Classification-

- ➔ It is a task of assigning objects to one particular group based upon test results.
- ➔ It is a task of learning a target function f that maps each attribute set x to one of the predefined class labels y.

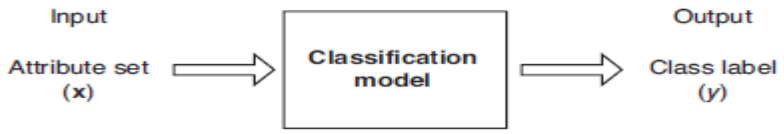


Figure 4.2. Classification as the task of mapping an input attribute set $\mathbf{x}$ into its class label y.

- ➔ The target function is also known as classification model.

Consider the below data set,

Table 4.1. The vertebrate data set.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

From the above data set we can identify or classify the class labels for an object based up on the attribute set.

A classification models are used to distinguish the objects in the following ways-

1. Descriptive modeling
2. Predictive modeling

Descriptive Modeling-

It is a classification model that acts as an explanatory tool to distinguish between objects of different classes.

It can summarize the data & explains what features define a mammal, reptile, bird, fish or amphibian.

Predictive Modeling-

It is a classification model that can be used to predict the class label of unknown records.

Suppose we are given the following characteristics of a creature called CAT, and to determine the class label to which the creature belongs. For that we can use a classification model that can built from the above data set.

Name	Body--Temperature	Skin Cover	Gives -Birth	Aquatic -Creature	Aerial -Creature	Has Legs	Hibernates	ClassLabel
CAT	WARM BLOOD	FUR	YES	NO	NO	YES	NO	?

Note - Classification techniques are most suited for binary or nominal attributes and less effective for ordinal attributes.

General approach to solve a classification problem

A Classification technique is a systematic approach to build a classification model.

Below is the list of classification techniques-

Decision tree induction

Rule Based classifiers

Support vector machines

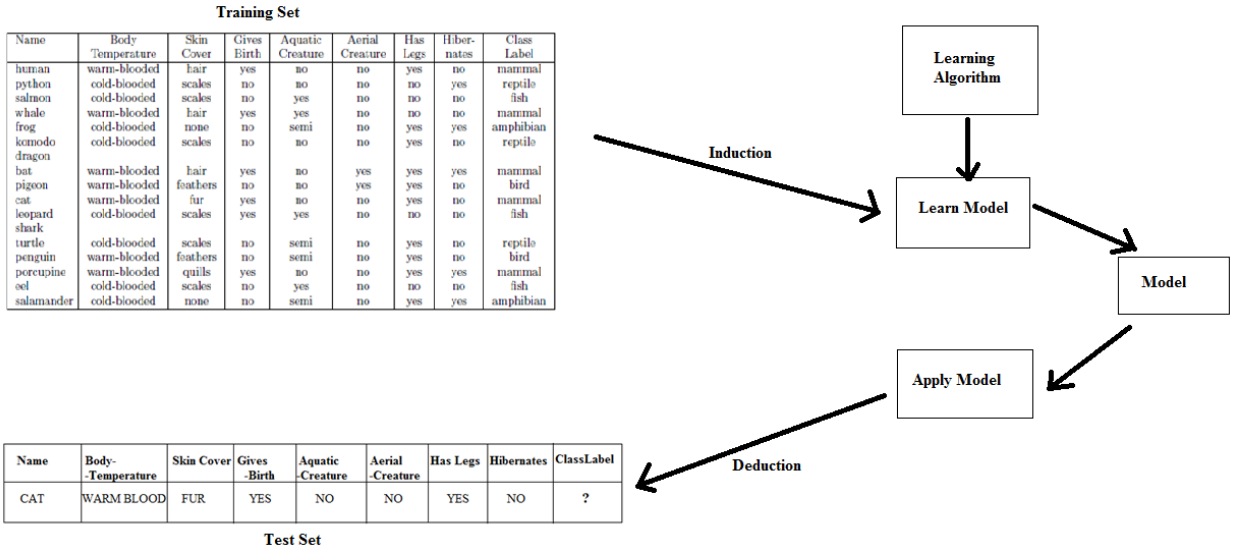
Bayes classifiers

Neural networks

Each technique is with a learning algorithm to identify a best fit model that relates the attribute set x & class label y.

