Chapter 6 Decision Tree Learning

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.

1.Model will classify both categorical and continuous valued target variables.
2.For a given training set X, hypothesis function f(X)as a decision tree.
3.The Decision tree generated a complete hypothesis space as a tree and allow us to search through the possible set of hypothesis.

Structure of a Decision Tree

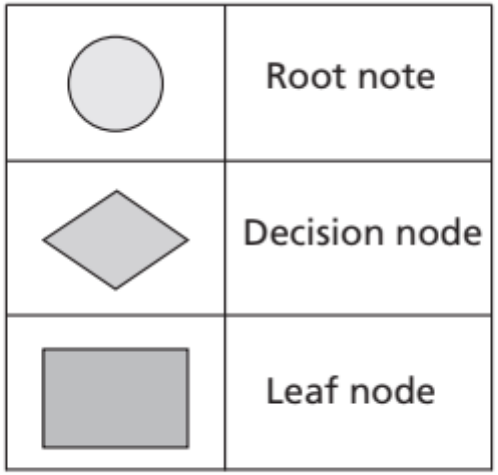


Figure 6.1: Nodes in a Decision Tree

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

1.It is a structure with root node ,internal node/decision nodes, branches and leaf nodes.
2.Topmost tree node is root node
3.Internal nodes are test nodes and also decision nodes
4.These nodes represent a choice or test an input attribute and the outputs for a test condition.
5.The branches are labelled as per the outcomes or output values of the test condition.
6.Each decision is a path to a leaf node .
7.The leaf nodes represent the label or outcomes of a decision path.
8.Every path from root to leaf node represents the logical rule corresponds to a conjunction of test attributes and the whole tree represents the disjunction of these conjunctions

Structure of a Decision Tree

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 6.2: 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.

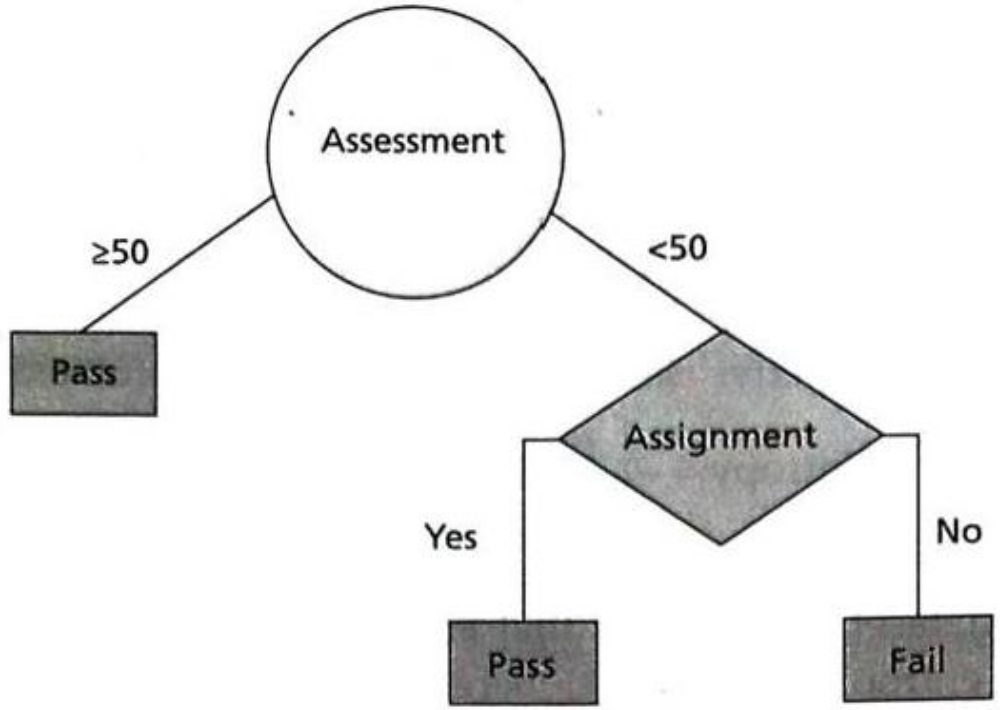


Figure 6.2: Illustration of a Decision Tree

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'.

Fundamentals of Entropy

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.

Higher the entropy → Higher the uncertainty

Lower the entropy → Lower the uncertainty

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

So, $P = P_1 \ldots\ldots P_n$ (6.2)

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

$$\text{Entropy\_Info}(P) = \text{Entropy\_Info}(P_1 \ldots P_n)$$
$$= -(P_1 \log_2(P_1) + P_2 \log_2(P_2) + \ldots\ldots + P_n \log_2(P_n)) \quad (6.3)$$

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|

General Algorithm for Decision Trees

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.
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.

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.

ID3 Tree Construction

1.ID3 is an example of univariate decision tree with only one feature at each decision node →axis aligned splits.
2.It constructs the tree using greedy approach in a top down fashion by identifying the best attribute of each level in the tree.
3.Works fine for discrete, otherwise it has to partitioned attributes to discrete features

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.

1. The algorithm builds a tree using pruning measure “information Gain” with given training dataset and constructed tree to classify test data.
2. Works fine for larger dataset.
3. Small dataset overfitting
4. Not accurate if missing values.
5. C4.5 and CART handle discrete and continuous values
6. . C4.5 and CART handle missing data as well.
7. C4.5 is prone to outliers and CART handle it well.

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.

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)

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

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

$$= 0.3609$$

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

$$= 0.5203$$

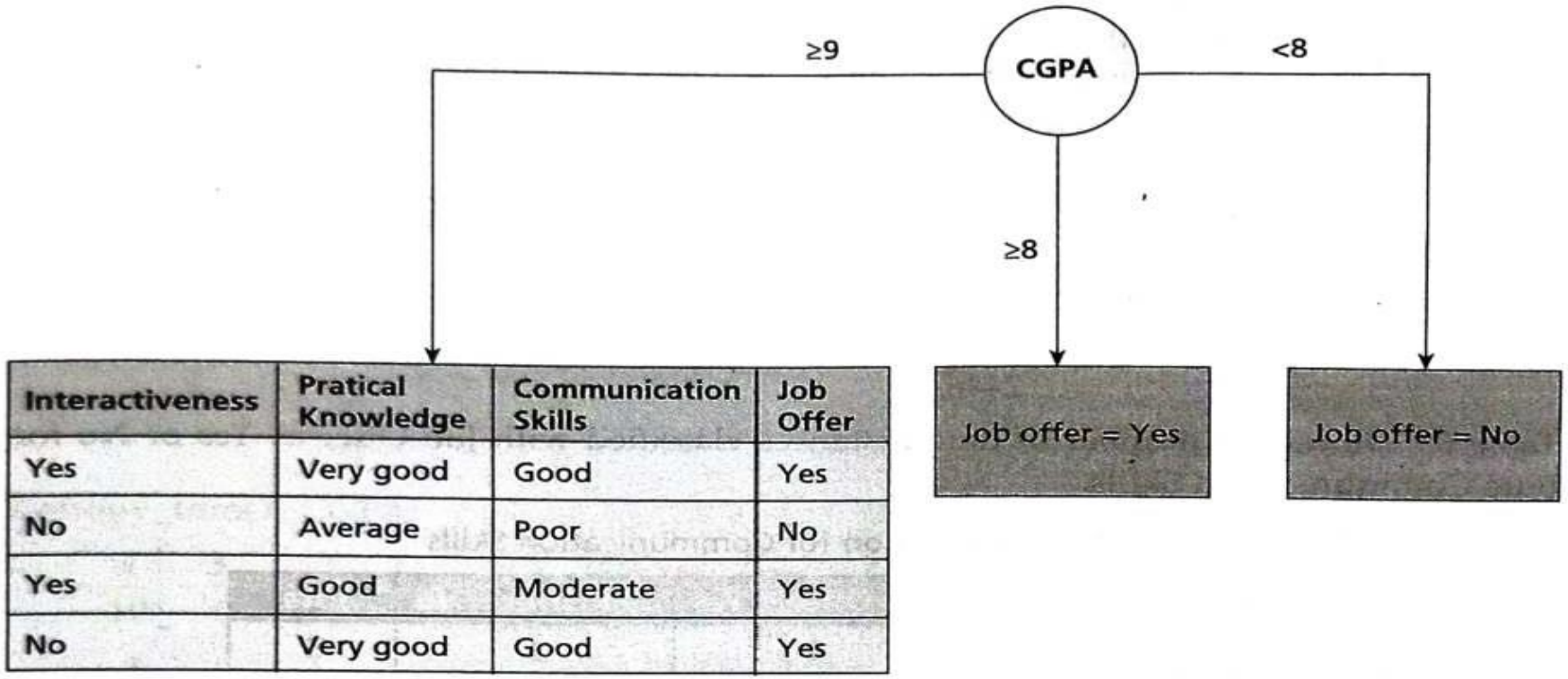
The Gain calculated for all the attributes is shown in Table 6.8:

