Chapter 6

Decision Tree Learning

"Prediction is very difficult, especially if it's about the future."
— **Niels Bohr**

Decision Tree Learning is a widely used predictive model for supervised learning that spans over a number of practical applications in various areas. It is used for both classification and regression tasks. The decision tree model basically represents logical rules that predict the value of a target variable by inferring from data features. This chapter provides a keen insight into how to construct a decision tree and infer knowledge from the tree.

Learning Objectives

- Understand the structure of the decision tree
- Know about the fundamentals of Entropy
- Learn and understand popular univariate Decision Tree Induction algorithms such as ID3, C4.5, and multivariate decision tree algorithm such as CART
- Deal with continuous attributes using improved C4.5 algorithm
- Construct Classification and Regression Tree (CART) for classifying both categorical and continuous-valued target variables
- Construct regression trees where the target feature is a continuous-valued variable
- Understand the basics of validating and pruning of decision trees

6.1 INTRODUCTION TO DECISION TREE LEARNING MODEL

Decision tree learning model, one of the most popular supervised predictive learning models, classifies data instances with high accuracy and consistency. The model performs an *inductive inference* that reaches a general conclusion from observed examples. This model is variably used for solving complex classification applications.

Decision tree is a concept tree which summarizes the information contained in the training dataset in the form of a tree structure. Once the concept model is built, test data can be easily classified.

This model can be used to classify both categorical target variables and continuous-valued target variables. Given a training dataset X, this model computes a hypothesis function $f(X)$ as decision tree.

Inputs to the model are data instances or objects with a set of features or attributes which can be discrete or continuous and the output of the model is a decision tree which predicts or classifies the target class for the test data object.

In statistical terms, attributes or features are called as independent variables. The target feature or target class is called as response variable which indicates the category we need to predict on a test object.

The decision tree learning model generates a complete hypothesis space in the form of a tree structure with the given training dataset and allows us to search through the possible set of hypotheses which in fact would be a smaller decision tree as we walk through the tree. This kind of search bias is called as preference bias.

6.1.1 Structure of a Decision Tree

A decision tree has a structure that consists of a root node, internal nodes/decision nodes, branches, and terminal nodes/leaf nodes. The topmost node in the tree is the root node. Internal nodes are the test nodes and are also called as decision nodes. These nodes represent a choice or test of an input attribute and the outcome or outputs of the test condition are the branches emanating from this decision node. The branches are labelled as per the outcomes or output values of the test condition. Each branch represents a sub-tree or subsection of the entire tree. Every decision node is part of a path to a leaf node. The leaf nodes represent the labels or the outcome of a decision path. The labels of the leaf nodes are the different target classes a data instance can belong to.

Every path from root to a leaf node represents a logical rule that corresponds to a conjunction of test attributes and the whole tree represents a disjunction of these conjunctions. The decision tree model, in general, represents a collection of logical rules of classification in the form of a tree structure.

Decision networks, otherwise called as influence diagrams, have a directed graph structure with nodes and links. It is an extension of Bayesian belief networks that represents information about each node's current state, its possible actions, the possible outcome of those actions, and their utility. The concept of Bayesian Belief Network (BBN) is discussed in Chapter 9.

Figure 6.1 shows symbols that are used in this book to represent different nodes in the construction of a decision tree. A circle is used to represent a root node, a diamond symbol is used to represent a decision node or the internal nodes, and all leaf nodes are represented with a rectangle.

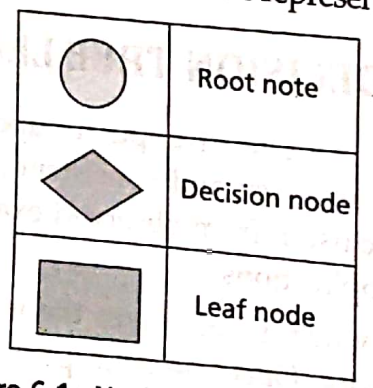


Figure 6.1: Nodes in a Decision Tree

A decision tree consists of two major procedures discussed below.

Building the Tree

Goal Construct a decision tree with the given training dataset. The tree is constructed in a top-down fashion. It starts from the root node. At every level of tree construction, we need to find the best split attribute or best decision node among all attributes. This process is recursive and continued until we end up in the last level of the tree or finding a leaf node which cannot be split further. The tree construction is complete when all the test conditions lead to a leaf node. The leaf node contains the target class or output of classification.

Output Decision tree representing the complete hypothesis space.

Knowledge Inference or Classification

Goal Given a test instance, infer to the target class it belongs to.

Classification Inferring the target class for the test instance or object is based on inductive inference on the constructed decision tree. In order to classify an object, we need to start traversing the tree from the root. We traverse as we evaluate the test condition on every decision node with the test object attribute value and walk to the branch corresponding to the test's outcome. This process is repeated until we end up in a leaf node which contains the target class of the test object.

Output Target label of the test instance.

Advantages of Decision Trees

1. Easy to model and interpret
2. Simple to understand
3. The input and output attributes can be discrete or continuous predictor variables.
4. Can model a high degree of nonlinearity in the relationship between the target variables and the predictor variables
5. Quick to train

Disadvantages of Decision Trees

Some of the issues that generally arise with a decision tree learning are that:

1. It is difficult to determine how deeply a decision tree can be grown or when to stop growing it.
2. If training data has errors or missing attribute values, then the decision tree constructed may become unstable or biased.
3. If the training data has continuous valued attributes, handling it is computationally complex and has to be discretized.
4. A complex decision tree may also be over-fitting with the training data.
5. Decision tree learning is not well suited for classifying multiple output classes.
6. Learning an optimal decision tree is also known to be NP-complete.

Example 6.1: How to draw a decision tree to predict a student's academic performance based on the given information such as class attendance, class assignments, home-work assignments, tests, participation in competitions or other events, group activities such as projects and presentations, etc.

Solution: The target feature is the student performance in the final examination whether he will pass or fail in the examination. The decision nodes are test nodes which check for conditions like 'What's the student's class attendance?', 'How did he perform in his class assignments?', 'Did he do his home assignments properly?' 'What about his assessment results?', 'Did he participate in competitions or other events?', 'What is the performance rating in group activities such as projects and presentations?'. Table 6.1 shows the attributes and set of values for each attribute.

Table 6.1: Attributes and Associated Values

Attributes	Values
Class attendance	Good, Average, Poor
Class assignments	Good, Moderate, Poor
Home-work assignments	Yes, No
Assessment	Good, Moderate, Poor
Participation in competitions or other events	Yes, No
Group activities such as projects and presentations	Yes, No
Exam Result	Pass, Fail

The leaf nodes represent the outcomes, that is, either 'pass', or 'fail'.

A decision tree would be constructed by following a set of if-else conditions which may or may not include all the attributes, and decision nodes outcomes are two or more than two. Hence, the tree is not a binary tree.

Note: *A decision tree is not always a binary tree. It is a tree which can have more than two branches.*

Example 6.2: Predict a student's academic performance of whether he will pass or fail based on the given information such as 'Assessment' and 'Assignment'. The following Table 6.2 shows the independent variables, Assessment and Assignment, and the target variable Exam Result with their values. Draw a binary decision tree.

Table 6.2: Attributes and Associated Values

Attributes	Values
Assessment	≥50, <50
Assignment	Yes, No
Exam Result	Pass, Fail

Solution: Consider the root node is 'Assessment'. If a student's marks are ≥50, the root node is branched to leaf node 'Pass' and if the assessment marks are <50, it is branched to another decision node. If the decision node in next level of the tree is 'Assignment' and if a student has submitted his assignment, the node branches to 'Pass' and if not submitted, the node branches to 'Fail'. Figure 6.2 depicts this rule.

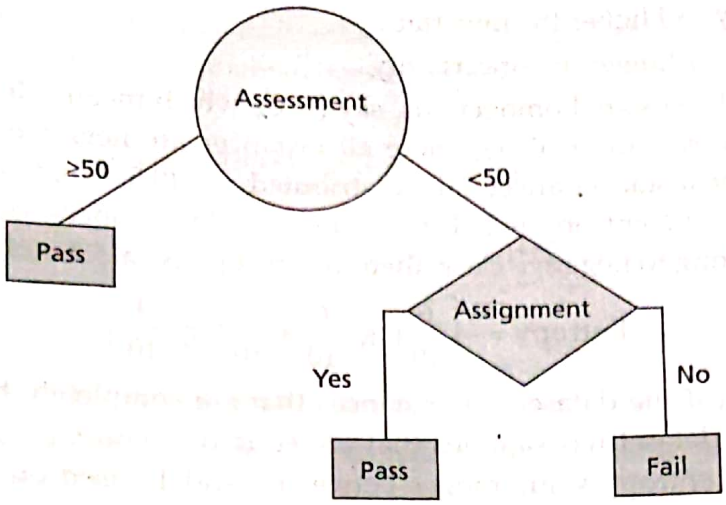


Figure 6.2: Illustration of a Decision Tree

This tree can be interpreted as a sequence of logical rules as follows:

if (Assessment ≥ 50) then 'Pass'

else if (Assessment < 50) then

if (Assignment == Yes) then 'Pass'

else if (Assignment == No) then 'Fail'

Now, if a test instance is given, such as a student has scored 42 marks in his assessment and has not submitted his assignment, then it is predicted with the decision tree that his exam result is 'Fail'.

Many algorithms exist which will be studied for constructing decision trees in the sections below.

6.1.2 Fundamentals of Entropy

Given the training dataset with a set of attributes or features, the decision tree is constructed by finding the attribute or feature that best describes the target class for the given test instances. The best split feature is the one which contains more information about how to split the dataset among all features so that the target class is accurately identified for the test instances. In other words, the best split attribute is more informative to split the dataset into sub datasets and this process is continued until the stopping criterion is reached. This splitting should be pure at every stage of selecting the best feature.

The best feature is selected based on the amount of information among the features which are basically calculated on probabilities. Quantifying information is closely related to information theory. In the field of information theory, the features are quantified by a measure called Shannon Entropy which is calculated based on the probability distribution of the events.

Entropy is the amount of uncertainty or randomness in the outcome of a random variable or an event. Moreover, entropy describes about the homogeneity of the data instances. The best feature is selected based on the entropy value. For example, when a coin is flipped, head or tail are the two outcomes, hence its entropy is lower when compared to rolling a dice which has got six outcomes. Hence, the interpretation is,

Higher the entropy → Higher the uncertainty

Lower the entropy → Lower the uncertainty

Similarly, if all instances are homogenous, say (1, 0), which means all instances belong to the same class (here it is positive) or (0, 1) where all instances are negative, then the entropy is 0. On the other hand, if the instances are equally distributed, say (0.5, 0.5), which means 50% positive and 50% negative, then the entropy is 1. If there are 10 data instances, out of which 6 belong to positive class and 4 belong to negative class, then the entropy is calculated as shown in Eq. (6.1),

$$\text{Entropy} = -\left[\frac{6}{10}\log_2\frac{6}{10} + \frac{4}{10}\log_2\frac{4}{10}\right] \tag{6.1}$$

It is concluded that if the dataset has instances that are completely homogeneous, then the entropy is 0 and if the dataset has samples that are equally divided (i.e., 50% – 50%), it has an entropy of 1. Thus, the entropy value ranges between 0 and 1 based on the randomness of the samples in the dataset. If the entropy is 0, then the split is pure which means that all samples in the set will partition into one class or category. But if the entropy is 1, the split is impure and the distribution of the samples is more random. The stopping criterion is based on the entropy value.

