e-PG Pathshala

Subject : Computer Science

Paper: Machine Learning

Module: Decision Trees

Module No: CS/ML/13

Quadrant I – e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning. In this module we will be discussing another popular machine learning technique – the Decision Tree approach in detail.

Learning Objectives:

The learning objectives of this module are as follows:

- To discuss the use and key requirements of decision trees
- To outline how decision trees work
- To illustrate decision trees with simple examples

13.1 Introduction

Let us consider a typical learning problem where we are given a set of training cases and their attribute values and we need to try to determine the target attribute value of new hitherto unseen examples. This type of problems can be either classification or prediction problems. Figure 13.1 shows the flow of a typical classification problem where initially we are given a set of training samples. Each sample is associated with a set of features or attributes with one of the attributes specifying the class. The classification model is determined for the target attribute which will be a function of the values of the other attributes. The classification model is then used to obtain the classification outcome given hitherto unseen examples.

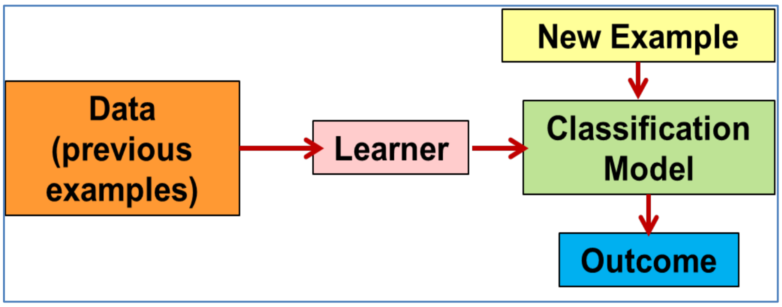


Figure 13.1 Classification Problem

This method of learning is called an inductive learning task where particular facts about training samples are used to make more generalized conclusions.

13.1.1 Why Decision Trees?

Decision trees are a powerful and very popular inductive learning technique used for classification & prediction. Decision trees is essence are if then *rules*, which is easily interpretable by humans. Decision trees are used to discover the relationships between input variables and the desired target (output) variable. When using decision trees we have a predictive model based on a series of Boolean tests. These smaller Boolean tests are less complex than a single stage classifier.

13.1.2 Advantages of Decision Trees

Some of the advantages of using decision trees is discussed below. Decision trees are relatively fast compared to other classification models. They give similar and, in some cases, even better accuracy compared to other models. One of the major advantages of decision trees is that they are simple and easy to understand. Moreover, these decision trees can be converted into simple and easy to understand classification rules.

13.1.3 Key Requirements of Decision Tree Learning

In order to use decision trees as the chosen machine learning technique, the problem should meet the following requirements:

- **Attribute-value description:** The problem should be expressible in terms of features or attributes and their values (e.g., the attribute temperature which can have values such as hot, mild, cold). The choice and extraction of this collection of features is an important aspect of any learning problem.
- **Predefined classes (target values):** The required target function has discrete output values or classes which can be either Boolean or multiclass. Please note that it is possible to express problem that demand continuous values as outcomes in terms of some discrete values.
- **Sufficient data:** There must be sufficient training data in order to learn the model. Since we are trying to generalize and learn a classification model from the given training data, insufficient data can give a completely wrong classification.

13.2 Decision Tree

What exactly is a decision tree? It is basically a visual representation of choices and a way of breaking down complicated combination of feature values into easier to handler simpler set of values. Decision tree is a classifier in the form of a tree structure. It is a tree structure, where each internal node (nonleaf node) called decision node denotes a test on a single attribute, each branch represents an outcome of the test or the split of the values of one attribute, and leaf node (or terminal) holds the value of the target attribute or in other words the class label. The topmost node in a tree is the root node. The path through the decision tree is a conjunction of attribute-value literals that are obtained by conducting disjunction tests at each tree node to arrive at the final decision available at the leaf.