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.

Entropy_Info(T) = 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.

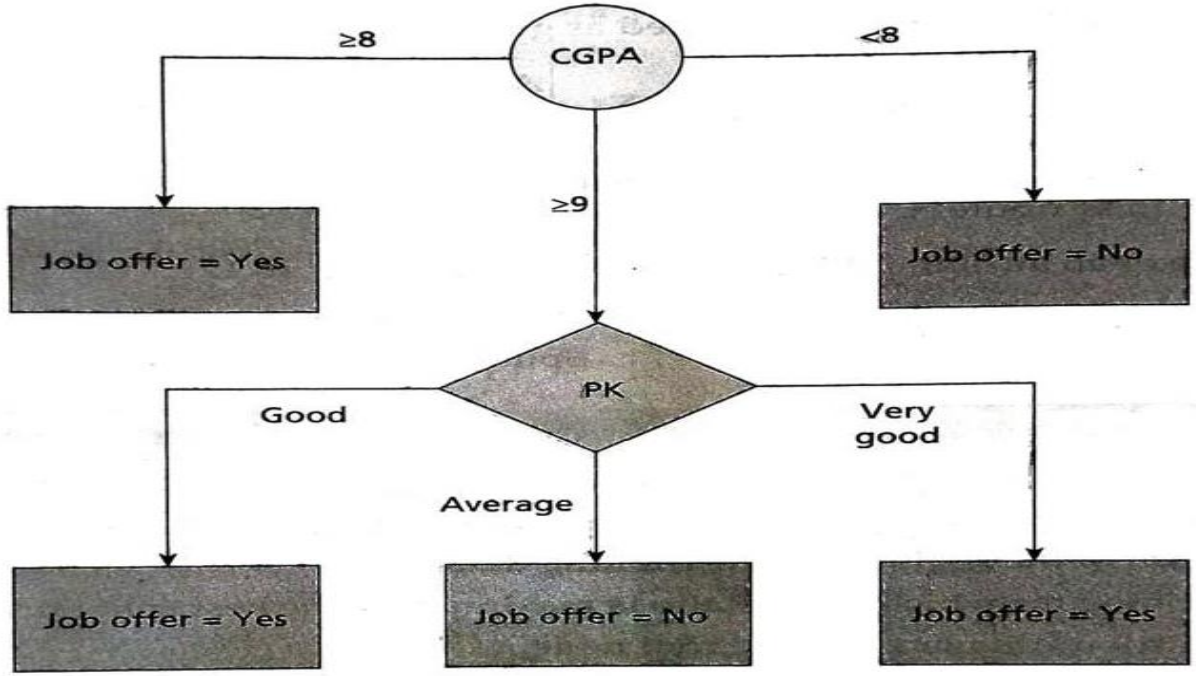


Figure 6.4: Final Decision Tree

C4.5 Construction

1. Works with missing values by marking ‘?’
2. Buids C5.0
3. Occam’s Razor problem ,two correct solutions, the simple one is chosen.
4. Need larger dataset for accuracy .
5. Uses gain ration as a measure during the construction of decision tree.

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.

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.

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

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 instanc for distinct value 'i' in attribute A.

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

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

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'.

$$\text{Entropy\_Info(Target Attribute = Job Offer)} = \text{Entropy\_Info}(7, 3) = -\left[\frac{7}{10}\log_2\frac{7}{10} + \frac{3}{10}\log_2\frac{3}{10}\right]$$
$$= (-0.3599 + -0.5208)$$
$$= 0.8807$$

Step 2: Calculate the Entropy_Info, Gain(Info_Gain), Split_Info, Gain_Ratio for each of the attril 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:

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

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

$$= 0.3898 + 0.3998 = 0.7896$$

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

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

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

$$= \frac{0.0911}{0.9704}$$

$$= 0.0939$$

$= 0.0939$

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:

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

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

$$= 0.3609$$

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

$$= 0.5202$$

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

$$= 1.4853$$

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

$$= \frac{0.5202}{1.4853} = 0.3502$$

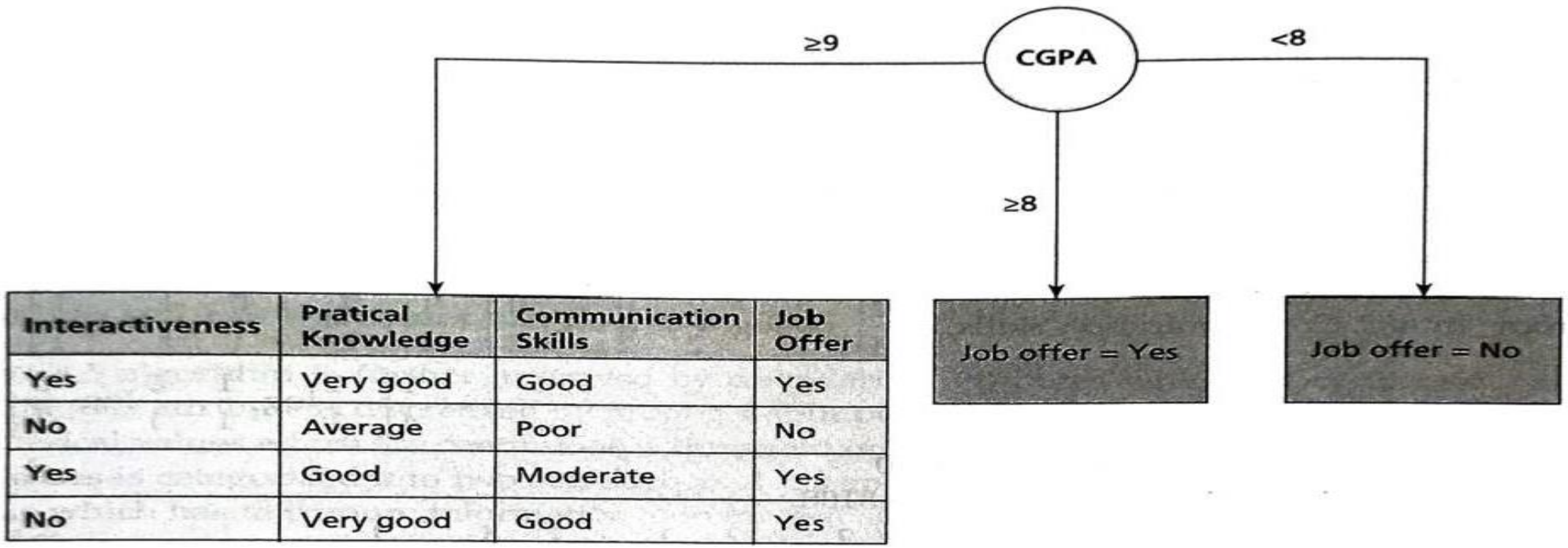
Table 6.10 shows the Gain_Ratio computed for all the attributes.