Let P be the probability distribution of data instances from 1 to n as shown in Eq. (6.2).

$$\text{So, } P = P_1 \ldots\ldots P_n \tag{6.2}$$

Entropy of P is the information measure of this probability distribution given in Eq. (6.3),

$$\begin{aligned}\text{Entropy\_Info}(P) &= \text{Entropy\_Info}(P_1 \ldots P_n) \\ &= -(P_1 \log_2(P_1) + P_2 \log_2(P_2) + \ldots\ldots + P_n \log_2(P_n))\end{aligned} \tag{6.3}$$

where, P_1 is the probability of data instances classified as class 1 and P_2 is the probability of data instances classified as class 2 and so on.

P_1 = |No of data instances belonging to class 1| / |Total no of data instances in the training dataset|

Entropy_Info(P) can be computed as shown in Eq. (6.4).

Thus,

$$\text{Entropy\_Info}(6, 4) \text{ is calculated as } -\left[\frac{6}{10}\log_2\frac{6}{10} + \frac{4}{10}\log_2\frac{4}{10}\right] \tag{6.4}$$

Mathematically, entropy is defined in Eq. (6.5) as:

$$\text{Entropy\_Info}(X) = \sum_{x \in \text{values}(X)} Pr[X = x] \cdot \log_2 \frac{1}{Pr[X = x]} \tag{6.5}$$

$Pr[X = x]$ is the probability of a random variable X with a possible outcome x.

Note: $\log_2 \frac{1}{Pr[X = x]} = -\log_2(Pr[X = x])$

Algorithm 6.1: General Algorithm for Decision Trees

1. Find the best attribute from the training dataset using an attribute selection measure and place it at the root of the tree.

(Continued)

2. Split the training dataset into subsets based on the outcomes of the test attribute and each subset in a branch contains the data instances or tuples with the same value for the selected test attribute.
3. Repeat step 1 and step 2 on each subset until we end up in leaf nodes in all the branches of the tree.
4. This splitting process is recursive until the stopping criterion is reached.

Stopping Criteria

The following are some of the common stopping conditions:

1. The data instances are homogenous which means all belong to the same class C_i and hence its entropy is 0.
2. A node with some defined minimum number of data instances becomes a leaf (Number of data instances in a node is between 0.25 and 1.00% of the full training dataset).
3. The maximum tree depth is reached, so further splitting is not done and the node becomes a leaf node.

6.2 DECISION TREE INDUCTION ALGORITHMS

There are many decision tree algorithms, such as ID3, C4.5, CART, CHAID, QUEST, GUIDE, CRUISE, and CTREE, that are used for classification in real-time environment. The most commonly used decision tree algorithms are ID3 (Iterative Dichotomizer 3), developed by J.R Quinlan in 1986, and C4.5 is an advancement of ID3 presented by the same author in 1993. CART, that stands for Classification and Regression Trees, is another algorithm which was developed by Breiman et al. in 1984.

The accuracy of the tree constructed depends upon the selection of the best split attribute. Different algorithms are used for building decision trees which use different measures to decide on the splitting criterion. Algorithms such as ID3, C4.5 and CART are popular algorithms used in the construction of decision trees. The algorithm ID3 uses 'Information Gain' as the splitting criterion whereas the algorithm C4.5 uses 'Gain Ratio' as the splitting criterion. The CART algorithm is popularly used for classifying both categorical and continuous-valued target variables. CART uses GINI Index to construct a decision tree.

Decision trees constructed using ID3 and C4.5 are also called as *univariate decision trees* which consider only one feature/attribute to split at each decision node whereas decision trees constructed using CART algorithm are *multivariate decision trees* which consider a conjunction of univariate splits. The details about univariate and multivariate data has been discussed in Chapter 2.

6.2.1 ID3 Tree Construction

ID3 is a supervised learning algorithm which uses a training dataset with labels and constructs a decision tree. ID3 is an example of univariate decision trees as it considers only one feature at each decision node. This leads to axis-aligned splits. The tree is then used to classify the future test instances. It constructs the tree using a greedy approach in a top-down fashion by identifying the best attribute at each level of the tree.

ID3 works well if the attributes or features are considered as discrete/categorical values. If some attributes are continuous, then we have to partition attributes or features to be decretized or nominal attributes or features.

The algorithm builds the tree using a purity measure called 'Information Gain' with the given training data instances and then uses the constructed tree to classify the test data. It is applied for training set with only nominal attributes or categorical attributes and with no missing values for classification. ID3 works well for a large dataset. If the dataset is small, overfitting may occur. Moreover, it is not accurate if the dataset has missing attribute values.

No pruning is done during or after construction of the tree and it is prone to outliers. C4.5 and CART can handle both categorical attributes and continuous attributes. Both C4.5 and CART can also handle missing values, but C4.5 is prone to outliers whereas CART can handle outliers as well.

Algorithm 6.2: Procedure to Construct a Decision Tree using ID3

1. Compute Entropy_Info Eq. (6.8) for the whole training dataset based on the target attribute.
2. Compute Entropy_Info Eq. (6.9) and Information_Gain Eq. (6.10) for each of the attribute in the training dataset.
3. Choose the attribute for which entropy is minimum and therefore the gain is maximum as the best split attribute.
4. The best split attribute is placed as the root node.
5. The root node is branched into subtrees with each subtree as an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into subsets.
6. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.

Note: *We stop branching a node if entropy is 0.*

The best split attribute at every iteration is the attribute with the highest information gain.

Definitions

Let T be the training dataset.

Let A be the set of attributes $A = \{A_1, A_2, A_3, \ldots\ldots A_n\}$.

Let m be the number of classes in the training dataset.

Let P_i be the probability that a data instance or a tuple 'd' belongs to class C_i.

It is calculated as,

P_i = Total no of data instances that belongs to class C_i in T/Total no of tuples in the training set T (6.6)

Mathematically, it is represented as shown in Eq. (6.7).

$$P_i = \frac{|d_{C_i}|}{|T|} \tag{6.7}$$

Expected information or Entropy needed to classify a data instance d' in T is denoted as Entropy_Info(T) given in Eq. (6.8).

$$\text{Entropy\_Info}(T) = -\sum_{i=1}^{m} P_i \log_2 P_i \tag{6.8}$$

Entropy of every attribute denoted as Entropy_Info(T, A) is shown in Eq. (6.9) as:

$$\text{Entropy\_Info}(T, A) = \sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \text{Entropy\_Info}\ (A_i) \tag{6.9}$$

where, the attribute A has got 'v' distinct values $\{a_1, a_2, \ldots a_v\}$, $|A_i|$ is the number of instances for distinct value 'i' in attribute A, and Entropy_Info (A_i) is the entropy for that set of instances.

Information_Gain is a metric that measures how much information is gained by branching on an attribute A. In other words, it measures the reduction in impurity in an arbitrary subset of data. It is calculated as given in Eq. (6.10):

$$\text{Information\_Gain}(A) = \text{Entropy\_Info}(T) - \text{Entropy\_Info}(T, A) \tag{6.10}$$

It can be noted that as entropy increases, information gain decreases. They are inversely proportional to each other.

Scan for *'Additional Examples'*

Example 6.3: Assess a student's performance during his course of study and predict whether a student will get a job offer or not in his final year of the course. The training dataset T consists of 10 data instances with attributes such as 'CGPA', 'Interactiveness', 'Practical Knowledge' and 'Communication Skills' as shown in Table 6.3. The target class attribute is the 'Job Offer'.

Table 6.3: Training Dataset T

S.No.	CGPA	Interactiveness	Practical Knowledge	Communication Skills	Job Offer
1.	≥9	Yes	Very good	Good	Yes
2.	≥8	No	Good	Moderate	Yes
3.	≥9	No	Average	Poor	No
4.	<8	No	Average	Good	No
5.	≥8	Yes	Good	Moderate	Yes
6.	≥9	Yes	Good	Moderate	Yes
7.	<8	Yes	Good	Poor	No
8.	≥9	No	Very good	Good	Yes
9.	≥8	Yes	Good	Good	Yes
10.	≥8	Yes	Average	Good	Yes

Solution:

Step 1:

Calculate the Entropy for the target class 'Job Offer'.

Entropy_Info(Target Attribute = Job Offer) = Entropy_Info(7, 3) =

$$= -\left[\frac{7}{10}\log_2\frac{7}{10} + \frac{3}{10}\log_2\frac{3}{10}\right] = -(-0.3599 + -0.5208) = 0.8807$$

Iteration 1:

Step 2:

Calculate the Entropy_Info and Gain(Information_Gain) for each of the attribute in the training dataset.

Table 6.4 shows the number of data instances classified with Job Offer as Yes or No for the attribute CGPA.

Table 6.4: Entropy Information for CGPA

CGPA	Job Offer = Yes	Job Offer = No	Total	Entropy
≥9	3	1	4	
≥8	4	0	4	0
<8	0	2	2	0

Entropy_Info(T, CGPA)

$$= \frac{4}{10}\left[-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right] + \frac{4}{10}\left[-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$

$$= \frac{4}{10}(0.3111 + 0.4997) + 0 + 0$$

$$= 0.3243$$

$$\text{Gain (CGPA)} = 0.8807 - 0.3243$$

$$= 0.5564$$

Table 6.5 shows the number of data instances classified with Job Offer as Yes or No for the attribute Interactiveness.

Table 6.5: Entropy Information for Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No	Total	Entropy
YES	5	1	6	
NO	2	2	4	

$$\text{Entropy\_Info}(T, \text{Interactiveness}) = \frac{6}{10}\left[-\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6}\right] + \frac{4}{10}\left[-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right]$$

$$= \frac{6}{10}(0.2191 + 0.4306) + \frac{4}{10}(0.4997 + 0.4997)$$

$$= 0.3898 + 0.3998 = 0.7896$$

$$\text{Gain(Interactiveness)} = 0.8807 - 0.7896$$

$$= 0.0911$$

Table 6.6 shows the number of data instances classified with Job Offer as Yes or No for the attribute Practical Knowledge.

