Chapter 4: Building Decision Tree Models to Predict Response and Risk

- Decision trees and regression trees
- Growing the tree, classifying the nodes, and pruning
- **Decision Tree** node
- Predicting response and risk by presenting two examples based on a hypothetical auto insurance company
- The response to a mail order campaign
- Risk as measured by claim frequency as a regression tree
- Shows how to develop decision trees interactively

Introduction

- This chapter shows you how to build decision tree models to predict a categorical target and how to build *regression tree models* to predict a continuous target.
- Examples are presented in our text.
- The first example shows how to build a *decision tree model* to predict *response* to direct mail. In this example, the target variable is binary, taking on the values *response* and *no response*.
- The second example shows how to build a regression tree model to forecast a continuous (but interval-scaled) target often used in the auto insurance industry, namely *loss frequency* .
- *Loss frequency* can also be modeled as a categorical target variable if it takes on only a few values but in this example it is treated as a continuous target.

An Overview of the Tree Methodology in SAS Enterprise Miner

- Decision Trees
- Decision Tree Models
- Decision Tree Models vs. Logistic Regression Models
- Calculation of the Worth of a Tree
- Roles of the Training and Validation Data in the Development of a Decision Tree
- Regression Tree

Decision Trees

- A decision tree represents a hierarchical segmentation of the data.
- The *original segment* is the entire data set and it is called the *root node* of the tree. The original segment is first partitioned into two or more *segments* by applying a series of simple rules.
- Each rule assigns an observation to a segment based on the value of an *input* (*explanatory variable*) for that observation. In a similar fashion, each resulting segment is further partitioned into *sub-segments* (segments within a segment); each sub-segment is further partitioned into more subsegments, and so on.
- This process continues until no more partitioning is possible. This process of segmenting is called *recursive partitioning*, and it results in a hierarchy of segments within segments.
- The hierarchy is called a *tree*, and each segment or sub-segment is called a *node*.

Decision Tree Models

A decision tree model is composed of several parts:

- node definitions, or rules, to assign each record of a data set to a leaf node
- posterior probabilities of each leaf node
- the assignment of a target level to each leaf node

Details: See Figure 4.1 in our current text

Remarks

- Node definitions are developed using the Training data set and are stated in terms of input ranges.
- Posterior probabilities are calculated for each node using the Training data set. The assignment of the target level to each node is also done during the training phase using the Training data set.
- Posterior probabilities are *observed proportions* of target levels within each node in the training data set.
- Take the example of a binary target. A binary target has two levels, which can be represented by response and no response, or 1 and 0. The *posterior probability* of response in a node is the proportion of records with the target level equal to response, or 1, within that node. Similarly, the *posterior probability* of no response of a node is the proportion of records with the target level equal to no response, or 0, within that node.
- These posterior probabilities are determined during the training of the tree and they become part of the decision tree model.
- As mentioned above, they are calculated using the training data.