The classification model is said to be fit only if it predicts the correct labels.

The below figure represents a general approach for building a classification model.



The evaluation of the performance of a classification model is based on the number of records correctly or incorrectly predicted by the model.

These predictions can be tabulated in a table known s **confusion matrix.**

Table 4.2. Confusion matrix for a 2-class problem.

		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

Each entry f_{ij} in the table denotes the number of records from class i predicted as class j.

For instance f_{01} means the number of records from class 0 incorrectly predicted as class 1.

Based on the above confusion matrix

The number of correct predictions done by the model is ($f_{11}+f_{00}$)

The number of incorrect predictions done by the model is ($f_{01}+f_{10}$)

The **performance metric** of the classification model can be predicted by **accuracy & error rate.**

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

$$Error\ rate = \frac{Number\ of\ incorrect\ predictions}{Total\ number\ of\ predictions} = \frac{f_{10} + f_{01}}{f_{00} + f_{01} + f_{10} + f_{11}}$$

Decision Tree Induction

We can solve a classification problem by asking a series of questions about the attributes of the test record. Each time we receive an answer and follow up a question until we reach a conclusion about the class label of a particular record.

These serious of questions and their possible answers can be organized in the form of a decision tree, which is a hierarchical structure.

The figure below represents a mammal classification problem which is categorized in two ways mammals & Non Mammals.

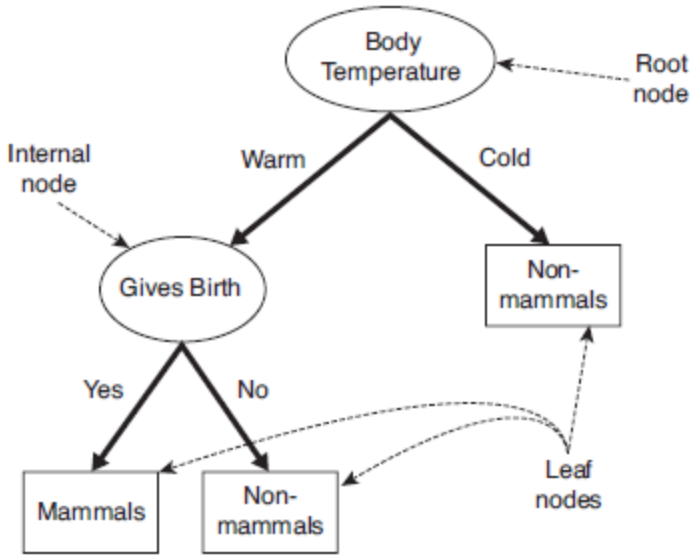


Figure 4.4. A decision tree for the mammal classification problem.

In a decision tree

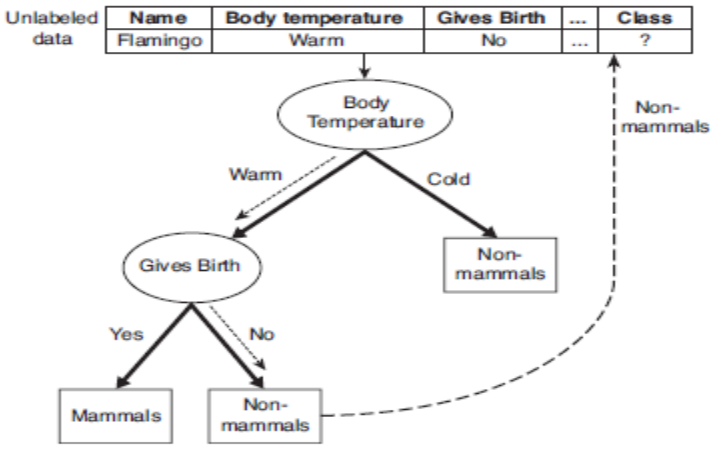
Each **leaf node** is assigned a class label.

