study.
- **grocery_coupons.sav.** This is a hypothetical data file that contains survey data collected by a grocery store chain interested in the purchasing habits of their customers. Each customer is followed for four weeks, and each case corresponds to a separate customer-week and records information about where and how the customer shops, including how much was spent on groceries during that week.
- **guttman.sav.** Bell presented a table to illustrate possible social groups. Guttman used a portion of this table, in which five variables describing such things as social interaction, feelings of belonging to a group, physical proximity of members, and formality of the relationship were crossed with seven theoretical social groups, including crowds (for example, people at a football game), audiences (for example, people at a theater or classroom lecture), public (for example, newspaper or television audiences), mobs (like a crowd but with much more intense interaction), primary groups (intimate), secondary groups (voluntary), and the modern community (loose confederation resulting from close physical proximity and a need for specialized services).
- **healthplans.sav.** This is a hypothetical data file that concerns an insurance group’s efforts to evaluate four different health care plans for small employers. Twelve employers are recruited to rank the plans by how much they would prefer to

offer them to their employees. Each case corresponds to a separate employer and records the reactions to each plan.

- **health_funding.sav.** This is a hypothetical data file that contains data on health care funding (amount per 100 population), disease rates (rate per 10,000 population), and visits to health care providers (rate per 10,000 population). Each case represents a different city.
- **hivassay.sav.** This is a hypothetical data file that concerns the efforts of a pharmaceutical lab to develop a rapid assay for detecting HIV infection. The results of the assay are eight deepening shades of red, with deeper shades indicating greater likelihood of infection. A laboratory trial was conducted on 2,000 blood samples, half of which were infected with HIV and half of which were clean.
- **hourlywagedata.sav.** This is a hypothetical data file that concerns the hourly wages of nurses from office and hospital positions and with varying levels of experience.
- **insure.sav.** This is a hypothetical data file that concerns an insurance company that is studying the risk factors that indicate whether a client will have to make a claim on a 10-year term life insurance contract. Each case in the data file represents a pair of contracts, one of which recorded a claim and the other didn't, matched on age and gender.
- **judges.sav.** This is a hypothetical data file that concerns the scores given by trained judges (plus one enthusiast) to 300 gymnastics performances. Each row represents a separate performance; the judges viewed the same performances.
- **kinship_dat.sav.** Rosenberg and Kim set out to analyze 15 kinship terms (aunt, brother, cousin, daughter, father, granddaughter, grandfather, grandmother, grandson, mother, nephew, niece, sister, son, uncle). They asked four groups of college students (two female, two male) to sort these terms on the basis of similarities. Two groups (one female, one male) were asked to sort twice, with the second sorting based on a different criterion from the first sort. Thus, a total of six "sources" were obtained. Each source corresponds to a 15×15 proximity matrix, whose cells are equal to the number of people in a source minus the number of times the objects were partitioned together in that source.
- **kinship_ini.sav.** This data file contains an initial configuration for a three-dimensional solution for *kinship_dat.sav*.
- **kinship_var.sav.** This data file contains independent variables *gender*, *gener(ation)*, and *degree* (of separation) that can be used to interpret the dimensions of a solution for *kinship_dat.sav*. Specifically, they can be used to restrict the space of the solution to a linear combination of these variables.