Table 4.1

Actual target level/class	Decision1	Decision2
1	$10	0
0	-$1	0

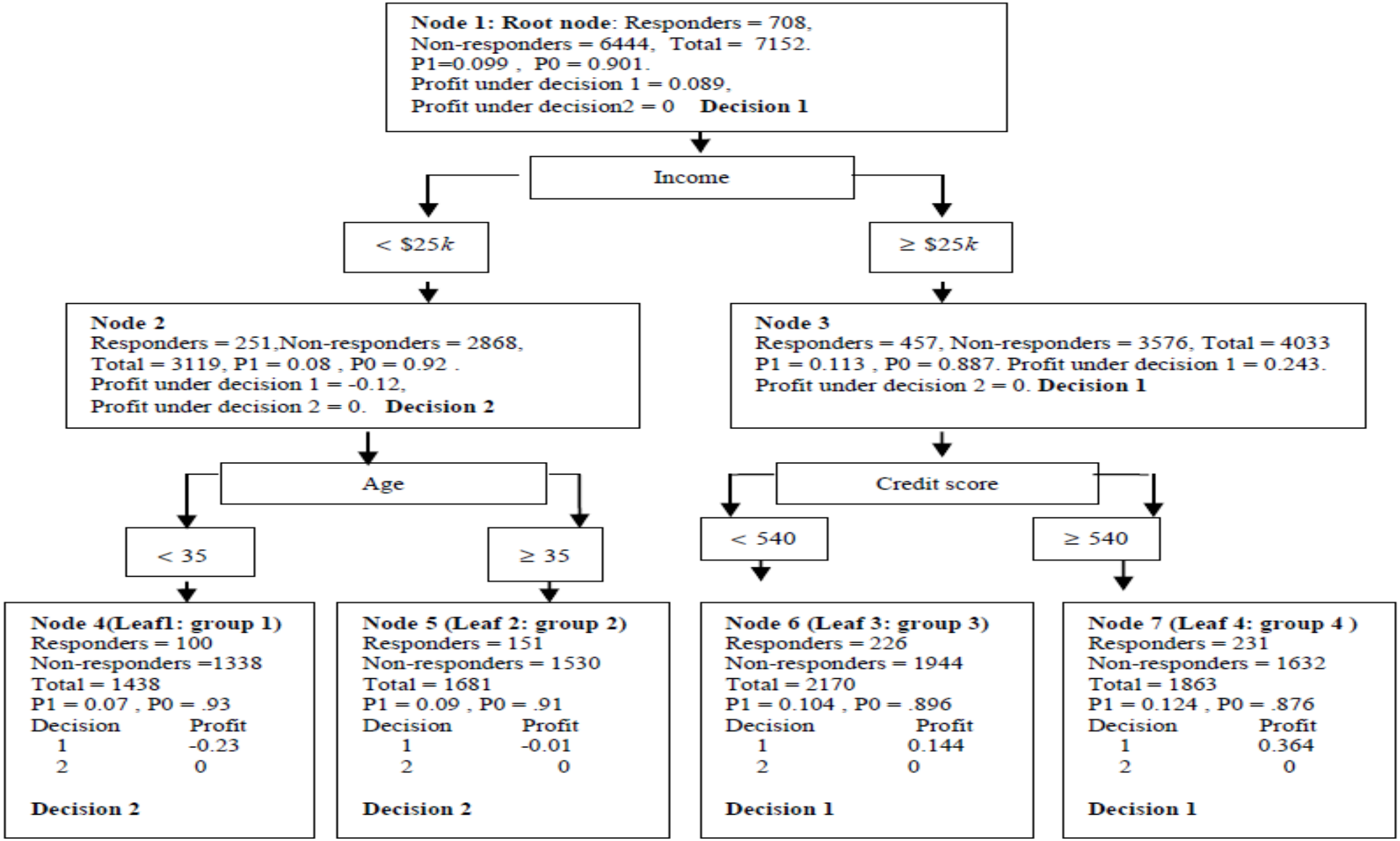
In Table 4.1, Decision1 means assigning a record to the target level 1 (response), and Decision2 means assigning a record to target level 0 (non-response). This profit matrix indicates that if a true responder is correctly classified as a responder, then the profit earned is $10. If a true non-responder is classified as responder, a loss of $1 is incurred. If a true responder is classified as non-responder, then the profit is zero.[1] If a true non-responder is classified as a non-responder, then also the profit is zero.

Suppose E_1 represents the expected profit under Decision1, and E_2 represents the expected profit under Decision2. If $E_1 > E_2$ then Decision1 is taken, and the record is assigned to the target level 1. If $E_2 > E_1$ then Decision2 is taken, and the record is assigned to the target level 0. Given a profit matrix, E_1 and E_2 are based on the *posterior probabilities* of the node to which the record belongs. In general, each record is assigned the posterior probabilities of the node in which it resides. The expected profits (E_1 and E_2) are calculated as $E_1 = 10 * \hat{P}_1 - 1 * \hat{P}_0$ and $E_2 = 0 * \hat{P}_1 + 0 * \hat{P}_0$ for each record, where $\hat{P}_1$ and $\hat{P}_0$ represent the posterior probabilities of response and no response for the record.

It follows from these profit equations that all the records in a node have the same posterior probabilities, and are therefore assigned to the same target level.

An example of a tree is shown in Figure 4.1. The tree shown in Figure 4.1 is handcrafted[2] in order to highlight the process of assigning target levels to the nodes.