Table 6.6: Entropy Information for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No	Total	Entropy
Very Good	2	0	2	0
Average	1	2	3	
Good	4	1	5	

Entropy_Info(T, Practical Knowledge)

$$= \frac{2}{10}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{3}{10}\left[-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right] + \frac{5}{10}\left[-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right]$$

$$= \frac{2}{10}(0) + \frac{3}{10}(0.5280 + 0.3897) + \frac{5}{10}(0.2574 + 0.4641)$$

$$= 0 + 0.2753 + 0.3608$$

$$= 0.6361$$

$$\text{Gain(Practical Knowledge)} = 0.8807 - 0.6361$$

$$= 0.2446$$

Table 6.7 shows the number of data instances classified with Job Offer as Yes or No for the attribute Communication Skills.

Table 6.7: Entropy Information for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No	Total
Good	4	1	5
Moderate	3	0	3
Poor	0	2	2

Entropy_Info(T, Communication Skills)

$$= \frac{5}{10}\left[-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right] + \frac{3}{10}\left[-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$

$$= \frac{5}{10}(0.5280 + 0.3897) + \frac{3}{10}(0) + \frac{2}{10}(0)$$

$$= 0.3609$$

$$\text{Gain(Communication Skills)} = 0.8813 - 0.36096$$

$$= 0.5203$$

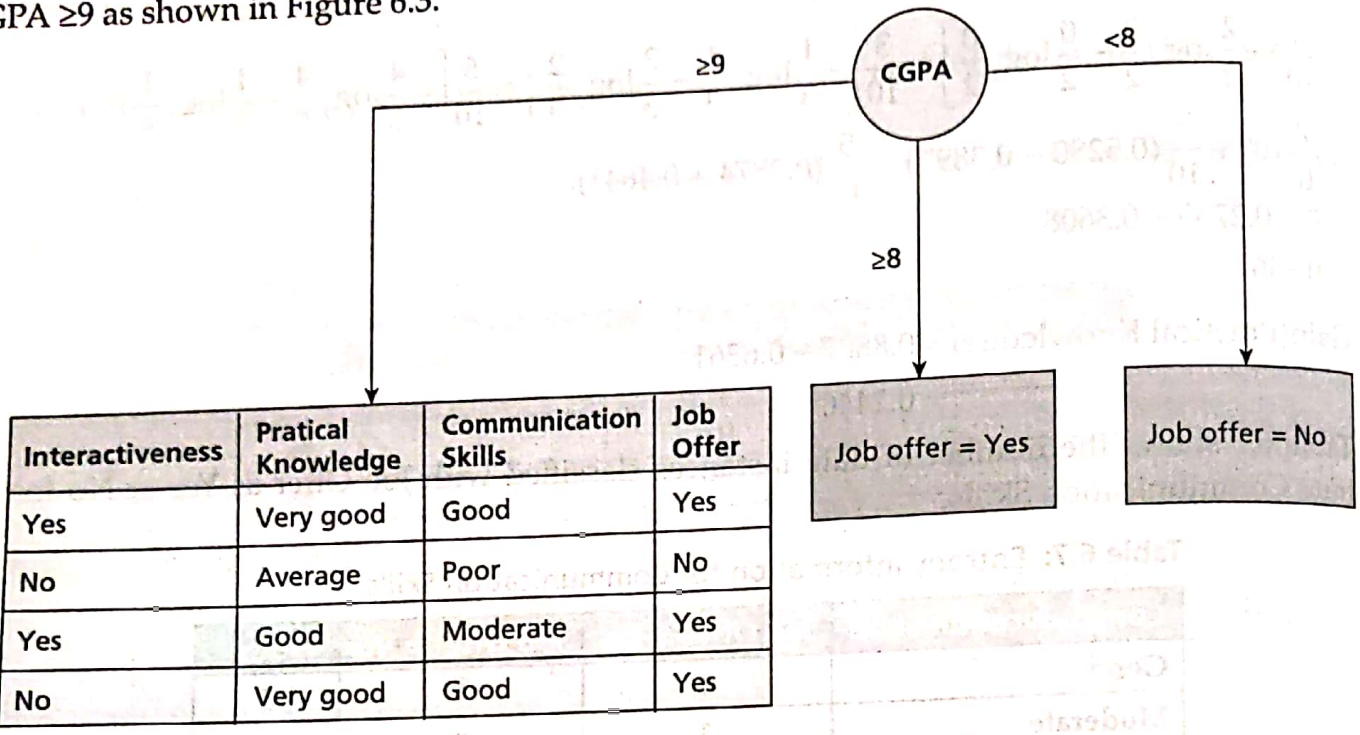
The Gain calculated for all the attributes is shown in Table 6.8:

Table 6.8: Gain

Attributes	Gain
CGPA	0.5564
Interactiveness	0.0911
Practical Knowledge	0.2246
Communication Skills	0.5203

Step 3: From Table 6.8, choose the attribute for which entropy is minimum and therefore the gain is maximum as the best split attribute.

The best split attribute is CGPA since it has the maximum gain. So, we choose CGPA as the root node. There are three distinct values for CGPA with outcomes ≥9, ≥8 and <8. The entropy value is 0 for ≥8 and <8 with all instances classified as Job Offer = Yes for ≥8 and Job Offer = No for <8. Hence, both ≥8 and <8 end up in a leaf node. The tree grows with the subset of instances with CGPA ≥9 as shown in Figure 6.3.



Interactiveness	Pratical Knowledge	Communication Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Figure 6.3: Decision Tree After Iteration 1

Now, continue the same process for the subset of data instances branched with CGPA ≥ 9.

Iteration 2:

In this iteration, the same process of computing the Entropy_Info and Gain are repeated with the subset of training set. The subset consists of 4 data instances as shown in the above Figure 6.3.

$$\text{Entropy\_Info}(T) = \text{Entropy\_Info}(3, 1) = -\left[\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right]$$
$$= -(-0.3111 + -0.4997)$$
$$= 0.8108$$

$$\text{Entropy\_Info}(T, \text{Interactiveness}) = \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{2}{4}\left[-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right]$$
$$= 0 + 0.4997$$

$$\text{Gain(Interactiveness)} = 0.8108 - 0.4997$$
$$= 0.3111$$

Entropy_Info(T, Practical Knowledge)

$$= \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right]$$
$$= 0$$

Gain(Practical Knowledge) = 0.8108

Entropy_Info(T, Communication Skills)

$$= \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right]$$

$$= 0$$

Gain(Communication Skills) = 0.8108

The gain calculated for all the attributes is shown in Table 6.9.

Table 6.9: Total Gain

Attributes	Gain
Interactiveness	0.3111
Practical Knowledge	**0.8108**
Communication Skills	**0.8108**

Here, both the attributes 'Practical Knowledge' and 'Communication Skills' have the same Gain. So, we can either construct the decision tree using 'Practical Knowledge' or 'Communication Skills'. The final decision tree is shown in Figure 6.4.

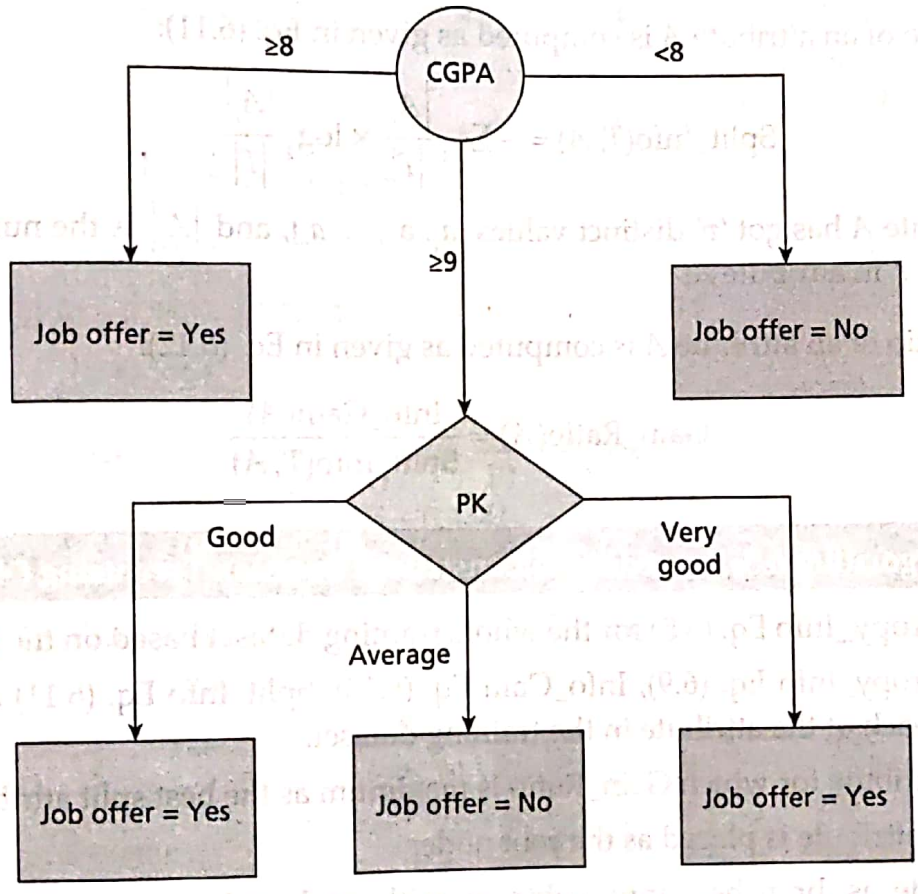


Figure 6.4: Final Decision Tree

6.2.2 C4.5 Construction

C4.5 is an improvement over ID3. C4.5 works with continuous and discrete attributes and missing values, and it also supports post-pruning. C5.0 is the successor of C4.5 and is more efficient and used for building smaller decision trees. C4.5 works with missing values by marking as '?', but these missing attribute values are not considered in the calculations.

The algorithm C4.5 is based on Occam's Razor which says that given two correct solutions, the simpler solution has to be chosen. Moreover, the algorithm requires a larger training set for better accuracy. It uses Gain Ratio as a measure during the construction of decision trees. ID3 is more biased towards attributes with larger values. For example, if there is an attribute called 'Register No' for students it would be unique for every student and will have distinct value for every data instance resulting in more values for the attribute. Hence, every instance belongs to a category and would have higher Information Gain than other attributes. To overcome this bias issue, C4.5 uses a purity measure Gain ratio to identify the best split attribute. In C4.5 algorithm, the Information Gain measure used in ID3 algorithm is normalized by computing another factor called Split_Info. This normalized information gain of an attribute called as Gain_Ratio is computed by the ratio of the calculated Split_Info and Information Gain of each attribute. Then, the attribute with the highest normalized information gain, that is, highest gain ratio is used as the splitting criteria.

As an example, we will choose the same training dataset shown in Table 6.3 to construct a decision tree using the C4.5 algorithm.

Given a Training dataset T,

The Split_Info of an attribute A is computed as given in Eq. (6.11):

$$\text{Split\_Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|} \tag{6.11}$$

where, the attribute A has got 'v' distinct values $\{a_1, a_2, \ldots. a_v\}$, and $|A_i|$ is the number of instances for distinct value 'i' in attribute A.