Decision trees classify instances or examples by starting at the root of the tree and depending on the value of the attribute for the example, choosing the appropriate sub-tree at each decision node until a leaf node is reached. An unknown case is classified by following a matching path to a leaf node.

13.2.1 Creation of a Decision Tree

We first make a list of attributes that we can measure. The attributes chosen must be discrete or must be convertible into a discrete form. We then choose the target attribute that we want to predict. We create what is called an experience table that lists what we have seen in the past, that is the values of the list of attributes and the target attribute associated with each given training example.

13.2.2 Example of a Decision Tree

Let us look at a sample decision tree given in Figure 13.2. The rectangle nodes, in our example Leave At, Road Block and Accident nodes are the decision nodes. Leave At is the root of the decision tree and has three arcs corresponding to the three values that Leave At can take namely 10AM, 8AM, and 9AM. Road Block and Accident, the two internal nodes are Boolean. The leaf nodes corresponding to the values of the target attribute which is the time taken for travel which has possible values namely Short, Long and Medium.

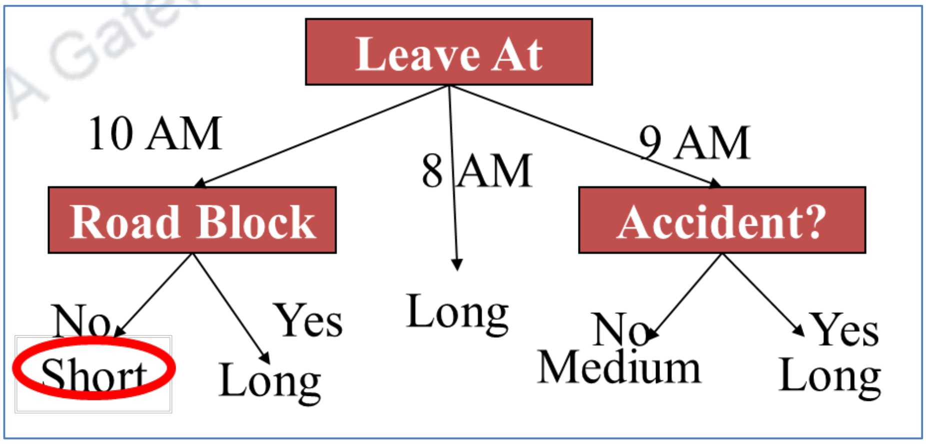


Figure 13.2 Example of a Decision Tree

13.2.3 Inductive Learning

Now let us understand how to follow a path in the decision tree (Figure 13.2). The question asked is ***"If we leave at 10 AM and there are no cars stalled on the road, what will our commute time be?"*** In this decision tree, we make a series of Boolean decisions and followed the corresponding branch. First we go to the root of the tree and check for when we left that is the value of the attribute Leave At. Since we will leave at 10AM we will follow that edge. Now we come across the next decision node on the path which is Road Block. Since we know that we did not see cars stalled we assume that there was no Road Block. Now when we follow the edge no, we reach a leaf node which tells us that our target function has a value of Short. Therefore by basically answering each of these multiple discrete value (Leave At) questions or yes/no questions, we can discover how long our commute might take.

13.2.4 Decision Trees as Rules

The example given in Figure 13.2 need not have been represented graphically as a tree but could also have been represented as rules (Figure 13.3). However the decision tree representation makes it easier to understand the problem and decision points.

```
if hour == 8am
    commute time = long
else if hour == 9am
    if accident == yes
       commute time = long
  else commute time = medium
else if hour == 10am
    if  road block ==yes
        commute time = long
  else  commute time = short
```