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:

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

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

$$\text{Gain\_Ratio(Interactiveness)} = \frac{\text{Gain(Interactiveness)}}{\text{Split\_Info}(T, \text{Interactiveness})}$$
$$= \frac{0.3112}{1} = 0.3112$$

Practical Knowledge:

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

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

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

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

Communication Skills:

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

$$= 0$$

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

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

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

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.

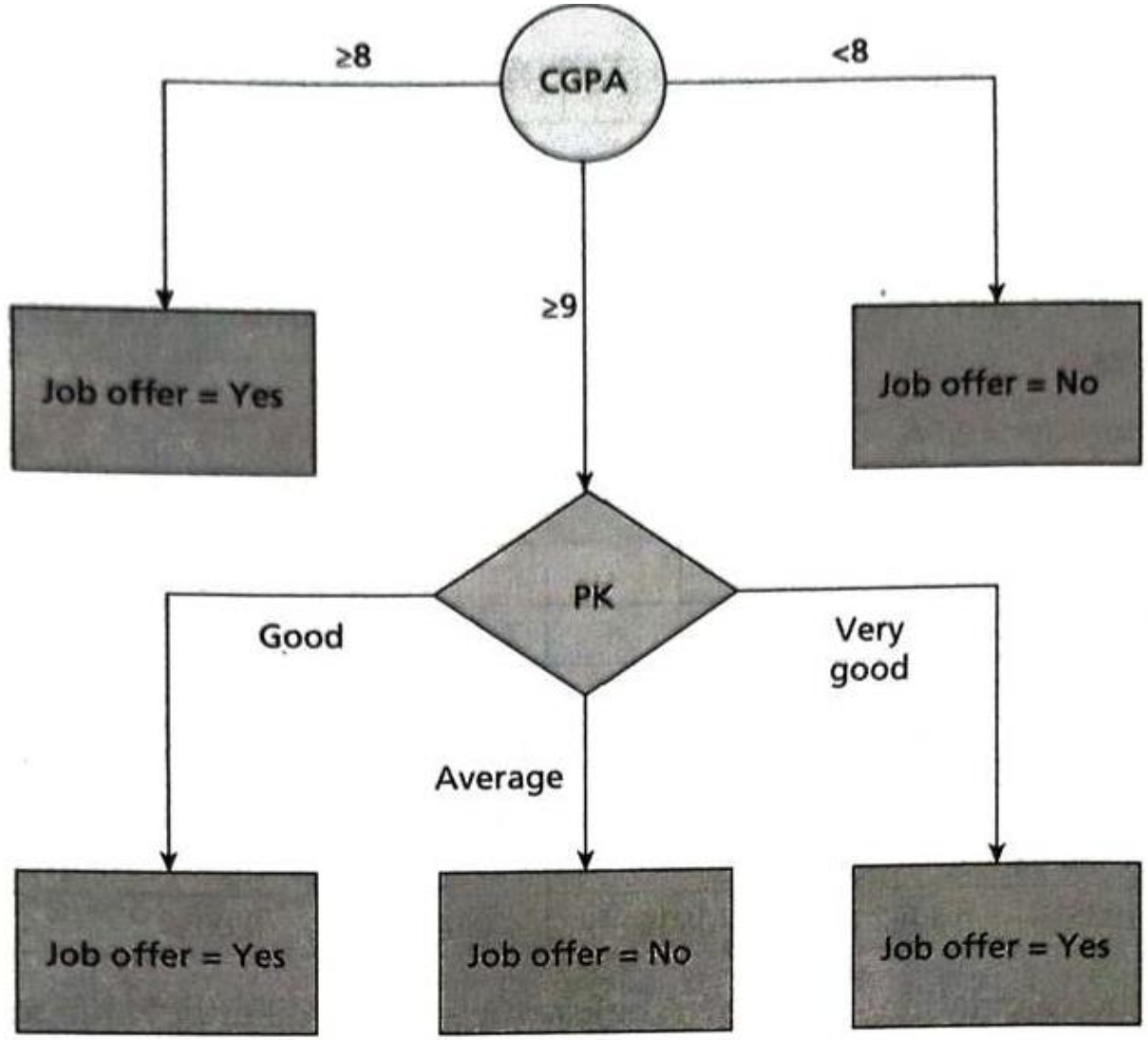


Figure 6.6: Final Decision Tree

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.

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

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

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

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

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

Classification and Regression Trees Construction

1. Multivariate decision tree algorithm
2. Oblique splits
3. Categorical ->classification tree otherwise ->regression tree.
4. Uses GINI index to construct decision tree
5. GINI ->number of data instances or its proportion of instances
6. Tree construction by splitting the nodes into two nodes
7. If an attribute has more than two possible values, GINI index is calculated for all the subsets of the attributes and the subset which has a maximum value is selected as the best split subset.

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 $\in$ {(≥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.

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

Repeat the same process for the remaining attributes in the dataset such as for Interactivenes 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

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

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

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

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

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

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

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

Gini_Index(T, Practical Knowledge $\in$ {Very Good, Good}, 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