Figure 4.1


Node 1: Root node: Responders = 708,
Non-responders = 6444, Total = 7152.
P1=0.099 , P0 = 0.901.
Profit under decision 1 = 0.089,
Profit under decision2 = 0 Decision 1
Income
< $25k
≥ $25k
Node 2
Responders = 251,Non-responders = 2868,
Total = 3119, P1 = 0.08 , P0 = 0.92 .
Profit under decision 1 = -0.12,
Profit under decision 2 = 0. Decision 2
Node 3
Responders = 457, Non-responders = 3576, Total = 4033
P1 = 0.113 , P0 = 0.887. Profit under decision 1 = 0.243.
Profit under decision 2 = 0. Decision 1
Age
Credit score
< 35
≥ 35
< 540
≥ 540
Node 4(Leaf1: group 1)
Responders = 100
Non-responders =1338
Total = 1438
P1 = 0.07 , P0 = .93
Decision Profit
1 -0.23
2 0
Decision 2
Node 5 (Leaf 2: group 2)
Responders = 151
Non-responders = 1530
Total = 1681
P1 = 0.09 , P0 = .91
Decision Profit
1 -0.01
2 0
Decision 2
Node 6 (Leaf 3: group 3)
Responders = 226
Non-responders = 1944
Total = 2170
P1 = 0.104 , P0 = .896
Decision Profit
1 0.144
2 0
Decision 1
Node 7 (Leaf 4: group 4)
Responders = 231
Non-responders = 1632
Total = 1863
P1 = 0.124 , P0 = .876
Decision Profit
1 0.364
2 0
Decision 1

Decision Tree Models vs. Logistic Regression Models

- A decision tree model is composed of a set of rules that can be applied to partition the data into disjoint groups.
- Unlike the logistic regression model, the tree model does not have any equations or coefficients.
- Instead, for each disjoint group or leaf node, it contains posterior probabilities which are themselves used as predicted probabilities.
- These posterior probabilities are developed during the *training* of the tree.

Applying the Decision Tree Model to Prospect Data

- The SAS code generated by the **Tree** node can be applied to the prospect data set for purposes of prediction and classification.
- The code places each record in one of the predefined leaf nodes.
- The predicted probability of a target level (such as response) for each record is the posterior probability associated with the leaf node into which the record is placed.
- Since a target level is assigned to each leaf node, all records that fall into a leaf node are assigned the same target level. For example, the target level might be responder or non-responder.

Calculation of the Worth of a Tree

- There are situations in which you must calculate the *worth* of a tree.
- The worth of a tree can be calculated using the Validation data set, Test data set, or any other data set where the target levels are known and the inputs necessary for defining the leaf nodes are available.
- Only leaf nodes are used for calculating the worth.

Roles of the Training and Validation Data in the Development of a Decision Tree

- To develop a tree model, you need two data sets.
- The first is for training the model, and the second is for pruning or fine-tuning the model. A third data set is optionally needed for an independent assessment of the model.
- These three data sets are referred to in SAS Enterprise Miner as Training, Validation, and Test, respectively.

Regression Tree

- When the target is continuous, the tree is called a *regression tree*.
- The regression tree model consists of the rules (node definitions) for partitioning the data set into leaf nodes and also the mean of the target for each leaf node.

Development of the Tree in SAS Enterprise Miner

- Growing an Initial Tree
- *P*-value Adjustment Options
- Controlling Tree Growth: Stopping Rules
- Pruning: Selecting the Right-Sized Tree Using Validation Data
- Step-by-Step Illustration of Growing and Pruning a Tree
- Average Profit vs. Total Profit for Comparing Trees of Different Sizes
- Accuracy /Misclassification Criterion in Selecting the Right-sized Tree
- Assessment of a Tree or Sub-tree Using Average Square Error
- Selection of the Right-sized Treeracy /Misclassification Criterion in Selecting the Right-sized Tree

Growing an Initial Tree

- As described in Section 4.2.1 of the text, growing a tree involves successively partitioning the data set into segments using the method of *recursive partitioning*.
- Inputs are selected at each step of the sequence of recursive partitioning.
- Two-way, or binary, splits of the inputs, although SAS Enterprise Miner can also perform three-way and multi-way splits.
- In addition, the discussion assumes that the target is binary.