Figure 13.3 Decision Tree represented as Rules

13.3 Working of Decision Trees

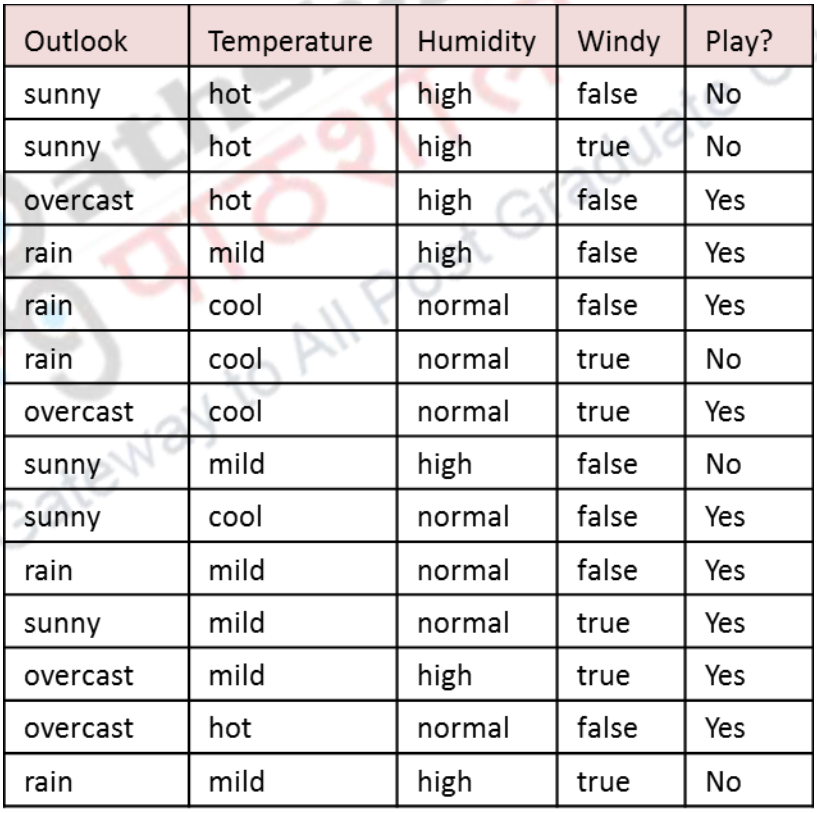
The decision tree rules partitions samples into mutually exclusive groups where each group is associated with a terminal node. This leaf node indicates the class assignment. The decision trees are a disjunctive normal form of the feature or attribute set. The structure of the decision tree in *disjunctive normal form* limits the form of a decision rule to conjunctions or adding of terms. It allows disjunction (or-ing) over a set of rules. At each non-terminal node the model identifies an attribute to be tested. This test splits attribute into mutually exclusive disjoint sets and this splitting continues until a one class (leaf node) is reached.

The path through the decision tree starts at the root and ends at a leaf. Each path represents a decision rule which essentially combines or ANDs all the decision tests along the path. For any one set of attribute values only one path will be followed. When there are separate paths that end at the same leaf or decision they are disjunctions or ORs of attribute values of each path. All paths in this case are essentially mutually exclusive.

13.4 Other Examples of Decision Trees

13.4.1 Weather Data: To Play or not to Play?

Let us consider an example of Play Tennis often used to explain many machine learning algorithms. Figure 13.4 shows a table containing 14 training samples, where four attributes namely *Outlook, Temperature, Humidity* and *Windy* indicate the weather conditions while *Play* indicates the target attribute. In this example given weather conditions, we need to find out whether to Play or not based on the classification model learnt from the training data.



Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Figure 13.4 Play Tennis Example