-
- **mailresponse.sav.** This is a hypothetical data file that concerns the efforts of a clothing manufacturer to determine whether using first class postage for direct mailings results in faster responses than bulk mail. Order-takers record how many weeks after the mailing each order is taken.
 - **marketvalues.sav.** This data file concerns home sales in a new housing development in Algonquin, Ill., during the years from 1999–2000. These sales are a matter of public record.
 - **mutualfund.sav.** This data file concerns stock market information for various tech stocks listed on the S&P 500. Each case corresponds to a separate company.
 - **nhis2000_subset.sav.** The National Health Interview Survey (NHIS) is a large, population-based survey of the U.S. civilian population. Interviews are carried out face-to-face in a nationally representative sample of households. Demographic information and observations about health behaviors and status are obtained for members of each household. This data file contains a subset of information from the 2000 survey. National Center for Health Statistics. National Health Interview Survey, 2000. Public-use data file and documentation. ftp://ftp.cdc.gov/pub/Health_Statistics/NCHS/Datasets/NHIS/2000/. Accessed 2003.
 - **ozone.sav.** The data include 330 observations on six meteorological variables for predicting ozone concentration from the remaining variables. Previous researchers , , among others found nonlinearities among these variables, which hinder standard regression approaches.
 - **pain_medication.sav.** This hypothetical data file contains the results of a clinical trial for anti-inflammatory medication for treating chronic arthritic pain. Of particular interest is the time it takes for the drug to take effect and how it compares to an existing medication.
 - **patient_los.sav.** This hypothetical data file contains the treatment records of patients who were admitted to the hospital for suspected myocardial infarction (MI, or “heart attack”). Each case corresponds to a separate patient and records many variables related to their hospital stay.
 - **patlos_sample.sav.** This hypothetical data file contains the treatment records of a sample of patients who received thrombolytics during treatment for myocardial infarction (MI, or “heart attack”). Each case corresponds to a separate patient and records many variables related to their hospital stay.

- **polishing.sav.** This is the “Nambeware Polishing Times” data file from the Data and Story Library. It concerns the efforts of a metal tableware manufacturer (Nambe Mills, Santa Fe, N. M.) to plan its production schedule. Each case represents a different item in the product line. The diameter, polishing time, price, and product type are recorded for each item.
- **poll_cs.sav.** This is a hypothetical data file that concerns pollsters’ efforts to determine the level of public support for a bill before the legislature. The cases correspond to registered voters. Each case records the county, township, and neighborhood in which the voter lives.
- **poll_cs_sample.sav.** This hypothetical data file contains a sample of the voters listed in *poll_cs.sav*. The sample was taken according to the design specified in the *poll.csplan* plan file, and this data file records the inclusion probabilities and sample weights. Note, however, that because the sampling plan makes use of a probability-proportional-to-size (PPS) method, there is also a file containing the joint selection probabilities (*poll_jointprob.sav*). The additional variables corresponding to voter demographics and their opinion on the proposed bill were collected and added to the data file after the sample was taken.
- **property_assess.sav.** This is a hypothetical data file that concerns a county assessor’s efforts to keep property value assessments up to date on limited resources. The cases correspond to properties sold in the county in the past year. Each case in the data file records the township in which the property lies, the assessor who last visited the property, the time since that assessment, the valuation made at that time, and the sale value of the property.
- **property_assess_cs.sav.** This is a hypothetical data file that concerns a state assessor’s efforts to keep property value assessments up to date on limited resources. The cases correspond to properties in the state. Each case in the data file records the county, township, and neighborhood in which the property lies, the time since the last assessment, and the valuation made at that time.
- **property_assess_cs_sample.sav.** This hypothetical data file contains a sample of the properties listed in *property_assess_cs.sav*. The sample was taken according to the design specified in the *property_assess.csplan* plan file, and this data file records the inclusion probabilities and sample weights. The additional variable *Current value* was collected and added to the data file after the sample was taken.
- **recidivism.sav.** This is a hypothetical data file that concerns a government law enforcement agency’s efforts to understand recidivism rates in their area of jurisdiction. Each case corresponds to a previous offender and records their

demographic information, some details of their first crime, and then the time until their second arrest, if it occurred within two years of the first arrest.