The Gain_Ratio of an attribute A is computed as given in Eq. (6.12):

$$\text{Gain\_Ratio}(A) = \frac{\text{Info\_Gain}(A)}{\text{Split\_Info}(T, A)} \tag{6.12}$$

Algorithm 6.3: Procedure to Construct a Decision Tree using C4.5

1. Compute Entropy_Info Eq. (6.8) for the whole training dataset based on the target attribute.
2. Compute Entropy_Info Eq. (6.9), Info_Gain Eq. (6.10), Split_Info Eq. (6.11) and Gain_Ratio Eq. (6.12) for each of the attribute in the training dataset.
3. Choose the attribute for which Gain_Ratio is maximum as the best split attribute.
4. The best split attribute is placed as the root node.
5. The root node is branched into subtrees with each subtree as an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into subsets.
6. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.

Example 6.4: Make use of Information Gain of the attributes which are calculated in ID3 algorithm in Example 6.3 to construct a decision tree using C4.5.

Solution:

Iteration 1:

Step 1: Calculate the Class_Entropy for the target class 'Job Offer'.

Entropy_Info(Target Attribute = Job Offer) = Entropy_Info(7, 3) =

$$= -\left[\frac{7}{10}\log_2\frac{7}{10} + \frac{3}{10}\log_2\frac{3}{10}\right]$$

$$= (-0.3599 + -0.5208)$$

$$= 0.8807$$

Step 2: Calculate the Entropy_Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attribute in the training dataset.

CGPA:

$$\text{Entropy Info}(T, \text{CGPA}) = \frac{4}{10}\left[-\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}\right] + \frac{4}{10}\left[-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$

$$= \frac{4}{10}(0.3111 + 0.4997) + 0 + 0$$

$$= 0.3243$$

$$\text{Gain(CGPA)} = 0.8807 - 0.3243$$

$$= 0.5564$$

$$\text{Split\_Info}(T, \text{CGPA}) = -\frac{4}{10}\log_2\frac{4}{10} - \frac{4}{10}\log_2\frac{4}{10} - \frac{2}{10}\log_2\frac{2}{10}$$

$$= 0.5285 + 0.5285 + 0.4641$$

$$= 1.5211$$

$$\text{Gain Ratio(CGPA)} = (\text{Gain(CGPA)})/(\text{Split\_Info}(T, \text{CGPA}))$$

$$= \frac{0.5564}{1.5211} = 0.3658$$

Interactiveness:

$$\text{Entropy Info}(T, \text{Interactiveness}) = \frac{6}{10}\left[-\frac{5}{6}\log_2\frac{5}{6} - \frac{1}{6}\log_2\frac{1}{6}\right] + \frac{4}{10}\left[-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}\right]$$

$$= \frac{6}{10}(0.2191 + 0.4306) + \frac{4}{10}(0.4997 + 0.4997)$$

$$= 0.3898 + 0.3998 = 0.7896$$

$$\text{Gain(Interactiveness)} = 0.8807 - 0.7896 = 0.0911$$

$$\text{Gain(Interactiveness)} = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10} = 0.9704$$

$$\text{Gain\_Ratio(Interactiveness)} = \frac{\text{Gain(Interactiveness)}}{\text{Split\_Info}(T, \text{Interactiveness})}$$
$$= \frac{0.0911}{0.9704}$$
$$= 0.0939$$

Practical Knowledge:

$$\text{Entropy\_Info}(T, \text{Practical Knowledge}) = \frac{2}{10}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{3}{10}\left[-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right] + \frac{5}{10}\left[-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right]$$
$$= \frac{2}{10}(0) + \frac{3}{10}(0.5280 + 0.3897) + \frac{5}{10}(0.2574 + 0.4641)$$
$$= 0 + 0.2753 + 0.3608 = 0.6361$$

$$\text{Gain(Practical Knowledge)} = 0.8807 - 0.6361$$
$$= 0.2448$$

$$\text{Split\_Info}(T, \text{Practical Knowledge}) = -\frac{2}{10}\log_2\frac{2}{10} - \frac{5}{10}\log_2\frac{5}{10} - \frac{3}{10}\log_2\frac{3}{10}$$
$$= 1.4853$$

$$\text{Gain\_Ratio(Practical Knowledge)} = \frac{\text{Gain(Practical Knowledge)}}{\text{Split\_Info}(T, \text{Practical Knowledge})}$$
$$= \frac{0.2448}{1.4853}$$
$$= 0.1648$$

Communication Skills:

$$\text{Entropy\_Info}(T, \text{Communication Skills}) = \frac{5}{10}\left[-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}\right] + \frac{3}{10}\left[-\frac{3}{3}\log_2\frac{3}{3} - \frac{0}{3}\log_2\frac{0}{3}\right] + \frac{2}{10}\left[-\frac{0}{2}\log_2\frac{0}{2} - \frac{2}{2}\log_2\frac{2}{2}\right]$$
$$= \frac{5}{10}(0.5280 + 0.3897) + \frac{3}{10}(0) + \frac{2}{10}(0)$$
$$= 0.3609$$

$$\text{Gain(Communication Skills)} = 0.8813 - 0.36096$$
$$= 0.5202$$

$$\text{Split\_Info}(T, \text{Communication Skills}) = -\frac{5}{10}\log_2\frac{5}{10} - \frac{3}{10}\log_2\frac{3}{10} - \frac{2}{10}\log_2\frac{2}{10}$$
$$= 1.4853$$

$$\text{Gain\_Ratio(Communication Skills)} = \frac{\text{Gain(Communication Skills)}}{\text{Split\_Info}(T, \text{Communication Skills})}$$
$$= \frac{0.5202}{1.4853} = 0.3502$$

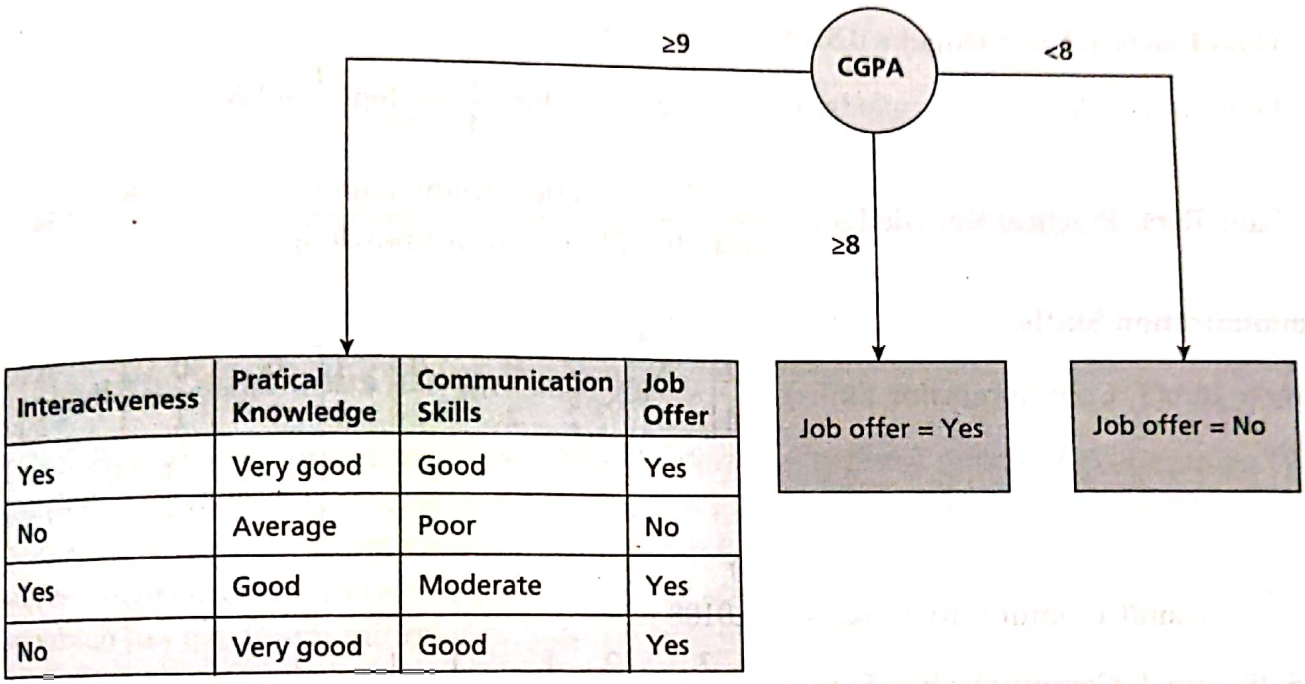
Table 6.10 shows the Gain_Ratio computed for all the attributes.

Table 6.10: Gain_Ratio

Attribute	Gain Ratio
CGPA	0.3658
INTERACTIVENESS	0.0939
PRACTICAL KNOWLEDGE	0.1648
COMMUNICATION SKILLS	0.3502

Step 3: Choose the attribute for which Gain_Ratio is maximum as the best split attribute.

From Table 6.10, we can see that CGPA has highest gain ratio and it is selected as the best split attribute. We can construct the decision tree placing CGPA as the root node shown in Figure 6.5. The training dataset is split into subsets with 4 data instances.



Interactiveness	Pratical Knowledge	Communication Skills	Job Offer
Yes	Very good	Good	Yes
No	Average	Poor	No
Yes	Good	Moderate	Yes
No	Very good	Good	Yes

Figure 6.5: Decision Tree after Iteration 1

Iteration 2:

Total Samples: 4

Repeat the same process for this resultant dataset with 4 data instances.

Job Offer has 3 instances as Yes and 1 instance as No.

$$\text{Entropy\_Info(Target Class = Job Offer)} = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4}$$
$$= 0.3112 + 0.5$$
$$= 0.8112$$

Interactiveness:

$$\text{Entropy\_Info}(T, \text{Interactiveness}) = \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{2}{4}\left[-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}\right]$$
$$= 0 + 0.4997$$
$$\text{Gain(Interactiveness)} = 0.8108 - 0.4997 = 0.3111$$

$$\text{Split\_Info}(T, \text{Interactiveness}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} = 0.5 + 0.5 = 1$$

$$\text{Gain\_Ratio}(\text{Interactiveness}) = \frac{\text{Gain}(\text{Interactiveness})}{\text{Split\_Info}(T, \text{Interactiveness})}$$

$$= \frac{0.3112}{1} = 0.3112$$

Practical Knowledge:

$$\text{Entropy\_Info}(T, \text{Practical Knowledge}) = \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right]$$

$$= 0$$

$$\text{Gain}(\text{Practical Knowledge}) = 0.8108$$

$$\text{Split\_Info}(T, \text{Practical Knowledge}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} = 1.5$$

$$\text{Gain\_Ratio}(\text{Practical Knowledge}) = \frac{\text{Gain}(\text{Practical Knowledge})}{\text{Split\_Info}(T, \text{Practical Knowledge})} = \frac{0.8108}{1.5} = 0.5408$$

Communication Skills:

$$\text{Entropy\_Info}(T, \text{Communication Skills}) = \frac{2}{4}\left[-\frac{2}{2}\log_2\frac{2}{2} - \frac{0}{2}\log_2\frac{0}{2}\right] + \frac{1}{4}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{1}{4}\left[-\frac{1}{1}\log_2\frac{1}{1} - \frac{0}{1}\log_2\frac{0}{1}\right]$$

$$= 0$$

$$\text{Gain}(\text{Communication Skills}) = 0.8108$$

$$\text{Split\_Info}(T, \text{Communication Skills}) = -\frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} = 1.5$$

$$\text{Gain\_Ratio}(\text{Communication Skills}) = \frac{\text{Gain}(\text{Practical Knowledge})}{\text{Split\_Info}(T, \text{Practical Knowledge})} = \frac{0.8108}{1.5} = 0.5408$$

Table 6.11 shows the Gain_Ratio computed for all the attributes.