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

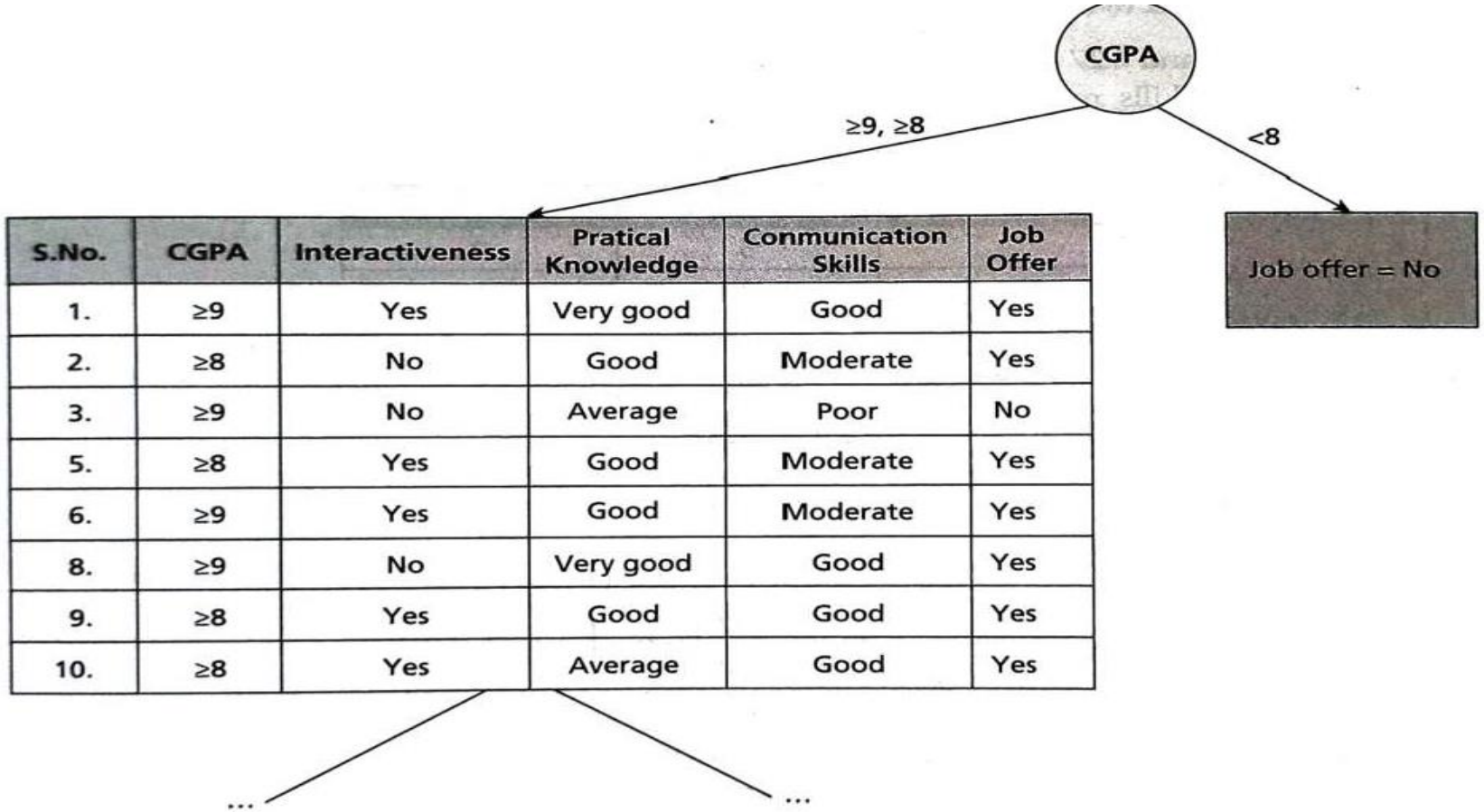
Table 6.22 shows the Gini_Index and ΔGini values calculated for all the attributes.

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

Regression Trees

1.Variant of decision tree with continuous .

2.Uses reduction in variance which uses standard deviation to choose the best splitting attribute.

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 6.6: 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

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

$$= 11.58$$

$$\text{Standard deviation reduction for Assessment} = 16.55 - 11.58 = 4.97$$

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

$$\text{Average} = (95 + 70 + 98 + 65 + 89) = 83.4$$

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

$$= 14.98$$

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

$$\text{Average} = (75 + 45 + 80 + 75 + 58) = 66.6$$

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

$$= 14.7$$

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

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

$$\text{Standard deviation reduction for Assessment} = 16.55 - 12.92 = 3.63$$

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

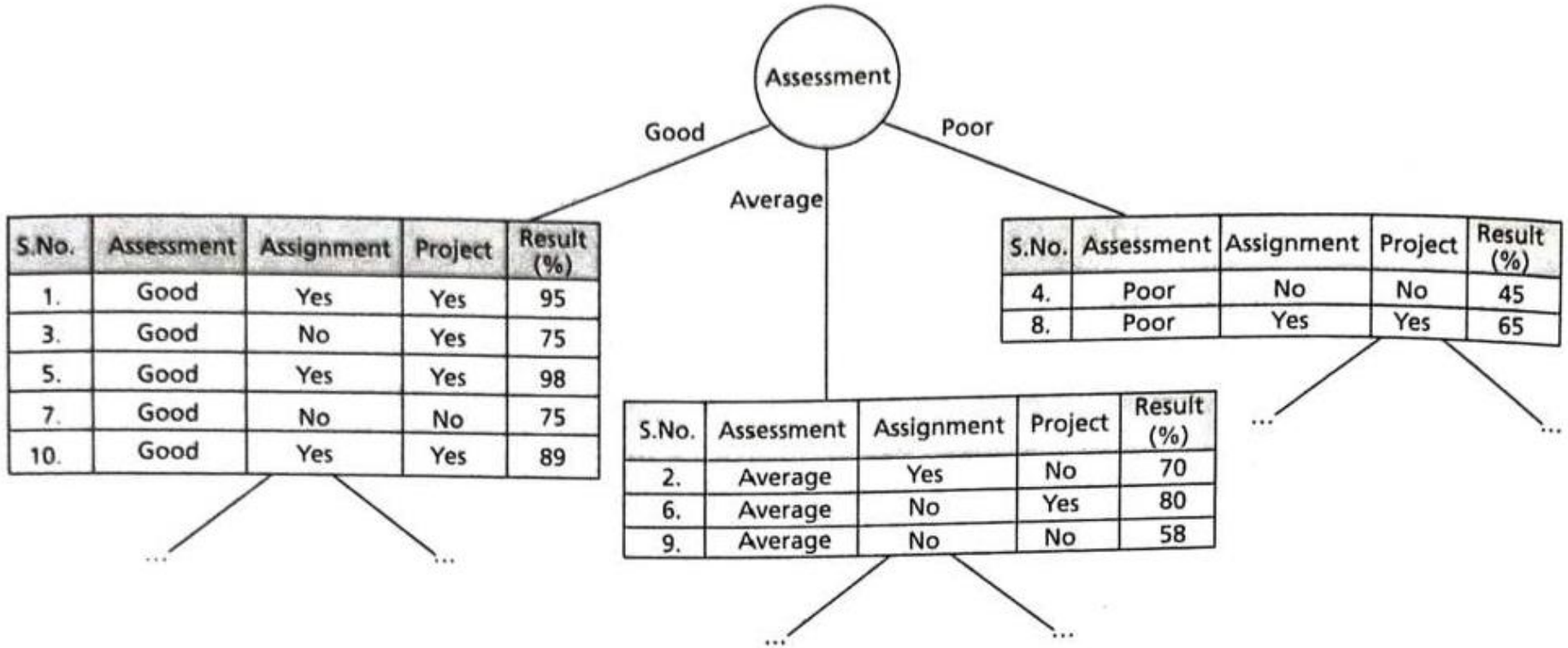


Figure 6.9: Regression Tree with Assessment as Root Node

Validating and Pruning of Decision Trees

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

Chapter 8 -Bayesian Learning

Bayesian Learning is a learning method that describes and represents knowledge in an uncertain domain and provides a way to reason about this knowledge using probability measure. It uses Bayes theorem to infer the unknown parameters of a model. Bayesian inference is useful in many applications which involve reasoning and diagnosis such as game theory, medicine, etc. Bayesian inference is much more powerful in handling missing data and for estimating any uncertainty in predictions.

Probability based learning

1. Prior knowledge or prior probabilities with observed data.
2. How to model the randomness ,uncertainty and noise to predict future events.
3. Probabilistic model –randomness deterministic –no randomness
4. Naïve bayes and Bayesian Belief Network (BBN)

Fundamentals of Bayes Theorem

Naïve Bayes Model relies on Bayes theorem that works on the principle of three kinds of probabilities called prior probability, likelihood probability, and posterior probability.

Prior Probability

It is the general probability of an uncertain event before an observation is seen or some evidence is collected. It is the initial probability that is believed before any new information is collected.

Likelihood Probability

Likelihood probability is the relative probability of the observation occurring for each class or the sampling density for the evidence given the hypothesis. It is stated as P (Evidence | Hypothesis), which denotes the likeliness of the occurrence of the evidence given the parameters.

