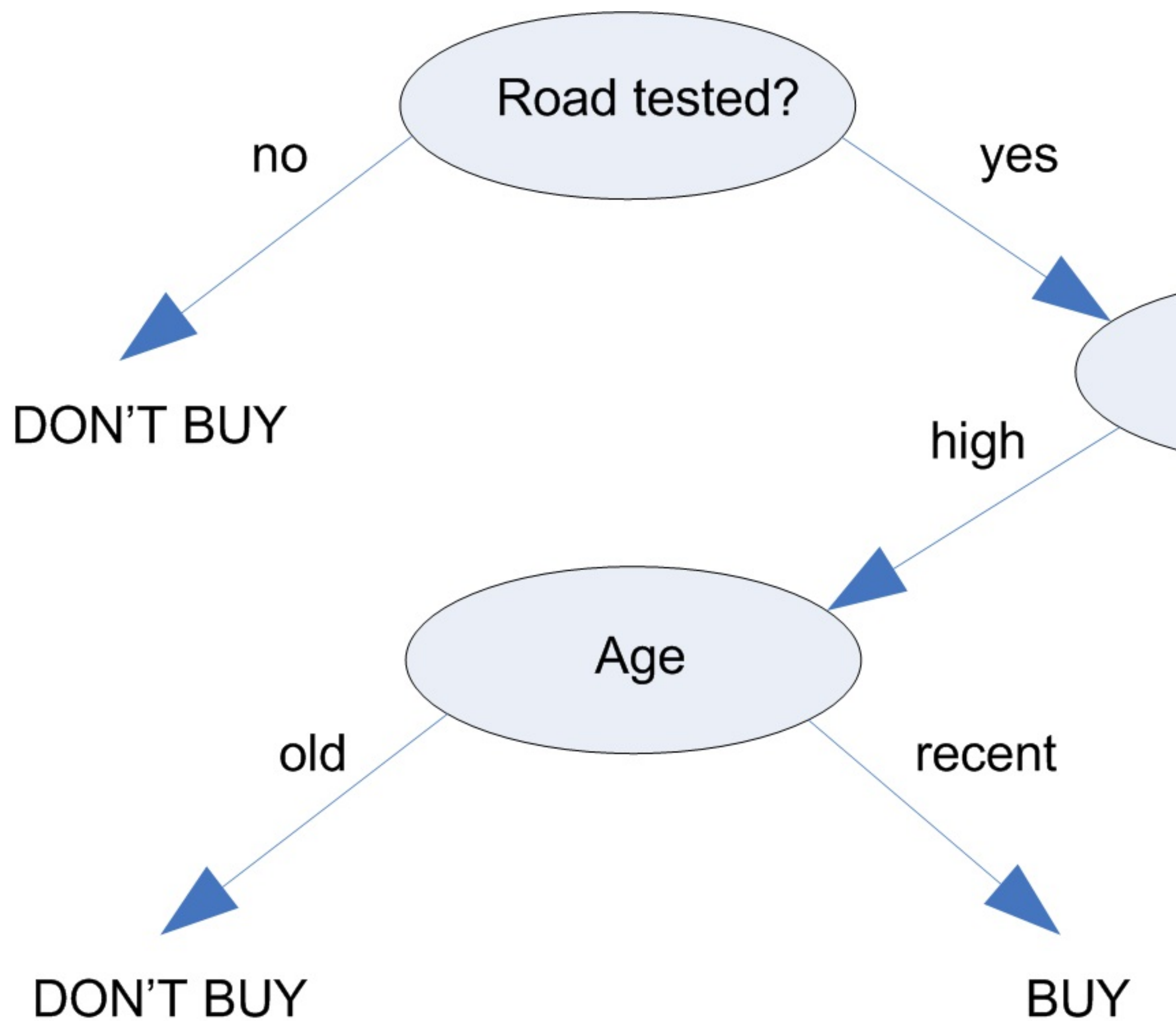


Decision Tree Models

Decision tree models allow you to develop classification systems that predict or classify future observations based on a set of decision rules. If you have data divided into classes that interest you (for example, high- versus low-risk loans, subscribers versus nonsubscribers, voters versus nonvoters, or types of bacteria), you can use your data to build rules that you can use to classify old or new cases with maximum accuracy. For example, you might build a tree that classifies credit risk or purchase intent based on age and other factors.

This approach, sometimes known as **rule induction**, has several advantages. First, the reasoning process behind the model is clearly evident when browsing the tree. This is in contrast to other "black box" modeling techniques in which the internal logic can be difficult to work out.

Simple decision tree for buying a car



Second, the process will automatically include in its rule only the attributes that really matter in making a decision. Attributes that do not contribute to the accuracy of the tree are ignored. This can yield very useful information about the data and can be used to reduce the data to relevant fields before training another learning technique, such as a neural net.





Decision tree model nuggets can be converted into a collection of if-then rules (a **rule set**), which in many cases show the information in a more comprehensible form. The decision-tree presentation is useful when you want to see how attributes in the data can **split**, or **partition**, the population into subsets relevant to the problem. The rule set presentation is useful if you want to see how particular groups of items relate to a specific conclusion. For example, the following rule gives us a **profile** for a group of cars that is worth buying:

IF tested = 'yes'AND mileage = 'low'THEN -> 'BUY'.



Tree-Building Algorithms

Four algorithms are available for performing classification and segmentation analysis. These algorithms all perform basically the same thing--they examine all of the fields of your dataset to find the one that gives the best classification or prediction by splitting the data into subgroups. The process is applied recursively, splitting subgroups into smaller and smaller units until the tree is finished (as defined by certain stopping criteria). The target and input fields used in tree building can be continuous (numeric range) or categorical, depending on the algorithm used. If a continuous target is used, a regression tree is generated; if a categorical target is used, a classification tree is generated.

	The Classification and Regression (C&R) Tree node generates a decision tree that allows you to predict or classify future observations. The method uses recursive partitioning to split the training records into segments by minimizing the impurity at each step, where a node in the tree is considered “pure” if 100% of cases in the node fall into a specific category of the target field. Target and input fields can be numeric ranges or categorical (nominal, ordinal, or flags); all splits are binary (only two subgroups). See the topic C&R Tree Node for more information.
	The CHAID node generates decision trees using chi-square statistics to identify optimal splits. Unlike the C&R Tree and QUEST nodes, CHAID can generate nonbinary trees, meaning that some splits have more than two branches. Target and input fields can be numeric range (continuous) or categorical. Exhaustive CHAID is a modification of CHAID that does a more thorough job of examining all possible splits but takes longer to compute. See the topic CHAID Node for more information.
	The QUEST node provides a binary classification method for building decision trees, designed to reduce the processing time required for large C&R Tree analyses while also reducing the tendency found in classification tree methods to favor inputs that allow more splits. Input fields can be numeric ranges (continuous), but the target field must be categorical. All splits are binary. See the topic QUEST Node for more information.
	The C5.0 node builds either a decision tree or a rule set. The model works by splitting the sample based on the field that provides the maximum information gain at each level. The target field must be categorical. Multiple splits into more than two subgroups are allowed. See the topic C5.0 Node for more information.

General Uses of Tree-Based Analysis

The following are some general uses of tree-based analysis:

- Segmentation.** Identify persons who are likely to be members of a particular class.
- 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. Prediction can also mean attempts to relate predictive attributes to values of a continuous variable.
- 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 banding continuous variables. Recode group predictor categories and continuous variables with minimal loss of information.