- **recidivism_cs_sample.sav.** This is a hypothetical data file that concerns a government law enforcement agency's efforts to understand recidivism rates in their area of jurisdiction. Each case corresponds to a previous offender, released from their first arrest during the month of June, 2003, and records their demographic information, some details of their first crime, and the data of their second arrest, if it occurred by the end of June, 2006. Offenders were selected from sampled departments according to the sampling plan specified in *recidivism_cs.csplan*; because it makes use of a probability-proportional-to-size (PPS) method, there is also a file containing the joint selection probabilities (*recidivism_cs_jointprob.sav*).
- **rfm_transactions.sav.** A hypothetical data file containing purchase transaction data, including date of purchase, item(s) purchased, and monetary amount of each transaction.
- **salesperformance.sav.** This is a hypothetical data file that concerns the evaluation of two new sales training courses. Sixty employees, divided into three groups, all receive standard training. In addition, group 2 gets technical training; group 3, a hands-on tutorial. Each employee was tested at the end of the training course and their score recorded. Each case in the data file represents a separate trainee and records the group to which they were assigned and the score they received on the exam.
- **satisf.sav.** This is a hypothetical data file that concerns a satisfaction survey conducted by a retail company at 4 store locations. 582 customers were surveyed in all, and each case represents the responses from a single customer.
- **screws.sav.** This data file contains information on the characteristics of screws, bolts, nuts, and tacks .
- **shampoo_ph.sav.** This is a hypothetical data file that concerns the quality control at a factory for hair products. At regular time intervals, six separate output batches are measured and their pH recorded. The target range is 4.5–5.5.
- **ships.sav.** A dataset presented and analyzed elsewhere that concerns damage to cargo ships caused by waves. The incident counts can be modeled as occurring at a Poisson rate given the ship type, construction period, and service period. The aggregate months of service for each cell of the table formed by the cross-classification of factors provides values for the exposure to risk.

- **site.sav.** This is a hypothetical data file that concerns a company's efforts to choose new sites for their expanding business. They have hired two consultants to separately evaluate the sites, who, in addition to an extended report, summarized each site as a "good," "fair," or "poor" prospect.
- **siteratings.sav.** This is a hypothetical data file that concerns the beta testing of an e-commerce firm's new Web site. Each case represents a separate beta tester, who scored the usability of the site on a scale from 0–20.
- **smokers.sav.** This data file is abstracted from the 1998 National Household Survey of Drug Abuse and is a probability sample of American households. Thus, the first step in an analysis of this data file should be to weight the data to reflect population trends.
- **smoking.sav.** This is a hypothetical table introduced by Greenacre . The table of interest is formed by the crosstabulation of smoking behavior by job category. The variable *Staff Group* contains the job categories *Sr Managers*, *Jr Managers*, *Sr Employees*, *Jr Employees*, and *Secretaries*, plus the category *National Average*, which can be used as supplementary to an analysis. The variable *Smoking* contains the behaviors *None*, *Light*, *Medium*, and *Heavy*, plus the categories *No Alcohol* and *Alcohol*, which can be used as supplementary to an analysis.
- **storebrand.sav.** This is a hypothetical data file that concerns a grocery store manager's efforts to increase sales of the store brand detergent relative to other brands. She puts together an in-store promotion and talks with customers at check-out. Each case represents a separate customer.
- **stores.sav.** This data file contains hypothetical monthly market share data for two competing grocery stores. Each case represents the market share data for a given month.
- **stroke_clean.sav.** This hypothetical data file contains the state of a medical database after it has been cleaned using procedures in the Data Preparation option.
- **stroke_invalid.sav.** This hypothetical data file contains the initial state of a medical database and contains several data entry errors.
- **stroke_survival.** This hypothetical data file concerns survival times for patients exiting a rehabilitation program post-ischemic stroke face a number of challenges. Post-stroke, the occurrence of myocardial infarction, ischemic stroke, or hemorrhagic stroke is noted and the time of the event recorded. The sample is left-truncated because it only includes patients who survived through the end of the rehabilitation program administered post-stroke.