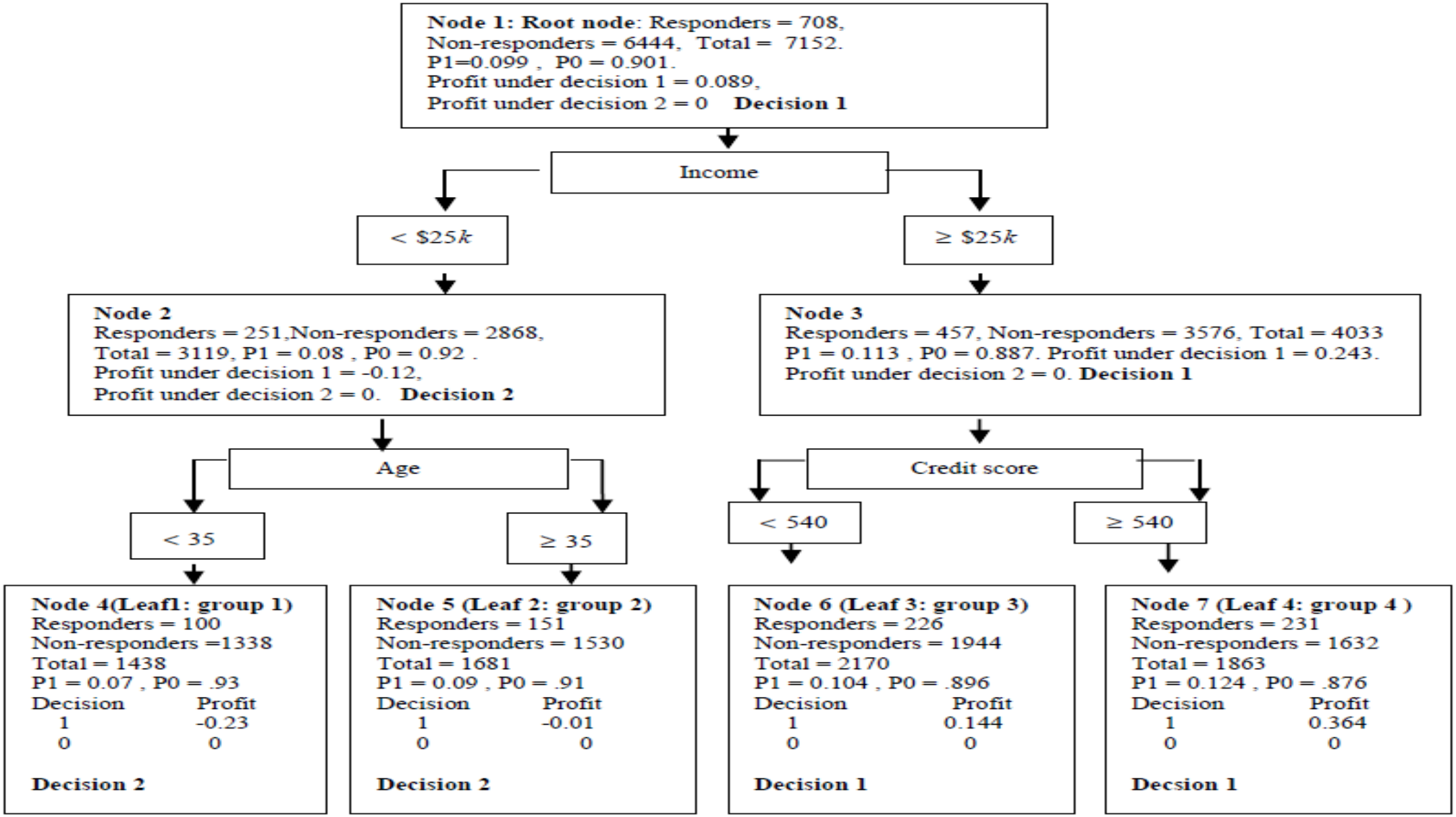
Step-by-Step Illustration of Growing and Pruning a Tree

- The tree is grown using the Training data set with 7152 records.
- The rules of partition are developed and the nodes are classified (so as to maximize profit or other criterion chosen) using the Training data set.
- The Validation data set that is used for pruning consists of 5364 records.
- The rules of partition and the node classifications that were developed on the Training data set are then applied to all 5364 records, in effect reproducing the tree with the Validation data set.
- The node definitions and classification of the nodes are the same, but the records in each node are from the validation data set.
- The process begins by displaying the maximal tree constructed from the training data set:

Step IA: A tree is developed using the training data set

The initial tree is developed using the Training data set. This tree is shown in Figure 4.4.

Figure 4.4



Decision Tree Model to Predict Response to Direct Marketing

- Decision Tree Model to Predict Response to Direct Marketing
- Testing Model Performance with a Test Data Set
- Applying the Decision Tree Model to Score a Data Set
- Developing a Regression Tree Model to Predict Risk
- Summary of the Regression Tree Model to Predict Risk

A Decision Tree Model to Predict Response to Direct Marketing

- In this section, a response model using the **Decision Tree** node. The model is based on simulated data representing the direct mail campaign of a hypothetical insurance company.
- The main purpose of this section is to show how to develop a response model using the **Decision Tree** node, and to make you familiar with the various components of the Results window.
- Since this model predicts an individual's response to direct mailing, it is called a *response model*.
- In this model, the target variable is binary: it takes on values of response and no response.