Table 6.11: Gain-Ratio

Attributes	Gain_Ratio
Interactiveness	0.3112
Practical Knowledge	0.5408
Communication Skills	0.5408

Both 'Practical Knowledge' and 'Communication Skills' have the highest gain ratio. So, the best splitting attribute can either be 'Practical Knowledge' or 'Communication Skills', and therefore, the split can be based on any one of these.

Here, we split based on 'Practical Knowledge'. The final decision tree is shown in Figure 6.6.

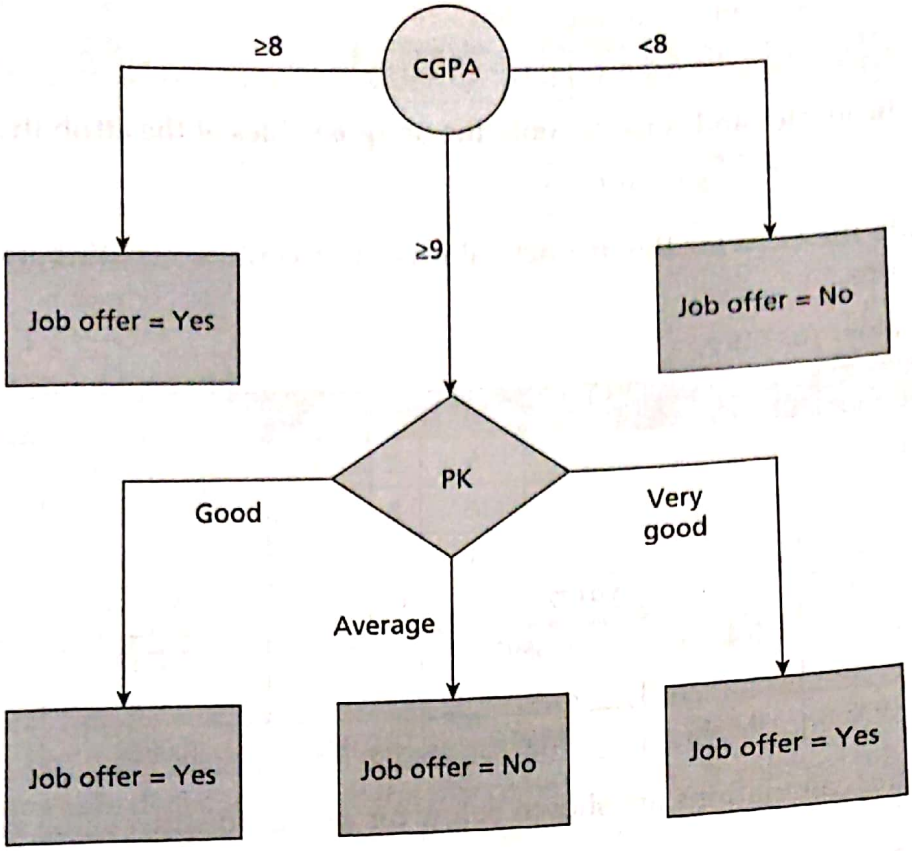


Figure 6.6: Final Decision Tree

Dealing with Continuous Attributes in C4.5

The C4.5 algorithm is further improved by considering attributes which are continuous, and a continuous attribute is discretized by finding a split point or threshold. When an attribute '*A*' has numerical values which are continuous, a threshold or best split point '*s*' is found such that the set of values is categorized into two sets such as $A < s$ and $A \geq s$. The best split point is the attribute value which has maximum information gain for that attribute.

Now, let us consider the set of continuous values for the attribute CGPA in the sample dataset as shown in Table 6.12.

Table 6.12: Sample Dataset

S.No.	CGPA	Job Offer
1.	9.5	Yes
2.	8.2	Yes
3.	9.1	No
4.	6.8	No
5.	8.5	Yes
6.	9.5	Yes
7.	7.9	No
8.	9.1	Yes
9.	8.8	Yes
10.	8.8	Yes

First, sort the values in an ascending order.

6.8	7.9	8.2	8.5	8.8	8.8	9.1	9.1	9.5	9.5

Remove the duplicates and consider only the unique values of the attribute.

6.8	7.9	8.2	8.5	8.8	8.8	9.1	9.5

Now, compute the Gain for the distinct values of this continuous attribute. Table 6.13 shows the computed values.

Table 6.13: Gain Values for CGPA

	6.8		7.9		8.2		8.5		8.8		9.1		9.5	
Range	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>
Yes	0	7	0	7	1	6	2	5	4	3	5	2	7	0
No	1	2	2	1	2	1	2	1	2	1	3	0	3	0
Entropy	0	0.7637	0	0.5433	0.9177	0.5913	1	0.6497	0.9177	0.8108	0.9538	0	0.8808	0
Entropy_Info (S, T)	0.6873		0.4346		0.6892		0.7898		0.8749		0.7630		0.8808	
Gain	0.1935		**0.4462**		0.1916		0.091		0.0059		0.1178		0	

For a sample, the calculations are shown below for a single distinct value say, CGPA ∈ 6.8.

$$\text{Entropy\_Info}(T, \text{Job\_Offer}) = -\left[\frac{7}{10}\log_2\frac{7}{10} + \frac{3}{10}\log_2\frac{3}{10}\right]$$
$$= -(-0.3599 + -0.5209)$$
$$= 0.8808$$

$$\text{Entropy}(7, 2) = -\left[\frac{7}{9}\log_2\frac{7}{9} + \frac{2}{9}\log_2\frac{2}{9}\right]$$
$$= -(-0.2818 + -0.4819)$$
$$= 0.7637$$

$$\text{Entropy\_Info}(T, \text{CGPA} \in 6.8) = \frac{1}{10} \times \text{Entropy}(0, 1) + \frac{9}{10}\text{Entropy}(7, 2)$$
$$= \frac{1}{10}\left[-\frac{0}{1}\log_2\frac{0}{1} - \frac{1}{1}\log_2\frac{1}{1}\right] + \frac{9}{10}\left[-\frac{7}{9}\log_2\frac{7}{9} - \frac{2}{9}\log_2\frac{2}{9}\right]$$
$$= 0 + \frac{9}{10}(0.7637)$$
$$= 0.6873$$

$$\text{Gain}(\text{CGPA} \in 6.8) = 0.8808 - 0.6873$$
$$= 0.1935$$

Similarly, the calculations are done for each of the distinct value for the attribute CGPA and a table is created. Now, the value of CGPA with maximum gain is chosen as the threshold value or the best split point. From Table 6.13, we can observe that CGPA with 7.9 has the maximum gain as 0.4462. Hence, CGPA ∈ 7.9 is chosen as the split point. Now, we can dicretize the continuous values of CGPA as two categories with CGPA ≤ 7.9 and CGPA > 7.9. The resulting discretized instances are shown in Table 6.14.

Table 6.14: Discretized Instances

S.No.	CGPA Continuous	CGPA Discretized	Job Offer
1.	9.5	>7.9	Yes
2.	8.2	>7.9	Yes
3.	9.1	>7.9	No
4.	6.8	≤7.9	No
5.	8.5	>7.9	Yes
6.	9.5	>7.9	Yes
7.	7.9	≤7.9	No
8.	9.1	>7.9	Yes
9.	8.8	>7.9	Yes
10.	8.8	>7.9	Yes

6.2.3 Classification and Regression Trees Construction

The Classification and Regression Trees (CART) algorithm is a multivariate decision tree learning used for classifying both categorical and continuous-valued target variables. CART algorithm is an example of multivariate decision trees that gives oblique splits. It solves both classification and regression problems. If the target feature is categorical, it constructs a classification tree and if the target feature is continuous, it constructs a regression tree. CART uses GINI Index to construct a decision tree. GINI Index is defined as the number of data instances for a class or it is the proportion of instances. It constructs the tree as a binary tree by recursively splitting a node into two nodes. Therefore, even if an attribute has more than two possible values, GINI Index is calculated for all subsets of the attributes and the subset which has maximum value is selected as the best split subset. For example, if an attribute A has three distinct values say $\{a_1, a_2, a_3\}$, the possible subsets are $\{\}$, $\{a_1\}$, $\{a_2\}$, $\{a_3\}$, $\{a_1, a_2\}$, $\{a_1, a_3\}$, $\{a_2, a_3\}$, and $\{a_1, a_2, a_3\}$. So, if an attribute has 3 distinct values, the number of possible subsets is 2^3, which means 8. Excluding the empty set $\{\}$ and the full set $\{a_1, a_2, a_3\}$, we have 6 subsets. With 6 subsets, we can form three possible combinations such as:

$\{a_1\}$ with $\{a_2, a_3\}$

$\{a_2\}$ with $\{a_1, a_3\}$

$\{a_3\}$ with $\{a_1, a_2\}$

Hence, in this CART algorithm, we need to compute the best splitting attribute and the best split subset i in the chosen attribute.

Higher the GINI value, higher is the homogeneity of the data instances.

Gini_Index(T) is computed as given in Eq. (6.13).

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2 \tag{6.13}$$

where,

P_i be the probability that a data instance or a tuple 'd' belongs to class C_i. It is computed as:

P_i = |No. of data instances belonging to class i|/|Total no of data instances in the training dataset T|

GINI Index assumes a binary split on each attribute, therefore, every attribute is considered as a binary attribute which splits the data instances into two subsets S_1 and S_2.

Gini_Index(T, A) is computed as given in Eq. (6.14).

$$\text{Gini\_Index}(T, A) = \frac{|S_1|}{|T|}\text{Gini}(S_1) + \frac{|S_2|}{|T|}\text{Gini}(S_2) \tag{6.14}$$

The splitting subset with minimum Gini_Index is chosen as the best splitting subset for an attribute. The best splitting attribute is chosen by the minimum Gini_Index which is otherwise maximum ΔGini because it reduces the impurity.