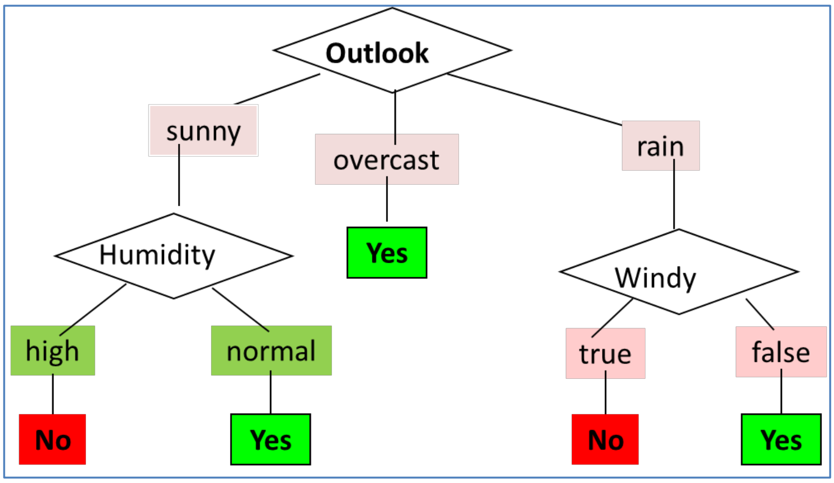


Figure 13.5 An Example Decision Tree for Play Tennis Example

Now let us discuss the decision tree (Figure 13.5) that corresponds to Play Tennis example table (Figure 13.4). Please note here that a number of decision trees can be constructed from the table. The methods used to construct a good decision tree will be explained in future modules.

Let us look at the values that each of the decision attributes can take. *Outlook* can take three values namely *sunny, overcast and rain, Temperature* can take three values namely *hot, mild, cool, Humidity* takes only two values namely *high and normal* and *Windy* again takes two values *true and false.* The target attribute *Play* is also Boolean and takes values of *Yes and No.*

For the present, let us understand the decision tree shown in Figure 13.5.

1. The root node here is the decision node *Outlook* which can take three values namely *sunny, overcast and rain* and hence the 14 examples are split into three disjoint subsets with 5 examples associated with sunny, 4 examples associated with overcast, and 5 examples associated with rain.
2. Now let us look at the values of the target attribute Play. We find that for the examples where the attribute Outlook has value as sunny and rainy, it is not possible to make the decision whether to Play or not.
3. However, for the 4 examples where the attribute Outlook has value as overcast it is possible to make the decision whether to Play or not based only (in this example Yes) with this information and other attributes need not be considered. This can be seen in the decision tree of the example.
4. Now for the remaining 9 examples we need to split further using decision nodes associated with other attributes. Now let us consider the decision node corresponding to the attribute Humidity. Now for the 5 examples corresponding to sunny, this decision node Humidity splits it into two with 3 examples for the value high and 2 examples for the value normal.
5. The ANDing of the attribute values sunny of attribute Outlook and values of attribute Humidity, the value of target attribute can be determined. Sunny and high gives a value of No for target attribute Play and sunny and normal gives a value of No for Play.
6. Now let us consider the decision node corresponding to the attribute Windy. Now for the 5 examples corresponding to rain, this decision node Windy

splits it into two with 3 examples for the value false and 2 examples for the value true.

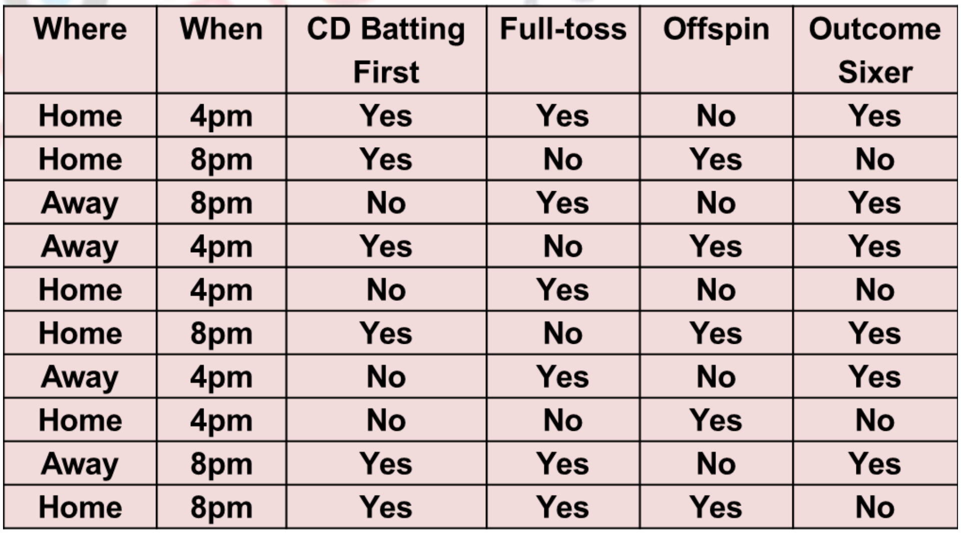
7. The ANDing of the attribute values rain of attribute Outlook and values of attribute Windy, the value of target attribute can be determined. Rain and false gives a value of Yes for target attribute Play and rain and true gives a value of No for Play.