Applying the Decision Tree Model to Score a Data Set

- This section shows the application of the decision tree model developed earlier to score a prospect data set.
- The records in the prospect data set do not have the value of the target variable.
- The target variable has two levels, response and no response. We use the model to predict the probability of response for each record in the prospect data set.

Developing Decision Trees Interactively

- Interactively Modifying an Existing Decision Tree
- Growing a Tree Interactively Starting from the Root Node
- Developing the Maximal Tree in Interactive Mode

Summary

• *Growing a decision tree* means arriving at certain rules for segmenting a given data set, stated in terms of ranges of values for inputs, and assigning a target class3 (such as response or non-response) to each segment that is created.

• The *Training data set* is used for developing the rules for the segmentation, for calculating the *posterior probabilities* of each segment (or the target mean of each segment when the target variable iscontinuous), and for assigning each segment to a target class when the target is categorical.

• Node definitions, target means, posterior probabilities, and the assignment of the target levels to nodes do not change when the tree is applied to the Validation, or Prospect, data set.

• SAS Enterprise Miner offers several criteria for splitting a node; you can select the criterion suitable for the type of model being developed by setting the **Splitting Rule Criterion** property to an appropriate value.

Summary

• In general, there are a number of sub-trees within the initial tree that you need to examine during the tree development.

- The Training data set is used to create this initial tree, while the *Validation data set* is used to select the best sub-tree by applying an assessment measure that you choose.
- Several measures of assessment are available in SAS Enterprise Miner, and you can select one of them by setting the **Subtree Assessment Measure** property to a certain value.
- Selecting the best sub-tree is often referred to as *pruning*.

Summary

• SAS Enterprise Miner offers different methods for selecting a sub-tree: instead of applying an assessment measure to select a sub-tree, you can also select the tree with the maximum number of leaves, or the one with a specified number of leaves (regardless of whether the tree has maximum profit, or minimum cost, or meets any other criterion of assessment.)

• A *decision tree model* consists finally of only the *leaf* nodes (also known as the terminal nodes) of the selected tree, the attributes of which can be used for scoring prospect data, etc.

• The definition of the leaf nodes, the target classes assigned to the nodes, and their posterior probabilities can all be seen in the SAS code (score code) generated by SAS Enterprise Miner. You can export the SAS code if the scoring is to be applied to external data.

Summary

- The two models developed in this chapter, one for a binary target and one for a continuous target, serve as examples of how to use the **Decision Tree** node: how to interpret the results given in the Output window and how to interpret the various types of graphs and tables associated with each of them.

• The Regression Tree model developed for predicting risk is used to rank the customers according to their degree of risk. Each risk group is then profiled and represented as an exclusive segment of the tree model.

• You can develop decision trees interactively in SAS Enterprise Miner. You can modify a previously developed tree or grow a whole tree interactively, starting from the root node. You can also grow a maximal tree interactively.

Pearson's Chi-Square Test

As an illustration of the Chi-square test, consider a simple example. Suppose there are 100 cases in the parent node (τ), of which 10 are responders and 90 are non-responders. Suppose this node is split into two child nodes, A and B. Let A have 50 cases, and let B have 50 cases. Also, suppose there are 9 responders in child node A, and there is 1 responder in child node B. The parent node has 10 responders and 90 non-responders.

Here is a 2 x 2 contingency table representing the split of parent node τ :

	Child node A	**Child node B**
Responders	9	1
Non-responders	41	49

The null hypothesis is that the child nodes A and B are not different from their parent node in terms of class composition (i.e., their mix of *responders* and *non-responders*). Under the null hypothesis, each child node is similar to the parent node, and contains 10% responders and 90% non-responders.

Expected frequencies under the null hypothesis are as follows:

	Child node A	Child node B
Responders	5	5
Non-responders	45	45

$$\chi^2 = \sum \frac{(O-E)^2}{E} = \frac{(4)^2}{5} + \frac{(-4)^2}{45} + \frac{(-4)^2}{5} + \frac{(4)^2}{45} = 7.1$$

A Chi-square value of 7.1 implies the following values for *p*-value and for logworth:

p-value = 0.00766 and logworth = $\log_{10}$(*p*-value) = 2.11546.

The *p*value is the probability of the Chi_Square statistic taking a value of 7.1 or higher when there is no difference in the child nodes. In this case it is very small, 0.00766. This indicates that in this case, the child nodes do differ significantly with respect to their composition of responders and non-responders.