Posterior Probability

It is the updated or revised probability of an event taking into account the observations from the training data. P (Hypothesis | Evidence) is the posterior distribution representing the belief about the hypothesis, given the evidence from the training data. Therefore,

Posterior probability = prior probability + new evidence

Classification Using Bayes Model

1. **Based on Bayes principle**
2. **Based on mathematical formula used to determine posterior probability ,given prior probabilities of events .**
3. **Based on prior knowledge and posterior distributions**

Hypothesis h is the target class to be classified and evidence E is the given test instance

$$P(\text{Hypothesis } h \mid \text{Evidence } E) = \frac{P(\text{Evidence } E \mid \text{Hypothesis } h)\, P(\text{Hypothesis } h)}{P(\text{Evidence } E)} \qquad (8.1)$$

$$\text{Posterior Probability } \alpha \text{ Prior Probability} \times \text{Likelihood Probability}$$

Maximum A Posteriori (MAP) Hypothesis, h_{MAP}

Given a set of candidate hypotheses, the hypothesis which has the maximum value is considered as the *maximum probable hypothesis* or *most probable hypothesis*. This most probable hypothesis is called the Maximum A Posteriori Hypothesis h_{MAP}. Bayes theorem Eq. (8.1) can be used to find the h_{MAP}.

$$h_{MAP} = \max_{h \in H} P(Hypothesish | Evidence\ E)$$

$$= \max_{h \in H} \frac{P(Evidence\ E \mid Hypothesis\ h) P(Hypothesis\ h)}{P(Evidence\ E)}$$

$$= \max_{h \in H} P(Evidence\ E \mid Hypothesis\ h) P(Hypothesis\ h) \qquad (8.2)$$

Maximum Likelihood (ML) Hypothesis, h_{ML}

Given a set of candidate hypotheses, if every hypothesis is equally probable, only $P(E \mid h)$ is used to find the *most probable hypothesis*. The hypothesis that gives the maximum likelihood for $P(E \mid h)$ is called the Maximum Likelihood (ML) Hypothesis, h_{ML}.

$$h_{ML} = \max_{h \in H} P(Evidence\ E \mid Hypothesis\ h) \qquad (8.3)$$

Correctness of Bayes Theorem

Consider two events A and B in a sample space S.

A T F T T F T T F

B F T T F T F T F

$P(A) = 5/8$

$P(B) = 4/8$

$P(A \mid B) = 2/4$

$P(B \mid A) = 2/5$

$P(A \mid B) = P(B \mid A)\ P(A) / P(B) = = 2/4$

$P(B \mid A) = P(A \mid B)\ P(B) / P(A) = = 2/5$

Let us consider a numerical example to illustrate the use of Bayes theorem now:

Example 8.1: Consider a boy who has a volleyball tournament on the next day, but today he feels sick. It is unusual that there is only a 40% chance he would fall sick since he is a healthy boy. Now, Find the probability of the boy participating in the tournament. The boy is very much interested in volley ball, so there is a 90% probability that he would participate in tournaments and 20% that he will fall sick given that he participates in the tournament.

Solution: P (Boy participating in the tournament) = 90%

P (He is sick | Boy participating in the tournament) = 20%

P (He is Sick) = 40%

The probability of the boy participating in the tournament given that he is sick is:

P (Boy participating in the tournament | He is sick) = P (Boy participating in the tournament) $\times$ P (He is sick | Boy participating in the tournament)/P (He is Sick)

P (Boy participating in the tournament | He is sick) = $(0.9 \times 0.2)/0.4$

$= 0.45$

Hence, 45% is the probability that the boy will participate in the tournament given that he is sick.

Naïve Bayes Algorithm

Algorithm 8.1: Naïve Bayes

1. Compute the prior probability for the target class.
2. Compute Frequency matrix and likelihood Probability for each of the feature.
3. Use Bayes theorem Eq. (8.1) to calculate the probability of all hypotheses.
4. Use Maximum A Posteriori (MAP) Hypothesis, h_{MAP} Eq. (8.2) to classify the test object to the hypothesis with the highest probability.

Example 8.2: Assess a student's performance using Naïve Bayes algorithm with the dataset provided in Table 8.1. Predict whether a student gets a job offer or not in his final year of the course.

Table 8.1: Training Dataset

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: The training dataset T consists of 10 data instances with attributes such as 'CGPA', 'Interactiveness', 'Practical Knowledge' and 'Communication Skills' as shown in Table 8.1. The target variable is Job Offer which is classified as Yes or No for a candidate student.

Step 1: Compute the prior probability for the target feature 'Job Offer'. The target feature 'Job Offer' has two classes, 'Yes' and 'No'. It is a binary classification problem. Given a student instance, we need to classify whether 'Job Offer = Yes' or 'Job Offer = No'.

From the training dataset, we observe that the frequency or the number of instances with 'Job Offer = Yes' is 7 and 'Job Offer = No' is 3.

The prior probability for the target feature is calculated by dividing the number of instances belonging to a particular target class by the total number of instances.

Hence, the prior probability for 'Job Offer = Yes' is 7/10 and 'Job Offer = No' is 3/10 as shown in Table 8.2.

Table 8.2: Frequency Matrix and Prior Probability of Job Offer

Job Offer Classes	No. of Instances	Probability Value
Yes	7	P (Job Offer = Yes) = 7/10
No	3	P (Job Offer = No) = 3/10

Step 2: Compute Frequency matrix and Likelihood Probability for each of the feature.

Step 2(a): Feature – CGPA

Table 8.3 shows the frequency matrix for the feature CGPA.

Table 8.3: Frequency Matrix of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2
Total	7	3

Table 8.4 shows how the likelihood probability is calculated for CGPA using conditional probability.

Table 8.4: Likelihood Probability of CGPA

CGPA	*P* (Job Offer = Yes)	*P* (Job Offer = No)
≥9	*P* (CGPA ≥9 | Job Offer = Yes) = 3/7	*P* (CGPA ≥9 | Job Offer = No) = 1/3
≥8	*P* (CGPA ≥8 | Job Offer = Yes) = 4/7	*P* (CGPA ≥8 | Job Offer = No) = 0/3
<8	*P* (CGPA <8 | Job Offer = Yes) = 0/7	*P* (CGPA <8 | Job Offer = No) = 2/3

As explained earlier the Likelihood probability is stated as the sampling density for the evidence given the hypothesis. It is denoted as *P* (Evidence | Hypothesis), which says how likely is the occurrence of the evidence given the parameters.

It is calculated as the number of instances of each attribute value and for a given class value divided by the number of instances with that class value.

For example *P* (CGPA ≥9 | Job Offer = Yes) denotes the number of instances with 'CGPA ≥9' and 'Job Offer = Yes' divided by the total number of instances with 'Job Offer = Yes'.

From the Table 8.3 Frequency Matrix of CGPA, number of instances with 'CGPA ≥9' and 'Job Offer = Yes' is 3. The total number of instances with 'Job Offer = Yes' is 7. Hence, *P* (CGPA ≥9 | Job Offer = Yes) = 3/7.

Similarly, the Likelihood probability is calculated for all attribute values of feature CGPA.

Step 2(b): Feature – Interactiveness

Table 8.5 shows the frequency matrix for the feature Interactiveness.

Table 8.5: Frequency Matrix of Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
YES	5	1
NO	2	2
Total	7	3

Table 8.6 shows how the likelihood probability is calculated for Interactiveness using conditional probability.