The **non terminal** nodes include the root and other internal nodes contain attribute test conditions to separate records that have different characteristics.

Note-

1. A root node has no incoming edges and zero or more outgoing edges.
2. Internal node has exactly one incoming edge and two or more outgoing edges.
3. Leaf or terminal mode has exactly one incoming edge and no outgoing edge.

The figure below Classifies a test record which is straight forward once a decision tree has been constructed. Starting from the root node, we apply the test condition to the record and follow the appropriate branch based on the outcome of the test. It then leads to another internal node for which a new test condition is applied or to a leaf node. The class label associated with the leaf node is then assigned to the record.



How to build a Decision Tree

The algorithms use greedy strategy to build a decision tree for optimum decisions to partitioning the data.

There are some decision tree induction algorithms like ID3,C4.5,CART and also-

-Hunt's algorithm is one such decision tree algorithm.

Hunt's Algorithm

Hunt's algorithm grows a decision tree in a recursive fashion by partitioning the training records into successively purer subsets. Let D_t be the set of training records that reach a node t. The general recursive procedure is defined as below:

1. If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
2. If D_t is an empty set, then t is a leaf node labeled by the default class, y_t
3. If D_t contains records that belong to more than one class, use an attribute test to split the data into smaller subsets.

It recursively applies the procedure to each subset until all the records in the subset belong to the same class. The Hunt's algorithm assumes that each combination of attribute sets has a unique class label during the procedure. If all the records associated with D_t have identical attribute values except for the class label, then it is not possible to split these records any future. In this case, the node is declared a leaf node with the same class label as the majority class of training records associated with this node.

The below in the example to illustrate how the algorithm works-

	binary	categorical	continuous	class
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Figure 4.6. Training set for predicting borrowers who will default on loan payments.

Consider the problem of predicting whether a loan applicant will repay her loan obligations or become delinquent, subsequently defaulting on her loan.

A training set for this problem can be constructed by examining the records of previous borrowers.

In the above Figure each record contains the personal information of a borrower along with a class label indicating whether the borrower has defaulted on loan payments.

