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.

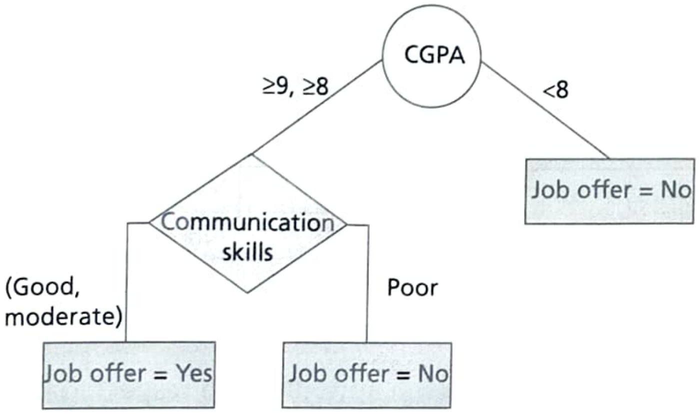


Figure 6.8: Final Tree

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:

Average = (95 + 70 + 75 + 45 + 98 + 80 + 75 + 65 + 58 + 89) = 75

$$\text{Standard Deviation} = \sqrt{\frac{\begin{array}{r}(95-75)^2+(70-75)^2+(75-75)^2+(45-75)^2+(98-75)^2+(80-7\\+(75-75)^2+(65-75)^2+(58-75)^2+(89-7\end{array}}{10}}$$

$$= 16.55$$

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

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

= 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

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

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

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

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

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

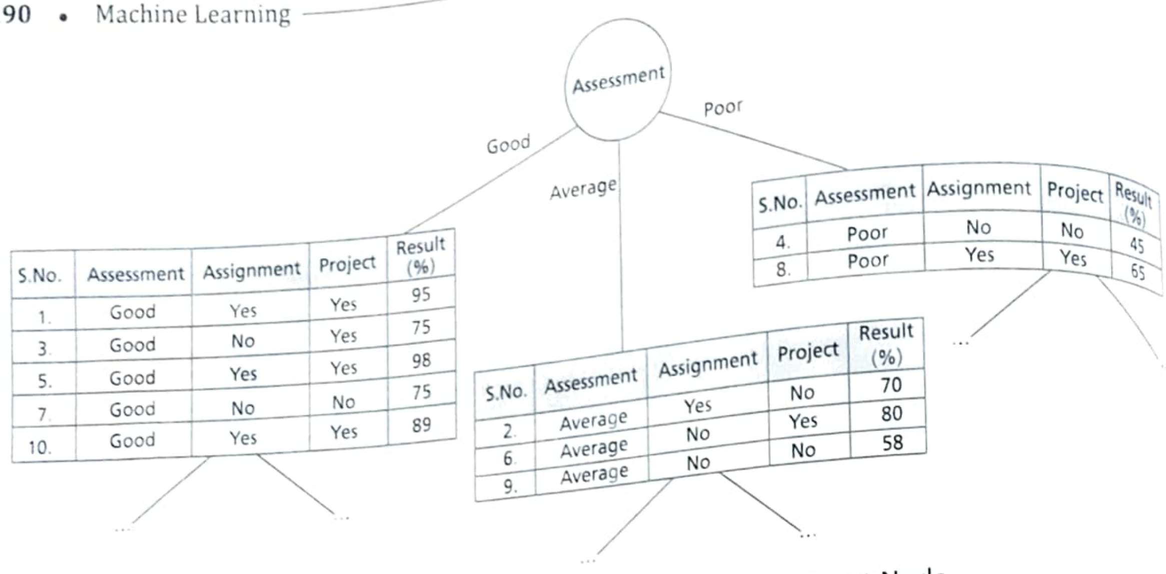
Table 6.41 shows the standard deviation reduction for each attribute in the training dataset.

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.



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

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

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

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 tances correctly classified and number of instances wrongly classified, Average Squared Error SE) is computed. The tree nodes are pruned based on these computations and the resulting tree validated until we get a tree that performs better. Cross validation is another way to construct optimal decision tree. Here, the dataset is split into k-folds, among which k–1 folds are used training the decision tree and the k^{th} fold is used for validation and errors are computed. The ocess is repeated for randomly k–1 folds and the mean of the errors is computed for different es. The tree with the lowest error is chosen with which the performance of the tree is improved. is 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 ror estimation and Chi-square test are used to estimate whether pruning or splitting is required r a particular node to find a better accurate tree.

The third approach is using a principle called Minimum Description Length which uses complexity measure for encoding the training set and the growth of the decision tree is stopped hen the encoding size (i.e., (size(tree)) + size(misclassifications(tree)) is minimized. CART and 4.5 perform post-pruning, that is, pruning the tree to a smaller size after construction in order minimize the misclassification error. CART makes use of 10-fold cross validation method to alidate and prune the trees, whereas C4.5 uses heuristic formula to estimate misclassification rror 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.

finding the attribute or feature that best describes the target class