Table 8.6: Likelihood Probability of Interactiveness

Interactiveness	P (Job Offer = Yes)	P (Job Offer = No)
YES	P (Interactiveness = Yes | Job Offer = Yes) = 5/7	P (Interactiveness = Yes | Job Offer = No) = 1/3
NO	P (Interactiveness = No | Job Offer = Yes) = 2/7	P (Interactiveness = No | Job Offer = No) = 2/3

Step 2(c): Feature – Practical Knowledge

Table 8.7 shows the frequency matrix for the feature Practical Knowledge.

Table 8.7: Frequency Matrix of Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Average	1	2
Good	4	1
Total	7	3

Table 8.8 shows how the likelihood probability is calculated for Practical Knowledge using conditional probability.

Table 8.8: Likelihood Probability of Practical Knowledge

Practical Knowledge	*P* (Job Offer = Yes)	*P* (Job Offer = No)
Very Good	*P* (Practical Knowledge = Very Good | Job Offer = Yes) = 2/7	*P* (Practical Knowledge = Very Good | Job Offer = No) = 0/3
Average	*P* (Practical Knowledge = Average | Job Offer = Yes) = 1/7	*P* (Practical Knowledge = Average | Job Offer = No) = 2/3
Good	*P* (Practical Knowledge = Good | Job Offer = Yes) = 4/7	*P* (Practical Knowledge = Good | Job Offer = No) = 1/3

Step 2(d): Feature – Communication Skills

Table 8.9 shows the frequency matrix for the feature Communication Skills.

Table 8.9: Frequency Matrix of Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2
Total	7	3

Table 8.10 shows how the likelihood probability is calculated for Communication Skills using conditional probability.

Table 8.10: Likelihood Probability of Communication Skills

Communication Skills	*P* (Job Offer = Yes)	*P* (Job Offer = No)
Good	*P* (Communication Skills = Good | Job Offer = Yes) = 4/7	*P* (Communication Skills = Good | Job Offer = No) = 1/3
Moderate	*P* (Communication Skills = Moderate | Job Offer = Yes) = 3/7	*P* (Communication Skills = Moderate | Job Offer = No) = 0/3
Poor	*P* (Communication Skills = Poor | Job Offer = Yes) = 0/7	*P* (Communication Skills = Poor | Job Offer = No) = 2/3

Step 3: Use Bayes theorem Eq. (8.1) to calculate the probability of all hypotheses.

Given the test data = (CGPA ≥9, Interactiveness = Yes, Practical knowledge = Average, Communication Skills = Good), apply the Bayes theorem to classify whether the given student gets a Job offer or not.

P (Job Offer = Yes | Test data) = (*P*(CGPA ≥9 | Job Offer = Yes) *P* (Interactiveness = Yes | Job Offer = Yes) *P* (Practical knowledge = Average | Job Offer = Yes) *P* (Communication Skills = Good | Job Offer = Yes) *P* (Job Offer = Yes)))/(*P* (Test Data))

We can ignore *P* (Test Data) in the denominator since it is common for all cases to be considered.

Hence, *P* (Job Offer = Yes | Test data) = (*P*(CGPA ≥9 |Job Offer = Yes) *P* (Interactiveness = Yes | Job Offer = Yes) *P* (Practical knowledge = Average | Job Offer = Yes) *P* (Communication Skills = Good | Job Offer = Yes) *P* (Job Offer = Yes)

= 3/7 × 5/7 × 1/7 × 4/7 × 7/10

= 0.0175

Similarly, for the other case 'Job Offer = No',

We compute the probability,

P (Job Offer = No| Test data) = (*P*(CGPA ≥9 |Job Offer =No) *P* (Interactiveness = Yes | Job Offer = No) *P* (Practical knowledge = Average | Job Offer = No) *P* (Communication Skills = Good | Job Offer =No) *P* (Job Offer = No))/(*P*(Test Data)).

P (CGPA ≥9 |Job Offer = No) *P* (Interactiveness = Yes | Job Offer = No) *P* (Practical knowledge = Average | Job Offer = No) *P* (Communication Skills = Good | Job Offer = No) *P* (Job Offer = No)

= 1/3 × 1/3 × 2/3 × 1/3 × 3/10

= 0.0074

Step 4: Use Maximum A Posteriori (MAP) Hypothesis, h_{MAP} Eq. (8.2) to classify the test object to the hypothesis with the highest probability.

Since P (Job Offer = Yes | Test data) has the highest probability value, the test data is classified as **'Job Offer = Yes'**.

ZERO PROBABILITY ERROR

Zero-probability error can be solved by applying a smoothing technique called Laplace correction which means given 1000 data instances in the training dataset, if there are zero instances for a particular value of a feature we can add 1 instance for each attribute value pair of that feature which will not make much difference for 1000 data instances and the overall probability does not become zero.

Zero Probability Error

In Example 8.1, consider the test data to be (CGPA ≥8, Interactiveness = Yes, Practical knowledge = Average, Communication Skills = Good)

When computing the posterior probability,

P (Job Offer = Yes | Test data) = (P(CGPA ≥8 | Job Offer = Yes) P (Interactiveness = Yes | Job Offer = Yes) P (Practical knowledge = Average | Job Offer = Yes) P (Communication Skills = Good | Job Offer = Yes) P (Job Offer = Yes)))/((P(Test Data))

P (Job Offer = Yes | Test data) = (P(CGPA ≥8 | Job Offer = Yes) P(Interactiveness = Yes | Job Offer = Yes) P (Practical knowledge = Average | Job Offer = Yes) P (Communication Skills = Good | Job Offer = Yes) P (Job Offer = Yes)

= 4/7 × 5/7 × 1/7 × 4/7 × 7/10

= 0.0233

Similarly, for the other case 'Job Offer = No',

When we compute the probability:

P (Job Offer = No | Test data) = (P(CGPA ≥8 | Job Offer = No) P (Interactiveness = Yes | Job Offer = No) P (Practical knowledge = Average | Job Offer = No) P (Communication Skills = Good | Job Offer = No) P (Job Offer = No))/(P(Test Data))

= P (CGPA ≥8 | Job Offer =No) P (Interactiveness = Yes | Job Offer = No) P (Practical knowledge = Average | Job Offer = No) P (Communication Skills = Good | Job Offer =No) P (Job Offer = No)

= 0/3 × 1/3 × 2/3 × 1/3 × 3/10

= 0

Now, let us scale the values given in Table 8.1 for 1000 data instances. The scaled values without Laplace correction are shown in Table 8.11.

Table 8.11: Scaled Values to 1000 without Laplace Correction

CGPA	*P* (Job Offer = Yes)	*P* (Job Offer = No)
≥9	*P* (CGPA ≥9 | Job Offer = Yes) = 300/700	*P* (CGPA ≥9 | Job Offer = No) = 100/300
≥8	*P* (CGPA ≥8 | Job Offer = Yes) = 400/700	*P* (CGPA ≥8 | Job Offer = No) = 0/300
<8	*P* (CGPA <8 | Job Offer = Yes) = 0/700	*P* (CGPA <8 | Job Offer = No) = 200/300

Now, add 1 instance for each CGPA-value pair for 'Job Offer = No'. Then,

P (CGPA ≥9 | Job Offer = No) = 101/303 = 0.333

P (CGPA ≥8 | Job Offer = No) = 1/303 = 0.0033

P (CGPA <8 | Job Offer = No) = 201/303 = 0.6634

With scaled values to 1003 data instances, we get

P (Job Offer = Yes | Test data) = (*P*(CGPA ≥8 |Job Offer = Yes) *P* (Interactiveness = Yes | Job Offer = Yes) *P* (Practical knowledge = Average | Job Offer = Yes) *P* (Communication Skills = Good | Job Offer = Yes) *P* (Job Offer = Yes)

= 400/700 × 500/700 × 100/700 × 400/700 × 700/1003

= 0.02325

P (Job Offer = No | Test data) = P(CGPA ≥8 |Job Offer = No) P (Interactiveness = Yes | Job Offer = No) P (Practical knowledge = Average | Job Offer = No) P (Communication Skills = Good | Job Offer = No) P (Job Offer = No)

= 1/303 × 100/300 × 200/300 × 100/300 × 303/1003

= 0.00007385

Thus, using Laplace Correction, Zero Probability error can be solved with Naïve Bayes classifier.