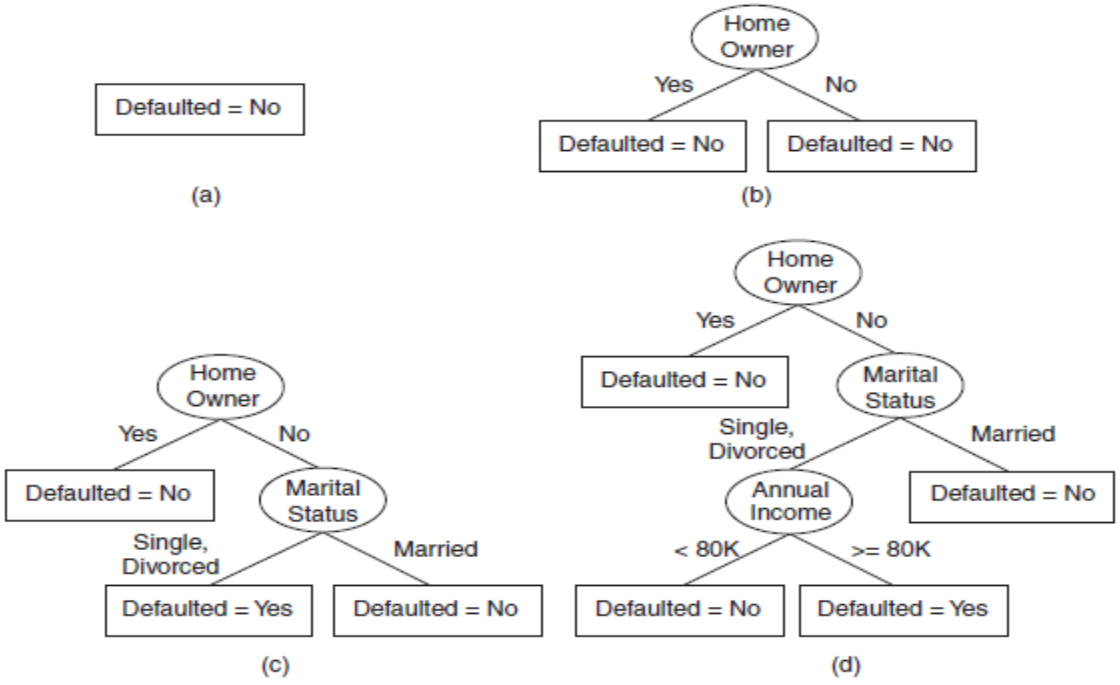


Figure 4.7. Hunt's algorithm for inducing decision trees.

The initial tree for the classification problem contains a single node with class label Defaulted = No (see Figure 4.7(a)), which means that most of the borrowers successfully repaid their loans.

The tree, however, needs to be refined since the root node contains records from both classes. The records are subsequently divided into smaller subsets based on the outcomes of the Home Owner test condition, as shown in Figure 4.7(b).

Now assume that this is the best criterion for splitting the data at this point. Hunt's algorithm is then applied recursively to each child of the root node. From the training set given in Figure 4.6 notice that all borrowers who are home owners successfully repaid their loans.

The left child of the root is therefore a leaf node labeled Defaulted = No (see Figure 4.7(b)).
The right child, we need to continue applying the recursive step of hunt's algorithm until all the records belong to the same class.
The trees resulting from each recursive step are shown in Figures 4.7(c) and (d).

Note-
After building a decision tree a **tree pruning** step can be performed to reduce the size of the decision tree.
Decision trees that are too large are said to be in a phenomenon known as **over-fitting**.

Methods for expressing attribute test conditions-

Decision tree induction algorithms provide method for expressing attribute test condition and its corresponding outcomes for different attribute types.

The test condition for **binary attribute** generates two outcomes as follows-

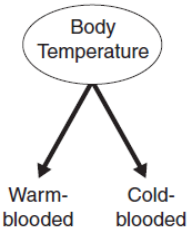
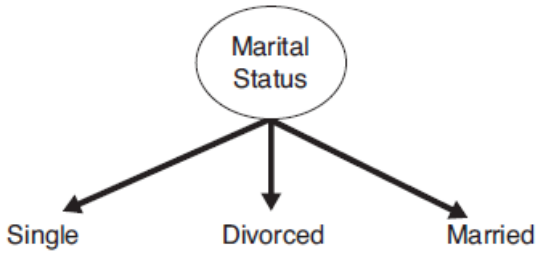


Figure 4.8. Test condition for binary attributes.

The test condition for **nominal attribute** can generate many values in two ways.

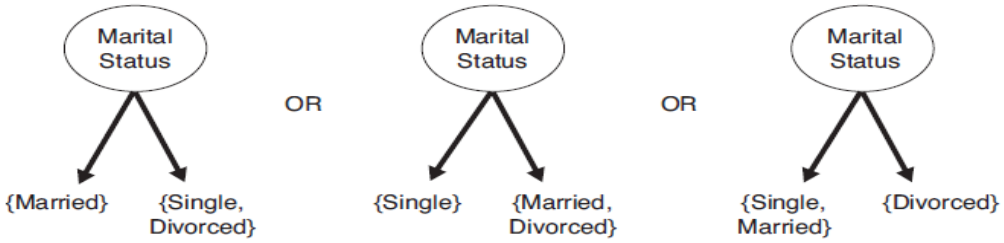
1. A multi-way split in which the number of outcomes depends on the number of distinct values for the corresponding attribute.



(a) Multiway split

2. By producing only binary splits in 2^{k-1}-1 ways of creating a binary partition of k attribute values.

The below are the three different ways of grouping the attribute values for marital status into two subsets.