- **stroke_valid.sav.** This hypothetical data file contains the state of a medical database after the values have been checked using the Validate Data procedure. It still contains potentially anomalous cases.
- **survey_sample.sav.** This hypothetical data file contains survey data, including demographic data and various attitude measures.
- **tastetest.sav.** This is a hypothetical data file that concerns the effect of mulch color on the taste of crops. Strawberries grown in red, blue, and black mulch were rated by taste-testers on an ordinal scale of 1 to 5 (far below to far above average). Each case represents a separate taste-tester.
- **telco.sav.** This is a hypothetical data file that concerns a telecommunications company's efforts to reduce churn in their customer base. Each case corresponds to a separate customer and records various demographic and service usage information.
- **telco_extra.sav.** This data file is similar to the *telco.sav* data file, but the "tenure" and log-transformed customer spending variables have been removed and replaced by standardized log-transformed customer spending variables.
- **telco_missing.sav.** This data file is a subset of the *telco.sav* data file, but some of the demographic data values have been replaced with missing values.
- **testmarket.sav.** This hypothetical data file concerns a fast food chain's plans to add a new item to its menu. There are three possible campaigns for promoting the new product, so the new item is introduced at locations in several randomly selected markets. A different promotion is used at each location, and the weekly sales of the new item are recorded for the first four weeks. Each case corresponds to a separate location-week.
- **testmarket_1month.sav.** This hypothetical data file is the *testmarket.sav* data file with the weekly sales "rolled-up" so that each case corresponds to a separate location. Some of the variables that changed weekly disappear as a result, and the sales recorded is now the sum of the sales during the four weeks of the study.
- **tree_car.sav.** This is a hypothetical data file containing demographic and vehicle purchase price data.
- **tree_credit.sav.** This is a hypothetical data file containing demographic and bank loan history data.
- **tree_missing_data.sav** This is a hypothetical data file containing demographic and bank loan history data with a large number of missing values.

- **tree_score_car.sav.** This is a hypothetical data file containing demographic and vehicle purchase price data.
- **tree_textdata.sav.** A simple data file with only two variables intended primarily to show the default state of variables prior to assignment of measurement level and value labels.
- **tv-survey.sav.** This is a hypothetical data file that concerns a survey conducted by a TV studio that is considering whether to extend the run of a successful program. 906 respondents were asked whether they would watch the program under various conditions. Each row represents a separate respondent; each column is a separate condition.
- **ulcer_recurrence.sav.** This file contains partial information from a study designed to compare the efficacy of two therapies for preventing the recurrence of ulcers. It provides a good example of interval-censored data and has been presented and analyzed elsewhere .
- **ulcer_recurrence_recoded.sav.** This file reorganizes the information in *ulcer_recurrence.sav* to allow you model the event probability for each interval of the study rather than simply the end-of-study event probability. It has been presented and analyzed elsewhere .
- **verd1985.sav.** This data file concerns a survey . The responses of 15 subjects to 8 variables were recorded. The variables of interest are divided into three sets. Set 1 includes *age* and *marital*, set 2 includes *pet* and *news*, and set 3 includes *music* and *live*. *Pet* is scaled as multiple nominal and *age* is scaled as ordinal; all of the other variables are scaled as single nominal.
- **virus.sav.** This is a hypothetical data file that concerns the efforts of an Internet service provider (ISP) to determine the effects of a virus on its networks. They have tracked the (approximate) percentage of infected e-mail traffic on its networks over time, from the moment of discovery until the threat was contained.
- **waittimes.sav.** This is a hypothetical data file that concerns customer waiting times for service at three different branches of a local bank. Each case corresponds to a separate customer and records the time spent waiting and the branch at which they were conducting their business.
- **webusability.sav.** This is a hypothetical data file that concerns usability testing of a new e-store. Each case corresponds to one of five usability testers and records whether or not the tester succeeded at each of six separate tasks.

- **wheeze_steubenville.sav.** This is a subset from a longitudinal study of the health effects of air pollution on children . The data contain repeated binary measures of the wheezing status for children from Steubenville, Ohio, at ages 7, 8, 9 and 10 years, along with a fixed recording of whether or not the mother was a smoker during the first year of the study.
- **workprog.sav.** This is a hypothetical data file that concerns a government works program that tries to place disadvantaged people into better jobs. A sample of potential program participants were followed, some of whom were randomly selected for enrollment in the program, while others were not. Each case represents a separate program participant.