Brute Force Bayes Algorithm

Applying Bayes theorem, Brute Force Bayes algorithm relies on the idea of concept learning wherein given a hypothesis space H for the training dataset T, the algorithm computes the posterior probabilities for all the hypothesis $h_i \in H$. Then, Maximum A Posteriori (MAP) Hypothesis, h_{MAP}, is used to output the hypothesis with maximum posterior probability. The algorithm is quite expensive since it requires computations for all the hypotheses. Although computing posterior probabilities is inefficient, this idea is applied in various other algorithms which is also quite interesting.

Bayes Optimal Classifier

Bayes optimal classifier is a probabilistic model, which in fact, uses the Bayes theorem to find the most probable classification for a new instance given the training data by combining the predictions of all posterior hypotheses, This is different from Maximum A Posteriori (MAP) Hypothesis, h_{MAP} which chooses the maximum probable hypothesis or the most probable hypothesis.

Here, a new instance can be classified to a possible classification value C_i by the following Eq. (8.4).

$$= \max_{C_i} \sum_{h_i \in H} P(C_i \mid h_i)\, P(h_i \mid T) \qquad (8.4)$$

Example 8.3: Given the hypothesis space with 4 hypothesis h_1, h_2, h_3 and h_4. Determine if the patient is diagnosed as COVID positive or COVID negative using Bayes Optimal classifier.

Solution: From the training dataset T, the posterior probabilities of the four different hypotheses for a new instance are given in Table 8.12.

Table 8.12: Posterior Probability Values

$P(h_i \mid T)$	P (COVID Positive $\mid h_i$)	P (COVID Negative $\mid h_i$)
0.3	0	1
0.1	1	0
0.2	1	0
0.1	1	0

h_{MAP} chooses h_1 which has the maximum probability value 0.3 as the solution and gives the result that the patient is COVID negative. But Bayes Optimal classifier combines the predictions of h_2, h_3 and h_4 which is 0.4 and gives the result that the patient is COVID positive.

$$\sum_{h_i \in H} P(\text{COVID Negative} \mid h_i) P(h_i \mid T) = 0.3 \times 1 = 0.3$$

$$\sum_{h_i \in H} P(\text{COVID Positive} \mid h_i) P(h_i \mid T) = 0.1 \times 1 + 0.2 \times 1 + 0.1 \times 1 = 0.4$$

Therefore, $\max_{C_i \in \{\text{COVID Positive, COVID Negative}\}} \sum_{h_i \in H} P(C_i \mid h_i) P(h_i \mid T) = \text{COVID Positive.}$

Thus, this algorithm, diagnoses the new instance to be COVID positive.

Gibbs Algorithm

The main drawback of Bayes optimal classifier is that it computes the posterior probability for all hypotheses in the hypothesis space and then combines the predictions to classify a new instance.

Gibbs algorithm is a sampling technique which randomly selects a hypothesis from the hypothesis space according to the posterior probability distribution and classifies a new instance. It is found that the prediction error occurs twice with the Gibbs algorithm when compared to Bayes Optimal classifier.

Naïve Bayes Algorithm For Continuous Attributes

There are two ways to predict with Naive Bayes algorithm for continuous attributes:

1. Discretize continuous feature to discrete feature.
2. Apply Normal or Gaussian distribution for continuous feature.

GAUSSIAN NAIVE BAYES ALGORITHM

IN GAUSSIAN NAIVE BAYES, THE VALUES OF CONTINUOUS FEATURES ARE ASSUMED TO BE SAMPLED FROM A GAUSSIAN DISTRIBUTION.

Example 8.4: Assess a student's performance using Naïve Bayes algorithm for the continuous attribute. Predict whether a student gets 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' and 'Interactiveness' as shown in Table 8.13. The target variable is Job Offer which is classified as Yes or No for a candidate student.

Table 8.13: Training Dataset with Continuous Attribute

S.No.	CGPA	Interactiveness	Job Offer
1.	9.5	Yes	Yes
2.	8.2	No	Yes
3.	9.3	No	No
4.	7.6	No	No
5.	8.4	Yes	Yes
6.	9.1	Yes	Yes
7.	7.5	Yes	No
8.	9.6	No	Yes
9.	8.6	Yes	Yes
10.	8.3	Yes	Yes

Solution:

Step 1: Compute the prior probability for the target feature 'Job Offer'.

Prior probabilities of both the classes are calculated using the same formula (refer to Table 8.14).

Table 8.14: Prior Probability of Target Class

Job Offer Classes	No. of Instances	Probability Value
Yes	7	P (Job Offer = Yes) = 7/10
No	3	P (Job Offer = No) = 3/10

Step 2: Compute Frequency matrix and Likelihood Probability for each of the feature.

Likelihood probabilities for a continuous attribute is obtained from Gaussian (Normal) Distribution. In the above data set, CGPA is a continuous attribute for which we need to apply Gaussian distribution to calculate the likelihood probability.

Gaussian distribution for continuous feature is calculated using the given formula,

$$P(X_i = x_k \mid C_j) = g(x_k, \mu_{ij}\sigma_{ij}) \tag{8.5}$$

where,

X_i is the i^{th} continuous attribute in the given dataset and x_k is a value of the attribute.

C_j denotes the j^{th} class of the target feature.

μ_{ij} denotes the mean of the values of that continuous attribute X_i with respect to the class j of the target feature.

σ_{ij} denotes the standard deviation of the values of that continuous attribute X_i with respect to the class j of the target feature.

Hence, the normal distribution formula is given as:

$$P(X_i = x_k \mid C_j) = \frac{1}{\sigma_{ij}\sqrt{2\pi}} e^{\frac{(x_k - \mu_{ij})^2}{2\sigma_{ij}^2}} \tag{8.6}$$

Step 2(a): Consider the feature CGPA

In this example CGPA is a continuous attribute,

To calculate the likelihood probability for this continuous attribute, first compute the mean and standard deviation for CGPA with respect to the target class 'Job Offer'.

Here, X_i = CGPA

C_j = 'Job Offer = Yes'

Mean and Standard Deviation for class 'Job Offer = Yes' are given as:

$\mu_{ij} = \mu_{CGPA-YES} = 8.814286$

$\sigma_{ij} = \sigma_{CGPA-YES} = 0.58146$

Mean and Standard Deviation for class 'Job Offer = No' are given as:

C_j = 'Job Offer = No'

$\mu_{ij} = \mu_{CGPA-NO} = 8.133333$

$\sigma_{ij} = \sigma_{CGPA-NO} = 1.011599$

Once Mean and Standard Deviation are computed, the likelihood probability for any test value using Gaussian distribution formula can be calculated.

Step 2(b): Consider the feature Interactiveness

Interactiveness is a discrete feature whose probability is calculated as earlier.