The test condition for **ordinal attribute** generates outcomes as binary or multi-way splits as follows-

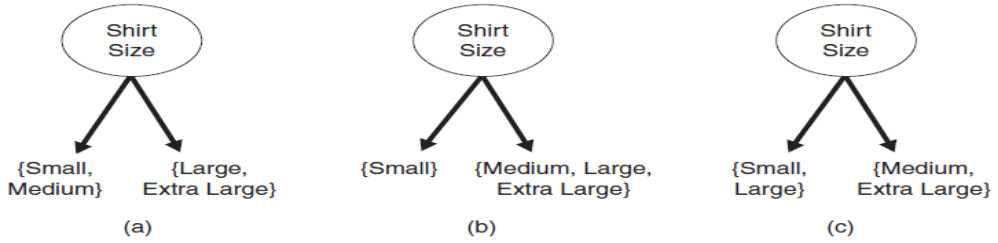


Figure 4.10. Different ways of grouping ordinal attribute values.

The test condition for **continuous attributes** can be expressed as comparison test (A< v) or (A>=v) with binary & multi-way split outcomes.

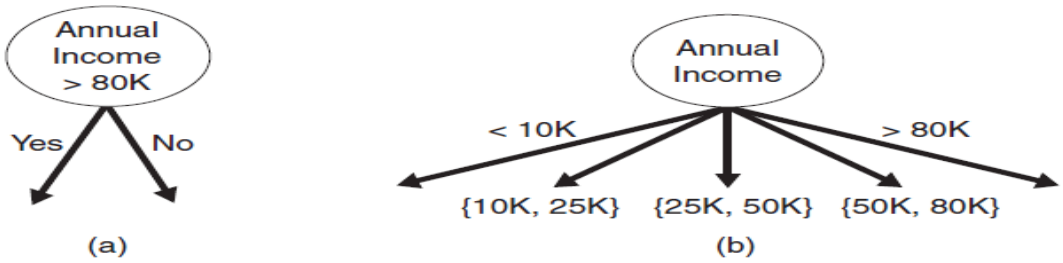


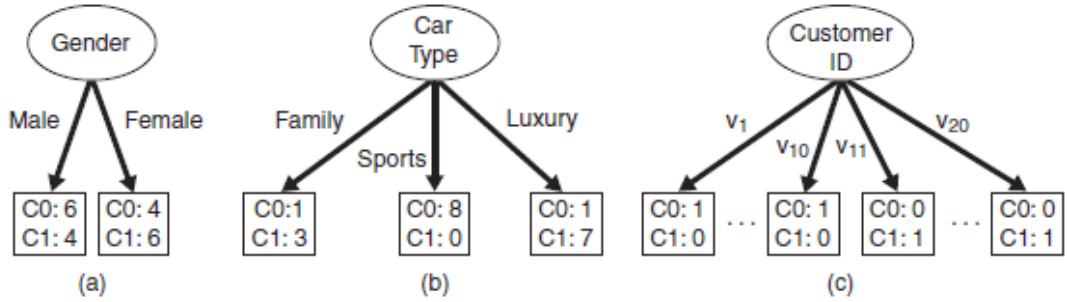
Figure 4.11. Test condition for continuous attributes.

Measures for Selecting the Best Split-

-There are many measures to determine the best way of **splitting the records**.
-These measures can be defined in terms of **class distribution** of records before and after splitting.

Let p (i|t) denote the fraction of records belonging to class i at a given node t.
In a two class problem the class distribution at any node can be written as (p_0, p_1)
where $p_1 = 1- p_0$.

Consider the test condition in the below figure.



- The class distribution before splitting is (0.5, 0.5) because there exists equal number of records from each class.
- If we split the data using gender attribute (figure a) then the class distributions of the child nodes are (0.6, 0.4) and (0.4, 0.6).
- Splitting on the second attribute Car type will result in purer partitions.
- The measures for selecting the best split are based on degree of impurity of the child nodes.
- The smaller degree of impurity means the class distribution is inaccurate.
- For example, a node with class distribution (0, 1) has zero impurity. Whereas, a node with uniform class distribution (0.5, 0.5) has highest impurity.
- The impurity of class distribution can be done measured as:

$$\text{Entropy(t)} = -\sum_{\text{i=0}}^{\text{c}-1} \text{p(i|t)}\log_2 \text{p(i|t)}$$

$$Gini(t) = 1 - \sum_{\text{i=0}}^{\text{c}-1} [\text{p(i|t)}]^2$$

$$Classification\ error(t) = 1 - \max[p(i|t)]$$

where c is the number of classes and $0 \log_2 0 = 0$ in entropy calculations.