ΔGini is computed as given in Eq. (6.15):

$$\Delta\text{Gini}(A) = \text{Gini}(T) - \text{Gini}(T, A) \tag{6.15}$$

Algorithm 6.4: Procedure to Construct a Decision Tree using CART

1. Compute Gini_Index Eq. (6.13) for the whole training dataset based on the target attribute.
2. Compute Gini_Index for each of the attribute Eq. (6.14) and for the subsets of each attribute in the training dataset.
3. Choose the best splitting subset which has minimum Gini_Index for an attribute.
4. Compute ΔGini Eq. (6.15) for the best splitting subset of that attribute.
5. Choose the best splitting attribute that has maximum ΔGini.
6. The best split attribute with the best split subset is placed as the root node.
7. The root node is branched into two subtrees with each subtree an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into two subsets.
8. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.

Example 6.5: Choose the same training dataset shown in Table 6.3 and construct a decision tree using CART algorithm.

Solution:

Step 1: Calculate the Gini_Index for the dataset shown in Table 6.3, which consists of 10 data instances. The target attribute 'Job Offer' has 7 instances as Yes and 3 instances as No.

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2$$
$$= 1 - 0.49 - 0.09$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.42$$

Step 2: Compute Gini_Index for each of the attribute and each of the subset in the attribute. CGPA has 3 categories, so there are 6 subsets and hence 3 combinations of subsets (as shown in Table 6.15).

Table 6.15: Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9, \geq 8\}) = 1 - (7/8)^2 - (1/8)^2$$
$$= 1 - 0.7806$$
$$= 0.2194$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{<8\}) = 1 - (0/2)^2 - (2/2)^2$$
$$= 1 - 1$$
$$= 0$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 9, \geq 8), <8\} = (8/10) \times 0.2194 + (2/10) \times 0$$
$$= 0.17552$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9, <8\}) = 1 - (3/6)^2 - (3/6)^2$$
$$= 1 - 0.5 = 0.5$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 8\}) = 1 - (4/4)^2 - (0/4)^2$$
$$= 1 - 1 = 0$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 9, <8), \geq 8\}) = (6/10) \times 0.5 + (4/10) \times 0$$
$$= 0.3$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 8, <8\}) = 1 - (4/6)^2 - (2/6)^2$$
$$= 1 - 0.555$$
$$= 0.445$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9\}) = 1 - (3/4)^2 - (1/4)^2$$
$$= 1 - 0.625$$
$$= 0.375$$
$$\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 8, <8), \geq 9\}) = (6/10) \times 0.445 + (4/10) \times 0.375$$
$$= 0.417$$

Table 6.16 shows the Gini_Index for 3 subsets of CGPA.

Table 6.16: Gini_Index of CGPA

Subsets		Gini_Index
(≥9, ≥8)	<8	0.1755
(≥9, <8)	≥8	0.3
(≥8, <8)	≥9	0.417

Step 3: Choose the best splitting subset which has minimum Gini_Index for an attribute.

The subset CGPA ∈ {(≥9 ≥8), <8} has the lowest Gini_Index value as 0.1755 is chosen as the best splitting subset.

Step 4: Compute ΔGini or the best splitting subset of that attribute.

$$\Delta\text{Gini(CGPA)} = \text{Gini}(T) - \text{Gini}(T, \text{CGPA})$$
$$= 0.42 - 0.1755$$
$$= 0.2445$$

Repeat the same process for the remaining attributes in the dataset such as for Interactiveness shown in Table 6.17, Practical Knowledge in Table 6.18, and Communication Skills in Table 6.20.

Table 6.17: Categories for Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
Yes	5	1
No	2	2

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes}\}) = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2$$
$$= 1 - 0.72$$
$$= 0.28$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{No}\}) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2$$
$$= 1 - 0.5$$
$$= 0.5$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes, No}\}) = \frac{6}{10}(0.28) + \frac{4}{10}(0.5)$$
$$= 0.168 + 0.2$$
$$= 0.368$$

$$\Delta\text{Gini(Interactiveness)} = \text{Gini}(T) - \text{Gini}(T, \text{Interactiveness})$$
$$= 0.42 - 0.368$$
$$= 0.052$$

Table 6.18: Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\} = \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2$$
$$= 1 - 0.7544$$
$$= 0.2456$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Average}\}) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2$$
$$= 1 - 0.555 = 0.445$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\}, \text{Average}) = \left(\frac{7}{10}\right)^2 \times 0.2456 + \left(\frac{3}{10}\right) \times 0.445$$

$$= 0.3054$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2$$

$$= 1 - 0.52$$

$$= 0.48$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good}\}) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2$$

$$= 1 - 0.68$$

$$= 0.32$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}, \text{Good}) = \left(\frac{5}{10}\right) \times 0.48 + \left(\frac{5}{10}\right) \times 0.32$$

$$= 0.40$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}) = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2$$

$$= 1 - 0.5312 = 0.4688$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good}\}) = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2$$

$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good, Average}\}, \text{Very Good}) = \left(\frac{8}{10}\right) \times 0.4688 + \left(\frac{2}{10}\right) \times 0$$

$$= 0.3750$$

Table 6.19 shows the Gini_Index for various subsets of Practical Knowledge.

Table 6.19: Gini_Index for Practical Knowledge

Subsets		Gini_Index
(Very Good, Good)	Average	0.3054
(Very Good, Average)	Good	0.40
(Good, Average)	Very Good	0.3750

$$\Delta\text{Gini}(\text{Practical Knowledge}) = \text{Gini}(T) - \text{Gini}(T, \text{Practical Knowledge})$$

$$= 0.42 - 0.3054 = 0.1146$$

Table 6.20: Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Moderate}\}) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$
$$= 1 - 0.7806$$
$$= 0.2194$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Poor}\}) = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Moderate}\}, \text{Poor}) = \left(\frac{8}{10}\right) \times 0.2194 + \left(\frac{2}{10}\right) \times 0$$
$$= 0.1755$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Poor}\}) = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2$$
$$= 1 - 0.5101$$
$$= 0.4899$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate}\}) = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Poor}\}, \text{Moderate}) = \left(\frac{7}{10}\right) \times 0.4899 + \left(\frac{3}{10}\right) \times 0$$
$$= 0.3429$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate, Poor}\}) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2$$
$$= 1 - 0.52$$
$$= 0.48$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good}\}) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2$$
$$= 1 - 0.68$$
$$= 0.32$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate, Poor}\}, \text{Good}) = \left(\frac{5}{10}\right)^2 \times 048 + \left(\frac{5}{10}\right)^2 \times 0.32$$
$$= 0.40$$

Table 6.21 shows the Gini_Index for various subsets of Communication Skills.

Table 6.21: Gini-Index for Subsets of Communication Skills

Subsets		Gini_Index
(Good, Moderate)	Poor	0.1755
(Good, Poor)	Moderate	0.3429
(Moderate, Poor)	Good	0.40

$$\Delta\text{Gini(Communication Skills)} = \text{Gini}(T) - \text{Gini}(T, \text{Communication Skills})$$
$$= 0.42 - 0.1755$$
$$= 0.2445$$

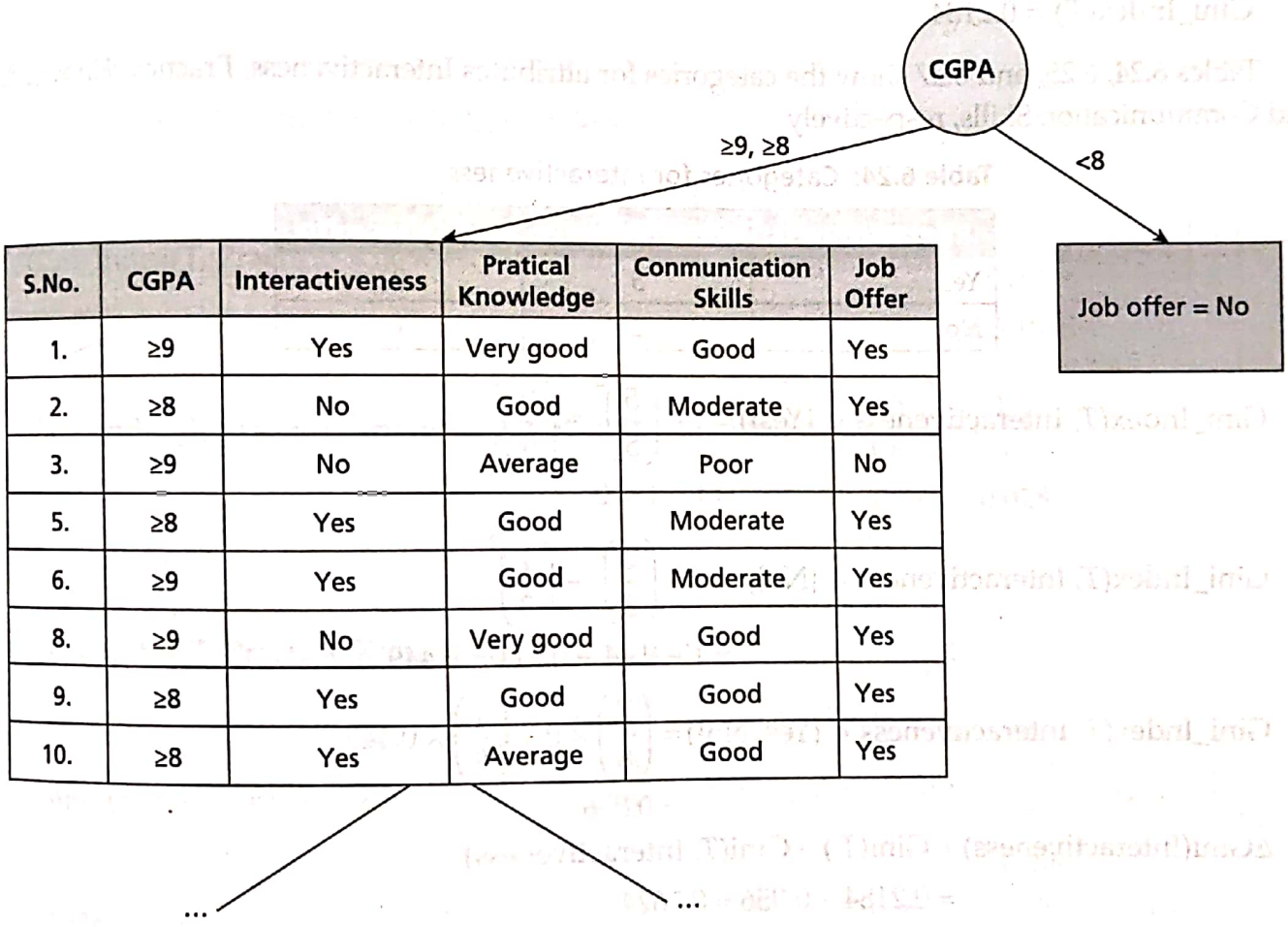
Table 6.22 shows the Gini_Index and ΔGini values calculated for all the attributes.

Table 6.22: Gini_Index and ΔGini for all Attributes

Attribute	Gini_Index	ΔGini
CGPA	0.1755	0.2445
Interactiveness	0.368	0.052
Practical knowledge	0.3054	0.1146
Communication Skills	0.1755	0.2445

Step 5: Choose the best splitting attribute that has maximum ΔGini.