Now we see that all the examples have been accounted for and the construction of the decision tree is over. Please note that the attribute Temperature was not used in the decision tree. The choice of which attribute to use as root node and at every level of the tree are very important and will be discussed in future modules.

13.4.2 Cricket Example

Let us next consider a table that gives training data to predict whether a six will be hit by *Chennai Delite* batsmen (Figure 13.6). The table shows available knowledge in the form of 5 Boolean attributes giving details about:

- Was the game at Home ground or Away?
- Was the starting time 4pm, or 8pm?
- Was *Chennai Delite* batting first?
- Was the opponent's ball was an off-spinner ?
- Was the opponent's ball was a full-toss?



Where	When	CD Batting First	Full-toss	Offspin	Outcome Sixer
Home	4pm	Yes	Yes	No	Yes
Home	8pm	Yes	No	Yes	No
Away	8pm	No	Yes	No	Yes
Away	4pm	Yes	No	Yes	Yes
Home	4pm	No	Yes	No	No
Home	8pm	Yes	No	Yes	Yes
Away	4pm	No	Yes	No	Yes
Home	4pm	No	No	Yes	No
Away	8pm	Yes	Yes	No	Yes
Home	8pm	Yes	Yes	Yes	No

Figure 13.6 Cricket Example

Figure 13.7 shows the path where we start at the root node, in this case *Chennai Delite* batting First. Next we check the starting time (When) of the match, then whether it is Full Toss, and whether it is Off spin. When we check these four decision nodes with values of CD batting first, 8 Pm start, ball is Full toss but not an off spinner, then the ball will go for a six. The basic issues here are which decision attribute to use as start node, which attribute to use next and when to stop.

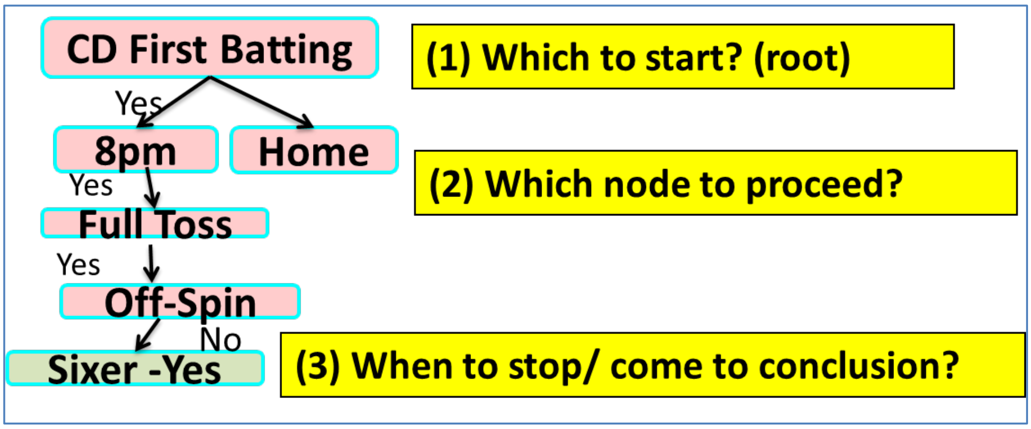

CD First Batting
Yes
8pm
Home
Yes
Full Toss
Yes
Off-Spin
No
Sixer -Yes
(1) Which to start? (root)
(2) Which node to proceed?
(3) When to stop/ come to conclusion?