To determine the performance of the test condition we need to compare the degree of impurity of the parent node (before splitting) with the degree of impurity of the child nodes (after splitting).

The larger their difference means better test condition.

Here, to determine goodness of the split we use gain Δ i.e.,

$$\Delta = \text{I (parent)} - \sum_{j=1}^{k} \frac{\text{N}(\text{V}_\text{j})}{N} \text{I}(\text{V}_\text{j})$$

Where I(.) is the impurity measure of a node,

N is the total number of records at the parent node,

K is the number of attribute values,

$\text{N}(\text{V}_\text{j})$ is the number of records associated with the child node V_j,

Note-

1. Decision tree induction algorithms choose a test condition that maximizes the gain Δ.
2. Maximizing the gain is equivalent to minimizing the weighted average impurity of child node.
3. When entropy is used as an impurity measure then the difference in entropy is known as **information gain** Δ_{info}.

Characteristics of Decision Tree Induction-

1. Decision tree induction is a non parametric approach for building classification models which means it does not require any prior assumptions regarding the type of probability distributions satisfied by the class and other attributes.
2. Finding an optimal decision tree is an NP-complete problem. Many decision tree algorithms employ a heuristic based approach to guide their search in the vast hypothesis space.
3. Techniques developed to construct decision trees are computationally inexpensive. But once tree has been built classifying a test record is extreme fast.
4. Smaller sized trees are easy to interpret and also accurate when compared to other classification techniques.
5. Decision trees provide an expressive representation for learning discrete valued functions and do not work well on certain types of Boolean problems.
6. Decision tree algorithms are robust to the presence of noise, especially for the methods to avoid over fitting.
7. The presence of redundant attributes does not affect the accuracy of decision trees and the irrelevant attributes may be accidently chosen may lead to a larger decision tree than necessary. (here feature selection technique can help to improve the accuracy by eliminating irrelevant data)
8. Since decision tree algorithms use a top down or recursive partitioning approach where the number of records becomes smaller as we traverse down the tree to take a significant decision about the class representation which is said to be **data fragmentation problem**. The solution for this is to stop splitting the records below a certain threshold.
9. A sub-tree can be replicated(same sub tree can appear at different branches) multiple times in a decision tree which makes the decision tree more complex to interpret.
10. During the tree growing procedure the attribute partitioning process builds border between two neighboring regions of different classes is known as **decision boundary**. A data set cannot be classified effectively by decision tree induction algorithm by using single attribute at a time. To overcome this limitation we can use oblique decision tree because it allows test conditions that involve more than one attribute.

Design Issues of Decision Tree Induction-

1. How should the training records be split?

   In this the each recursive step of tree growing process must select an attribute test condition to divide the records into smaller subsets.

   To implement this step the algorithm must provide a method for specifying the test condition for different attributes and also a measure for evaluating the goodness of each test condition.
2. How to stop splitting procedure?

   A stopping condition is needed to terminate the tree growing process.

To implement this step we have to continue expanding a node until either all the records belong to the same class or all the records have identical attributes.

Model Over-fitting

There are two types of errors occurred by a classification model:

1. Training error or re-substitution error or apparent error which is the number of misclassification errors committed on training records.
2. Generalization error which is the expected error of the model on previously unseen records.

A good classification model must have low training errors and low generalization errors.

Because a model that fits the training data will be with less generalization error when compared to a model with high training error. This situation is known as model over-fitting. The training and test error rates of a model are large when the size of the tree is very small. This situation is known as model under-fitting.