Index

- CHAID, 1
 - Bonferroni adjustment, 12
 - intervals for scale independent variables, 14
 - maximum iterations, 12
 - resplitting merged categories, 12
 - splitting and merging criteria, 12
- classification table, 84
- collapsing tree branches, 48
- command syntax
 - creating selection and scoring syntax for decision trees, 45, 57
- costs
 - misclassification, 19
 - tree models, 92
- crossvalidation
 - trees, 9
- CRT, 1
 - impurity measures, 15
 - pruning, 17

- decision trees , 1
 - CHAID method, 1
 - CRT method, 1
 - Exhaustive CHAID method, 1
 - forcing first variable into model, 1
 - measurement level, 1
 - QUEST method, 1, 16

- gain, 80
- gains chart, 82
- Gini, 15

- hiding nodes
 - vs. pruning, 17
- hiding tree branches, 48

- impurity
 - CRT trees, 15

- index
 - tree models, 80
- index chart, 83
- index values
 - trees, 34

- measurement level
 - decision trees, 1
 - in tree models, 61
- misclassification
 - costs, 19
 - rates, 84
 - trees, 34
- missing values
 - in tree models, 109
 - trees, 27
- model summary table
 - tree models, 77

- node number
 - saving as variable from decision trees, 29
- nodes
 - selecting multiple tree nodes, 48

- ordered twoing, 15

- predicted probability
 - saving as variable from decision trees, 29
- predicted values
 - saving as variable from decision trees, 29
 - saving for tree models, 85
- profits
 - prior probability, 23
 - trees, 21, 34
- pruning decision trees
 - vs. hiding nodes, 17

- QUEST, 1, 16
 - pruning, 17

- random number seed
 - decision tree validation, 9
- response
 - tree models, 80
- risk estimates
 - for categorical dependent variables, 84
 - for scale dependent variables in Decision Tree procedure, 104
 - trees, 34
- rules
 - creating selection and scoring syntax for decision trees, 45, 57

- sample files
 - location, 120
- scale variables
 - dependent variables in Decision Tree procedure, 97
- scores
 - trees, 25
- scoring
 - tree models, 97
- selecting multiple tree nodes, 48
- significance level for splitting nodes, 16
- split-sample validation
 - trees, 9
- SQL
 - creating SQL syntax for selection and scoring, 45, 57
- surrogates
 - in tree models, 109, 117
- syntax
 - creating selection and scoring syntax for decision trees, 45, 57

- tree models, 80
- trees, 1
 - applying models, 97
 - CHAID growing criteria, 12
 - charts, 39
 - colors, 54
 - controlling node size, 11
 - controlling tree display, 31, 53
 - crossvalidation, 9
 - CRT method, 15
 - custom costs, 92
 - editing, 48
 - effects of measurement level, 61
 - effects of value labels, 66
 - fonts, 54
 - gains for nodes table, 80
 - generating rules, 45, 57
 - hiding branches and nodes, 48
 - index values, 34
 - intervals for scale independent variables, 14
 - limiting number of levels, 11
 - misclassification costs, 19
 - misclassification table, 34
 - missing values, 27, 109
 - model summary table, 77
 - node chart colors, 54
 - predictor importance, 34
 - prior probability, 23
 - profits, 21
 - pruning, 17
 - risk estimates, 34
 - risk estimates for scale dependent variables, 104
 - saving model variables, 29
 - saving predicted values, 85
 - scale dependent variables, 97
 - scaling tree display, 51
 - scores, 25
 - scoring, 97
 - selecting cases in nodes, 86
 - selecting multiple nodes, 48
 - showing and hiding branch statistics, 31
 - split-sample validation, 9
 - surrogates, 109, 117
 - terminal node statistics, 34
 - text attributes, 54
 - tree contents in a table, 31
 - tree in table format, 79
 - tree map, 50
 - tree orientation, 31
 - working with large trees, 50
- twoing, 15

- validation
 - trees, 9
- value labels
 - trees, 66

weighting cases

fractional weights in decision trees, 1