CGPA and Communication Skills have the highest ΔGini value. We can choose CGPA as the root node and split the datasets into two subsets shown in Figure 6.7 since the tree constructed by CART is a binary tree.



S.No.	CGPA	Interactiveness	Pratical Knowledge	Conmunication Skills	Job Offer
1.	≥9	Yes	Very good	Good	Yes
2.	≥8	No	Good	Moderate	Yes
3.	≥9	No	Average	Poor	No
5.	≥8	Yes	Good	Moderate	Yes
6.	≥9	Yes	Good	Moderate	Yes
8.	≥9	No	Very good	Good	Yes
9.	≥8	Yes	Good	Good	Yes
10.	≥8	Yes	Average	Good	Yes

Figure 6.7: Decision Tree after Iteration 1

Iteration 2:

In the second iteration, the dataset has 8 data instances as shown in Table 6.23. Repeat the same process to find the best splitting attribute and the splitting subset for that attribute.

Table 6.23: Subset of the Training Dataset after Iteration 1

S.No.	CGPA	Interactiveness	Practical Knowledge	Communication Skills	Job Offer
1.	≥9	Yes	Very good	Good	Yes
2.	≥8	No	Good	Moderate	Yes
3.	≥9	No	Average	Poor	No
5.	≥8	Yes	Good	Moderate	Yes
6.	≥9	Yes	Good	Moderate	Yes
8.	≥9	No	Very good	Good	Yes
9.	≥8	Yes	Good	Good	Yes
10.	≥8	Yes	Average	Good	Yes

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$
$$= 1 - 0.766 - 0.0156$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.2184$$

Tables 6.24, 6.25, and 6.27 show the categories for attributes Interactiveness, Practical Knowledge, and Communication Skills, respectively.

Table 6.24: Categories for Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
Yes	5	0
No	2	1

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes}\}) = 1 - \left(\frac{5}{5}\right)^2 - \left(\frac{0}{5}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{No}\}) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2$$
$$= 1 - 0.44 - 0.111 = 0.449$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes, No}\}) = \left(\frac{7}{8}\right) \times 0 + \left(\frac{1}{8}\right) \times 0.449$$
$$= 0.056$$

$$\Delta\text{Gini(Interactiveness)} = \text{Gini}(T) - \text{Gini}(T, \text{Interactiveness})$$
$$= 0.2184 - 0.056 = 0.1624$$

Table 6.25: Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	0
Average	1	1

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\}) = 1 - \left(\frac{6}{6}\right)^2 - \left(\frac{0}{6}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Average}\}) = 1 - \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2$$
$$= 1 - 0.25 - 0.25$$
$$= 0.5$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\}, \text{Average}) = \left(\frac{6}{8}\right)^2 \times 0 + \left(\frac{2}{8}\right) \times 0.5$$
$$= 0.125$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2$$
$$= 1 - 0.5625 - 0.0625$$
$$= 0.375$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good}\}) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}, \text{Good}) = \left(\frac{4}{8}\right) \times 0.375 + \left(\frac{4}{8}\right) \times 0$$
$$= 0.1875$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good, Average}\}) = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2$$
$$= 1 - 0.694 - 0.028$$
$$= 0.278$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good}\}) = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good, Average}\}, \text{Very Good}) = \left(\frac{6}{8}\right)^2 \times 0.278 + \left(\frac{2}{8}\right)^2 \times 0$$
$$= 0.2085$$

Table 6.26 shows the Gini_Index values for various subsets of Practical Knowledge.

Table 6.26: Gini_Index for Subsets of Practical Knowledge

Subsets		Gini_Index
(Very Good, Good)	Average	0.125
(Very Good, Average)	Good	0.1875
(Good, Average)	Very Good	0.2085

$$\Delta\text{Gini(Practical Knowledge)} = \text{Gini}(T) - \text{Gini}(T, \text{Practical Knowledge})$$
$$= 0.2184 - 0.125$$
$$= 0.0934$$

Table 6.27: Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	0
Moderate	3	0
Poor	0	1

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Moderate}\}) = 1 - \left(\frac{7}{7}\right)^2 - \left(\frac{0}{7}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Poor}\}) = 1 - \left(\frac{0}{1}\right)^2 - \left(\frac{1}{1}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Moderate}\}, \text{Poor}) = \left(\frac{7}{8}\right)^2 \times 0 + \left(\frac{1}{8}\right) \times 0$$
$$= 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Poor}\}) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2$$
$$= 1 - 0.64 - 0.04$$
$$= 0.32$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate}\}) = 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good, Poor}\}, \text{Moderate}) = \left(\frac{5}{8}\right) \times 0.32 + \left(\frac{3}{8}\right) \times 0$$
$$= 0.2$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate, Poor}\}) = 1 - \left(\frac{3}{4}\right)^2 - \left(\frac{1}{4}\right)^2$$
$$= 1 - 0.5625 - 0.0625$$
$$= 0.375$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Good}\}) = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2$$
$$= 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{Communication Skills} \in \{\text{Moderate, Poor}\}, \text{Good}) = \left(\frac{4}{8}\right)^2 \times 0.375 + \left(\frac{4}{8}\right)^2 \times 0$$
$$= 0.1875$$

Table 6.28 shows the Gini_Index for subsets of Communication Skills.

Table 6.28: Gini_Index for Subsets of Communication Skills

Subsets		Gini_Index
(Good, Moderate)	Poor	0
(Good, Poor)	Moderate	0.2
(Moderate, Poor)	Good	0.1875

$$\Delta\text{Gini(Communication Skills)} = \text{Gini}(T) - \text{Gini}(T, \text{Communication Skills})$$
$$= 0.2184 - 0 = 0.2184$$

Table 6.29 shows the Gini_Index and ΔGini values for all attributes.

Table 6.29: Gini_Index and ΔGini Values for All Attributes

Attribute	Gini_Index	ΔGini
Interactiveness	0.056	0.1624
Practical knowledge	0.125	0.0934
Communication Skills	0	0.2184

Communication Skills has the highest ΔGini value. The tree is further branched based on the attribute 'Communication Skills'. Here, we see all branches end up in a leaf node and the process of construction is completed. The final tree is shown in Figure 6.8.

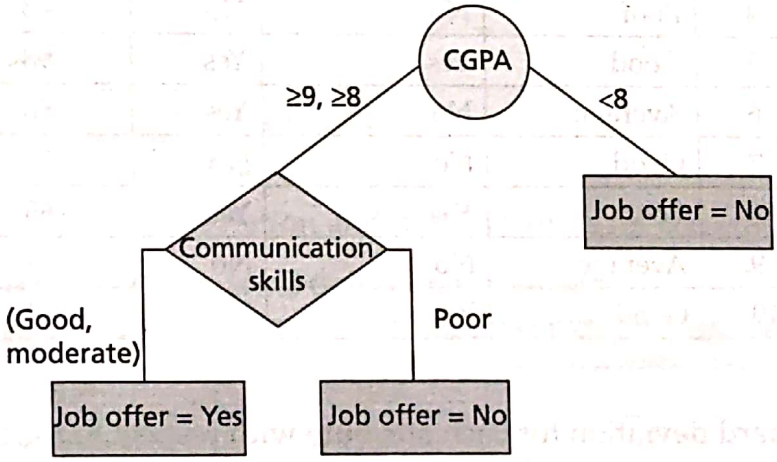


Figure 6.8: Final Tree

6.2.4 Regression Trees

Regression trees are a variant of decision trees where the target feature is a continuous valued variable. These trees can be constructed using an algorithm called reduction in variance which uses standard deviation to choose the best splitting attribute.

Algorithm 6.5: Procedure for Constructing Regression Trees

1. Compute standard deviation for each attribute with respect to target attribute.
2. Compute standard deviation for the number of data instances of each distinct value of an attribute.
3. Compute weighted standard deviation for each attribute.

(*Continued*)

4. Compute standard deviation reduction by subtracting weighted standard deviation for each attribute from standard deviation of each attribute.
5. Choose the attribute with a higher standard deviation reduction as the best split attribute.
6. The best split attribute is placed as the root node.
7. The root node is branched into subtrees with each subtree as an outcome of the test condition of the root node attribute. Accordingly, the training dataset is also split into different subsets.
8. Recursively apply the same operation for the subset of the training set with the remaining attributes until a leaf node is derived or no more training instances are available in the subset.

Example 6.6: Construct a regression tree using the following Table 6.30 which consists of 10 data instances and 3 attributes 'Assessment', 'Assignment' and 'Project'. The target attribute is the 'Result' which is a continuous attribute.

Table 6.30: Training Dataset

S.No.	Assessment	Assignment	Project	Result (%)
1.	Good	Yes	Yes	95
2.	Average	Yes	No	70
3.	Good	No	Yes	75
4.	Poor	No	No	45
5.	Good	Yes	Yes	98
6.	Average	No	Yes	80
7.	Good	No	No	75
8.	Poor	Yes	Yes	65
9.	Average	No	No	58
10.	Good	Yes	Yes	89

Solution:

Step 1: Compute standard deviation for each attribute with respect to the target attribute:

$$\text{Average} = (95 + 70 + 75 + 45 + 98 + 80 + 75 + 65 + 58 + 89) = 75$$

$$\text{Standard Deviation} = \sqrt{\frac{(95-75)^2 + (70-75)^2 + (75-75)^2 + (45-75)^2 + (98-75)^2 + (80-75)^2 + (75-75)^2 + (65-75)^2 + (58-75)^2 + (89-75)^2}{10}}$$

$$= 16.55$$

Assessment = Good (Table 6.31)

Table 6.31: Attribute Assessment = Good

S.No.	Assessment	Assignment	Project	Result (%)
1.	Good	Yes	Yes	95
3.	Good	No	Yes	75
5.	Good	Yes	Yes	98
7.	Good	No	No	75
10.	Good	Yes	Yes	89

Average = (95 + 75 + 98 + 75 + 89) = 86.4

$$\text{Standard Deviation} = \sqrt{\frac{(95-86.4)^2 + (75-86.4)^2 + (98-86.4)^2 + (75-86.4)^2 + (89-86.4)^2}{5}}$$

$$= 10.9$$

Assessment = Average (Table 6.32)

Table 6.32: Attribute Assessment = Average

S.No.	Assessment	Assignment	Project	Result (%)
2.	Average	Yes	No	70
6.	Average	No	Yes	80
9.	Average	No	No	58

Average = (70 + 80 + 58) = 69.3

$$\text{Standard Deviation} = \sqrt{\frac{(70-69.3)^2 + (80-69.3)^2 + (58-69.3)^2}{3}} = 11.01$$

Assessment = Poor (Table 6.33)

Table 6.33: Attribute Assessment = Poor

S.No.	Assessment	Assignment	Project	Result (%)
4.	Poor	No	No	45
8.	Poor	Yes	Yes	65

Average = (45 + 65) = 55

$$\text{Standard Deviation} = \sqrt{\frac{(45-55)^2 + (65-55)^2}{2}} = 14.14$$

Table 6.34 shows the standard deviation and data instances for the attribute-Assessment.