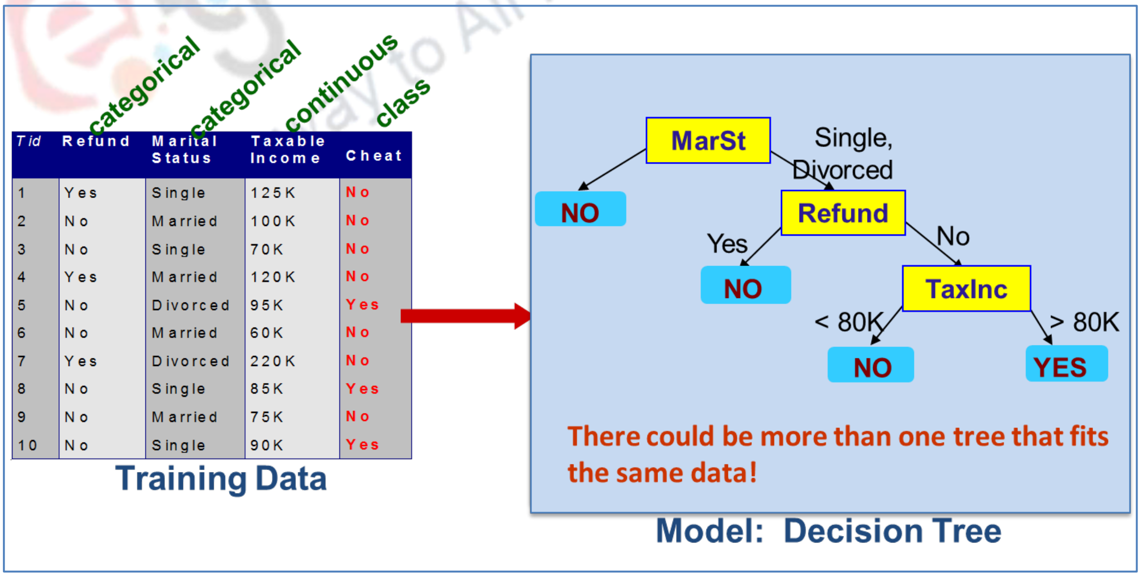
Figure 13.7 A Decision Path for Cricket Example

13.5 Building Decision Tree

The decision tree can be constructed in two ways - top-down tree construction and bottom-up tree construction. In the top-down tree construction, at the start, all training examples are at the root. We then recursively partition the examples by choosing one attribute each time. On the other hand, in the bottom-up tree pruning method we remove subtrees or branches, in a bottom-up manner, to improve the estimated accuracy on new cases.

13.6 Decision Tree Classification Task

Now let us discuss the use of the decision tree for the classification task using an example of finding about whether a person will be a cheat or not. The



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

There could be more than one tree that fits the same data!