Table 8.15 shows the frequency matrix for the feature Interactiveness.

Table 8.15: Frequency Matrix of Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
YES	5	1
NO	2	2
Total	7	3

Table 8.16 shows how the likelihood probability is calculated for Interactiveness using conditional probability.

Table 8.16: Likelihood Probability of Interactiveness

Interactiveness	*P* (Job Offer = Yes)	*P* (Job Offer = No)
YES	*P* (Interactiveness = Yes | Job Offer = Yes) = 5/7	*P* (Interactiveness = Yes | Job Offer = No) = 1/3
NO	*P* (Interactiveness = No | Job Offer = Yes) = 2/7	*P* (Interactiveness = No | Job Offer = No) = 2/3

Step 3: Use Bayes theorem to calculate the probability of all hypotheses.

Consider the test data to be (CGPA = 8.5, Interactiveness = Yes).

For the hypothesis 'Job Offer = Yes':

P (Job Offer = Yes | Test data) = (*P*(CGPA = 8.5 | Job Offer = Yes) × *P* (Interactiveness = Yes | Job Offer = Yes) × *P* (Job Offer = Yes)

To compute *P* (CGPA = 8.5 | Job Offer = Yes) use Gaussian distribution formula:

$P(X_i = x_k \mid C_j) = g(x_k, \mu_{ij}\sigma_{ij})$

$$P(X_{CGPA} = 8.5 \mid C_{\text{Job Offer = Yes}}) = \frac{1}{\sigma_{ij}\sqrt{2\pi}} e^{\frac{(x_k-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

$$P(X_{CGPA} = 8.5 \mid C_{\text{Job Offer = Yes}}) = \frac{1}{\sigma_{CGPA-YES}\sqrt{2\pi}} e^{\frac{(8.5-\mu_{CGPA-YES})^2}{2\sigma_{CGPA-YES}^2}}$$

$$P(\text{CGPA} = 8.5 \mid \text{Job Offer} = \text{Yes}) = g(x_k = 8.5, \mu_{ij} = 8.814\, \sigma_{ij} = 0.581)$$

$$= \frac{1}{0.581\sqrt{2\pi}} e^{\frac{(8.5-8.814)^2}{2\times 0.581^2}} = 0.594$$

P (Interactiveness = Yes | Job Offer = Yes) = 5/7

P (Job Offer = Yes) = 7/10

Hence:

P (Job Offer = Yes | Test data) = (*P*(CGPA = 8.5 | Job Offer = Yes) × *P* (Interactiveness = Yes | Job Offer = Yes) × *P* (Job Offer = Yes)

$= 0.594 \times 5/7 \times 7/10$

$= 0.297$

Similarly, for the hypothesis 'Job Offer = No':

P (Job Offer = No | Test data) = P (CGPA = 8.5 | Job Offer = No) × P (Interactiveness = Yes | Job Offer = No) × P (Job Offer = No)

P (CGPA = 8.5 | Job Offer = No) = g ($x_k = 8.5$, $\mu_{ij} = 8.133$ $\sigma_{ij} = 1.0116$)

$$P(X_{CGPA} = 8.5 \mid C_{\text{Job Offer = Yes}}) = \frac{1}{\sigma_{CGPA-NO}\sqrt{2\pi}} e^{\frac{-(8.5-\mu_{CGPA-NO})^2}{2\sigma_{CGPA-NO}^2}}$$

$$= \frac{1}{1.0116\sqrt{2\pi}} e^{\frac{(8.5-8.133)^2}{2\times 1.0116^2}} = 0.369$$

P (Interactiveness = Yes | Job Offer = No) = 1/3

= P (Job Offer = No) = 0.369

Hence,

P (Job Offer = No | Test data) = P (CGPA = 8.5 | Job Offer = No) P (Interactiveness = Yes | Job Offer = No) × P (Job Offer = No)

$= 0.369 \times 1/3 \times 3/10$

$= 0.0369$

Step 4: Use Maximum A Posteriori (MAP) Hypothesis, h_{MAP} to classify the test object to the hypothesis with the highest probability.

Since P (Job Offer = Yes | Test data) has the highest probability value of 0.297,the test data is classified as '**Job Offer = Yes**'.

```
# import necessary libraries
import pandas as pd
from sklearn import tree
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
# Load Data from CSV
data = pd.read_csv('tennisdata.csv')
print("The first 5 Values of data is :\n", data.head())
```

```
The first 5 Values of data is :
     Outlook Temperature Humidity   Windy PlayTennis
0     Sunny         Hot     High    Weak         No
1     Sunny         Hot     High  Strong         No
2  Overcast         Hot     High    Weak        Yes
3      Rain        Mild     High    Weak        Yes
4      Rain        Cool   Normal    Weak        Yes
```

	Outlook	Temperature	Humidity	Windy	PlayTennis
1	Outlook	Temperature	Humidity	Windy	PlayTennis
2	Sunny	Hot	High	Weak	No
3	Sunny	Hot	High	Strong	No
4	Overcast	Hot	High	Weak	Yes
5	Rain	Mild	High	Weak	Yes
6	Rain	Cool	Normal	Weak	Yes
7	Rain	Cool	Normal	Strong	No
8	Overcast	Cool	Normal	Strong	Yes
9	Sunny	Mild	High	Weak	No
10	Sunny	Cool	Normal	Weak	Yes
11	Rain	Mild	Normal	Weak	Yes
12	Sunny	Mild	Normal	Strong	Yes
13	Overcast	Mild	High	Strong	Yes
14	Overcast	Hot	Normal	Weak	Yes
15	Rain	Mild	High	Strong	No

```
# obtain train data and train output
X = data.iloc[:, :-1]
print("\nThe First 5 values of the train data is\n", X.head())
```

```
The First 5 values of the train data is
     Outlook Temperature Humidity   Windy
0     Sunny         Hot     High    Weak
1     Sunny         Hot     High  Strong
2  Overcast         Hot     High    Weak
3      Rain        Mild     High    Weak
4      Rain        Cool   Normal    Weak
```

```
y = data.iloc[:, -1]
print("\nThe First 5 values of train output is\n", y.head())
```

```
The First 5 values of train output is
 0     No
1     No
2    Yes
3    Yes
4    Yes
Name: PlayTennis, dtype: object
```

```
# convert them in numbers
le_outlook = LabelEncoder()
X.Outlook = le_outlook.fit_transform(X.Outlook)

le_Temperature = LabelEncoder()
X.Temperature = le_Temperature.fit_transform(X.Temperature)

le_Humidity = LabelEncoder()
X.Humidity = le_Humidity.fit_transform(X.Humidity)

le_Windy = LabelEncoder()
X.Windy = le_Windy.fit_transform(X.Windy)
print("\nNow the Train output is\n", X.head())
```

```
Now the Train output is
    Outlook  Temperature  Humidity  Windy
0        2            1         0      1
1        2            1         0      0
2        0            1         0      1
3        1            2         0      1
4        1            0         1      1
```

```
le_PlayTennis = LabelEncoder()
y = le_PlayTennis.fit_transform(y)
print("\n Now the Train output is\n",y)
```

```
Now the Train output is
 [0 0 1 1 1 0 1 0 1 1 1 1 1 0]
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.20)

classifier = GaussianNB()
classifier.fit(X_train, y_train)

from sklearn.metrics import accuracy_score
print("Accuracy is:", accuracy_score(classifier.predict(X_test), y_test))
```

```
Accuracy is: 0.3333333333333333
```