Table 6.34: Standard Deviation for Assessment

Assessment	Standard Deviation	Data Instances
Good	10.9	5
Average	11.01	3
Poor	14.14	2

$$\text{Weighted standard deviation for Assessment} = \left(\frac{5}{10}\right) \times 10.9 + \left(\frac{3}{10}\right) \times 11.01 + \left(\frac{2}{10}\right) \times 14.14$$

$$= 11.58$$

Standard deviation reduction for Assessment = 16.55 – 11.58 = 4.97

Assignment = Yes (Table 6.35)

Table 6.35: Assignment = Yes

S.No.	Assessment	Assignment	Project	Result (%)
1.	Good	Yes	Yes	95
2.	Average	Yes	No	70
5.	Good	Yes	Yes	98
8.	Poor	Yes	Yes	65
10.	Good	Yes	Yes	89

Average = $(95 + 70 + 98 + 65 + 89) = 83.4$

Standard Deviation = $\sqrt{\frac{(95 - 83.4)^2 + (70 - 83.4)^2 + (98 - 83.4)^2 + (65 - 83.4)^2 + (89 - 83.4)^2}{5}}$

$= 14.98$

Assignment = No (Table 6.36)

Table 6.36: Assignment = No

S.No.	Assessment	Assignment	Project	Result (%)
3.	Good	No	Yes	75
4.	Poor	No	No	45
6.	Average	No	Yes	80
7.	Good	No	No	75
9.	Average	No	No	58

Average = $(75 + 45 + 80 + 75 + 58) = 66.6$

Standard Deviation = $\sqrt{\frac{(75 - 66.6)^2 + (45 - 66.6)^2 + (80 - 66.6)^2 + (75 - 66.6)^2 + (58 - 66.6)^2}{5}}$

$= 14.7$

Table 6.37 shows the Standard Deviation and Data Instances for attribute, Assignment.

Table 6.37: Standard Deviation for Assignment

Assessment	Standard Deviation	Data Instances
Yes	14.98	5
No	14.7	5

Weighted standard deviation for Assignment = $\left(\frac{5}{10}\right) \times 14.98 + \left(\frac{5}{10}\right) \times 14.7 = 14.84$

Standard deviation reduction for Assignment = $16.55 - 14.84 = 1.71$

Project = Yes (Table 6.38)

Table 6.38: Project = Yes

S.No.	Assessment	Assignment	Project	Result (%)
1.	Good	Yes	Yes	95
3.	Good	No	Yes	75
5.	Good	Yes	Yes	98
6.	Average	No	Yes	80
8.	Poor	Yes	Yes	65
10.	Good	Yes	Yes	89

$$\text{Average} = (95 + 75 + 98 + 80 + 65 + 89) = 83.7$$

$$\text{Standard Deviation} = \sqrt{\frac{(95-83.7)^2 + (75-83.7)^2 + (98-83.7)^2 + (80-83.7)^2 + (65-83.7)^2 + (89-83.7)^2}{6}}$$

$$= 12.6$$

Project = No (Table 6.39)

Table 6.39: Project = No

S.No.	Assessment	Assignment	Project	Result (%)
2.	Average	Yes	No	70
4.	Poor	No	No	45
7.	Good	No	No	75
9.	Average	No	No	58

$$\text{Average} = (70 + 45 + 75 + 58) = 62$$

$$\text{Standard Deviation} = \sqrt{\frac{(70-75)^2 + (45-75)^2 + (75-75)^2 + (58-75)^2}{4}}$$

$$= 13.39$$

Table 6.40 shows the Standard Deviation and Data Instances for attribute, Project.

Table 6.40: Standard Deviation for Project

Project	Standard Deviation	Data Instances
Yes	12.6	6
No	13.39	4

$$\text{Weighted standard deviation for Assessment} = \left(\frac{6}{10}\right) \times 2.6 + \left(\frac{4}{10}\right) \times 13.39 = 12.92$$

Standard deviation reduction for Assessment = 16.55 – 12.92 = 3.63

Table 6.41 shows the standard deviation reduction for each attribute in the training dataset.

Table 6.41: Standard Deviation Reduction for Each Attribute

Attributes	Standard Deviation Reduction
Assessment	4.97
Assignment	1.71
Project	3.63

The attribute 'Assessment' has the maximum Standard Deviation Reduction and hence it is chosen as the best splitting attribute.

The training dataset is split into subsets based on the attribute 'Assessment' and this process is continued until the entire tree is constructed. Figure 6.9 shows the regression tree with 'Assessment' as the root node and the subsets in each branch.

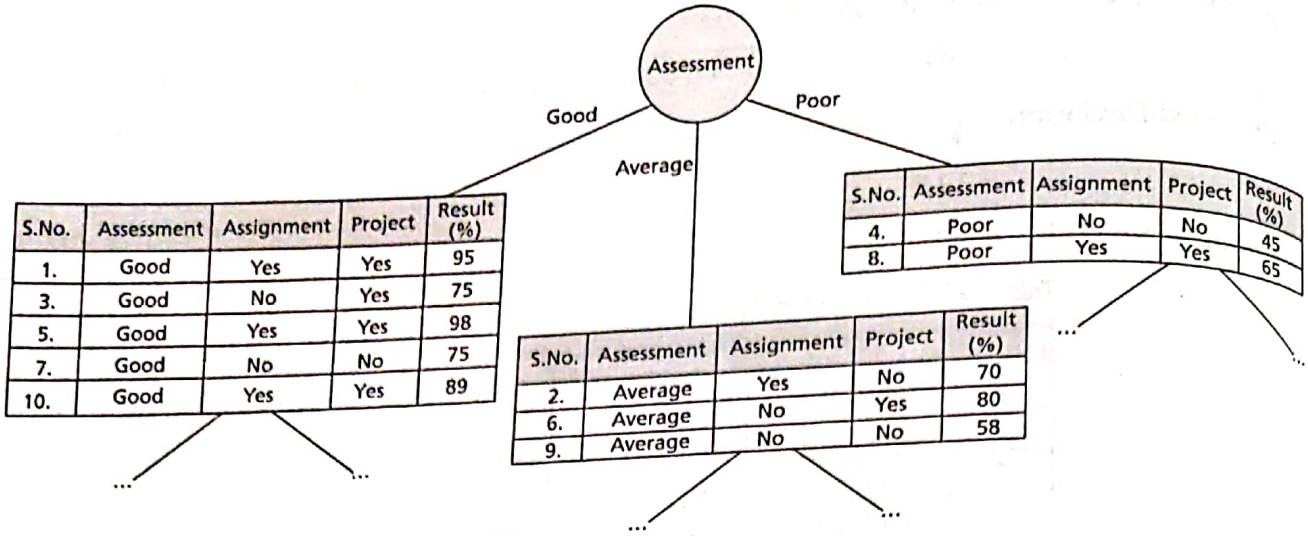


Figure 6.9: Regression Tree with Assessment as Root Node

The rest of regression tree construction can be done as an exercise.

6.3 VALIDATING AND PRUNING OF DECISION TREES

Inductive bias refers to a set of assumptions about the domain knowledge added to the training data to perform induction that is to construct a general model out of the training data. A bias is generally required as without it induction is not possible, since the training data can normally be generalized to a larger hypothesis space. Inductive bias in ID3 algorithm is the one that prefers the first acceptable shorter trees over larger trees, and when selecting the best split attribute during construction, attributes with high information gain are chosen. Thus, even though *ID3* searches a large space of decision trees, it constructs only a single decision tree when there may exist many alternate decision trees for the same training data. It applies a hill-climbing search that does not backtrack and may finally converge to a locally optimal solution that is not globally optimal. The shorter tree is preferred using Occam's razor principle which states that the simplest solution is the best solution.

Overfitting is also a general problem with decision trees. Once the decision tree is constructed, it must be validated for better accuracy and to avoid over-fitting and under-fitting. There is always a tradeoff between accuracy and complexity of the tree. The tree must be simple and accurate. If the tree is more complex, it can classify the data instances accurately for the training set but when test data is given, the tree constructed may perform poorly which means misclassifications are higher and accuracy is reduced. This problem is called as over-fitting.

To avoid overfitting of the tree, we need to prune the trees and construct an optimal decision tree. Trees can be pre-pruned or post-pruned. If tree nodes are pruned during construction or the construction is stopped earlier without exploring the nodes' branches, then it is called as pre-pruning whereas if tree nodes are pruned after the construction is over then it is called as post-pruning. Basically, the dataset is split into three sets called training dataset, validation dataset and test dataset. Generally, 40% of the dataset is used for training the decision tree and the remaining 60% is used for validation and testing. Once the decision tree is constructed, it is validated with the validation dataset and the misclassifications are identified. Using the number of

instances correctly classified and number of instances wrongly classified, Average Squared Error (ASE) is computed. The tree nodes are pruned based on these computations and the resulting tree is validated until we get a tree that performs better. Cross validation is another way to construct an optimal decision tree. Here, the dataset is split into k-folds, among which k–1 folds are used for training the decision tree and the k^{th} fold is used for validation and errors are computed. The process is repeated for randomly k–1 folds and the mean of the errors is computed for different trees. The tree with the lowest error is chosen with which the performance of the tree is improved. This tree can now be tested with the test dataset and predictions are made.

Another approach is that after the tree is constructed using the training set, statistical tests like error estimation and Chi-square test are used to estimate whether pruning or splitting is required for a particular node to find a better accurate tree.

The third approach is using a principle called Minimum Description Length which uses a complexity measure for encoding the training set and the growth of the decision tree is stopped when the encoding size (i.e., (size(tree)) + size(misclassifications(tree)) is minimized. CART and C4.5 perform post-pruning, that is, pruning the tree to a smaller size after construction in order to minimize the misclassification error. CART makes use of 10-fold cross validation method to validate and prune the trees, whereas C4.5 uses heuristic formula to estimate misclassification error rates.

Some of the tree pruning methods are listed below:

1. Reduced Error Pruning
2. Minimum Error Pruning (MEP)
3. Pessimistic Pruning
4. Error–based Pruning (EBP)
5. Optimal Pruning
6. Minimum Description Length (MDL) Pruning
7. Minimum Message Length Pruning
8. Critical Value Pruning

Summary

1. The decision tree learning model performs an *Inductive inference* that reaches a general conclusion from observed examples.
2. The decision tree learning model generates a complete hypothesis space in the form of a tree structure.
3. A decision tree has a structure that consists of a root node, internal nodes/decision nodes, branches, and terminal nodes/leaf nodes.
4. Every path from root to a leaf node represents a logical rule that corresponds to a conjunction of test attributes and the whole tree represents a disjunction of these conjunctions.
5. A decision tree consists of two major procedures, namely building the tree and knowledge inference or classification.
6. A decision tree is constructed by finding the attribute or feature that best describes the target class for the given test instances.