Figure 13.8 Example Decision Tree for Cheat Example

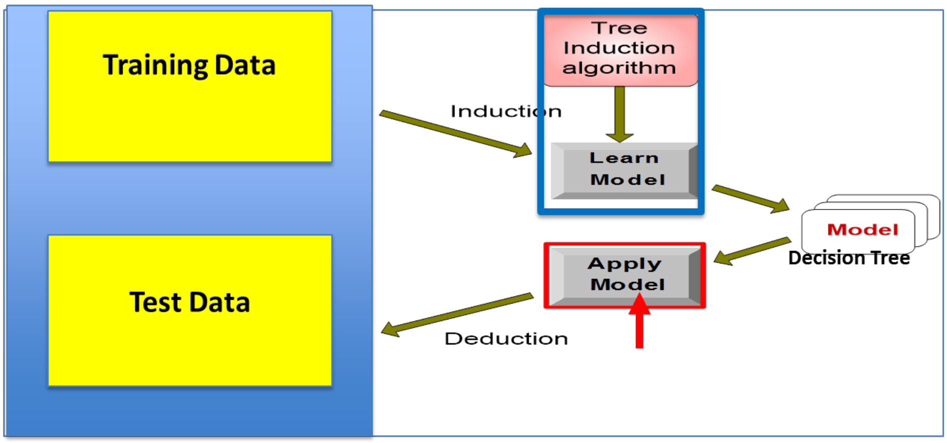
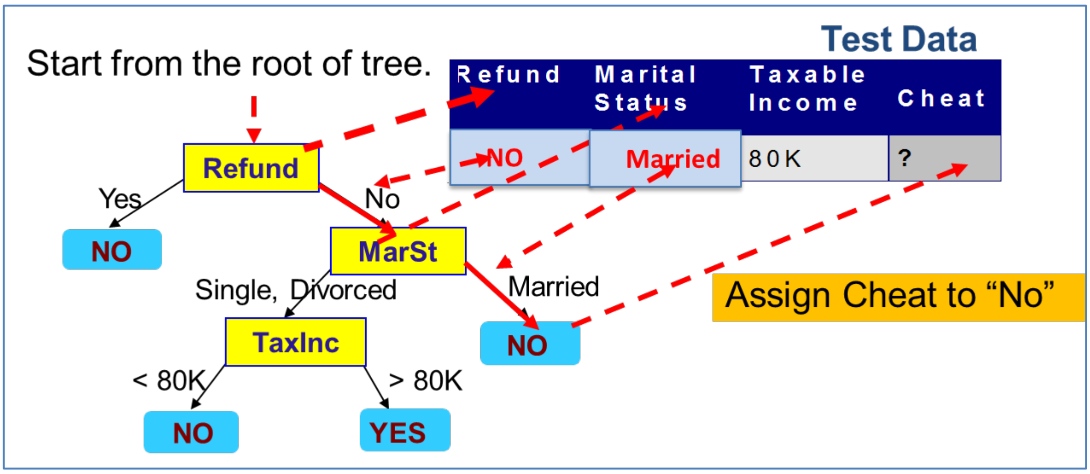


Figure 13.9 Classification with Decision Tree

training data here consists of 10 samples (Figure 13.8). This example has three decision attributes, Atttribute1 is categorical type Boolean attribute, Atttribute2 is categorical type three valued attribute, while Atttribute3 is continuous type attribute. In this example we are dealing with a continuous attribute. Continuous variables as attributes increase computational complexity, may result in prediction inaccuracy and can lead to overfitting of data. Generally continuous variables are converted into discrete intervals using “greater than or equal to” and “less than”.

The decision tree corresponding to the training set is shown in Figure 13.8. Figure 13.9 show how the induction learning uses the training samples to build the model and we apply the model learnt to deduce the class of the samples in the test set.

When we carry out deduction (Figure 13.10), we start at the root node of the decision tree in our case Refund, and for the example Refund value is n, so we follow that path and reach the decision node Marital Status, where for the example considered the value is married. We follow the path and end up with a target value of no. Therefore for the test example, the person is not a cheat.



Refund	Marital Status	Taxable Income	Cheat
NO	Married	80K	?

Figure 13.11 Decision Tree Path for Cheat Example

13.7 Evaluating Decision Trees

The main criterion for evaluating decision trees is the accuracy of predicting class label for new data. The next important aspect is scalability where the model learning and function prediction should be able to handle large data sets at a satisfactory speed. Another aspect is robustness where the performance should not affected by noisy or missing data. The decision tree method has an intuitive appeal since the results are easily understood. Further discussions on evaluating decision trees will be discussed in future modules.

Summary

- Discussed the key requirements for the use of decision trees
- Defined and explained decision trees
- Outlined the working of decision trees
- Described the use of decision tree models to test data for classification