The below are some of the causes for model over-fitting:

1. Over-fitting due to presence of noise.
2. Over-fitting due to lack of representative samples.
3. Over-fitting due to multiple comparison procedure.
   (for detailed explanation refer 4.4.1,4,4,2,4,4,3 in data mining vipin kumar text book from pgno-175to178)

To **estimate the generalization error** of the induced tree there are occur some methods-

1. **Using re-substitution estimate**- This approach assumes that the training set is a good representation of the overall data so that it can be used to provide an optimistic estimate for the generalization error.
2. **Pessimistic error estimate**- This approach explicitly computes generalization error as, the sum of training error and a penalty term for model complexity.

Let n(t) be the training records classified by node t and e(t) be the number of misclassified records then the pessimistic error estimate $e_g(T)$ is as follows-

$$e_g(T) = \frac{\sum_{i=1}^{k}[e(t_i) + \Omega(t_i)]}{\sum_{i=1}^{k} n(t_i)} = \frac{e(T) + \Omega(T)}{N_t}$$

Where k is the number of leaf nodes

e(T) is the overall training error of the decision tree

N_t is the number of training records and $\Omega(t_i)$ is the penalty associated with each node t_i.

3. **Minimum description length principle**-This is an information theoretic approach to incorporate model complexity.

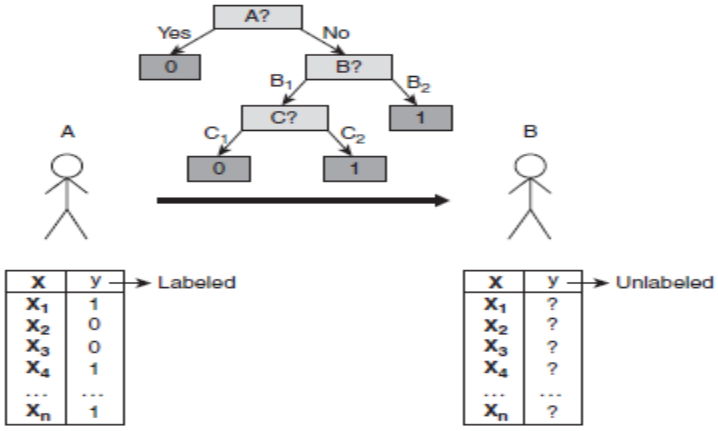


Figure 4.28. The minimum description length (MDL) principle.

In the above figure both A and B are given a set of records with known attribute values x. In addition, person A knows the exact class label for each record, while person B knows none of this information.

B can obtain the classification of each record by requesting that A transmits the class labels sequentially. Such a message would require O(n) bits of information, where n is the total number of records.

Alternatively A may decide to build a classification model that summarizes the relationship between x and y. The model can be encoded in a compact form before transmitted to B.

If the model is 100% accurate then, cost of transmission is equivalent to cost of encoding the model. Otherwise A must also transmit information about which record is classified incorrectly by the model.

Thus the overall cost of transmission is

Cost(model, data) = Cost(model) + Cost(data|model)

Cost(model) means cost of encoding the model,
Cost(data|model) means cost of encoding the mislabeled records.

According to MDL principle we have to obtain a model that is with minimum cost function.

4. **Estimating Statistical Bounds-** The generalization error can be estimated as a statistical correction to training error. Since generalization error is larger than training error, the statistical correction is usually computed as an upper bound to the training error by considering the number of training records that reach a particular leaf node.

5. **Using a validation set-** In this approach instead of using training set to estimate the generalization error, the original training data is divided into two smaller subsets. One subset is for training set and the other set is said to be validation set which is for estimating generalization error. This approach can be used to obtain models with different levels of complexity. The complexity of the best model can be estimated by

adjusting the learning algorithm until the algorithm attains lower error rate on validation set.

Note- typically two-third of training set is reserved for model building and one-third is used for error estimation.

Model over fitting in decision trees can be avoided using two strategies-

1. **Pre pruning-** In this approach the tree growing algorithm stop expanding a leaf node when observed gain in impurity measure falls below certain threshold. So that it avoids generating overly complex sub trees that over fits the training data.

   Note-It is difficult to choose right threshold for early termination. Too high threshold will result in under fitted models, while a too low threshold will result in over fitting problem. If no significant gain obtained means then subsequent splitting may result in better sub trees.

2. **Post pruning-** In this approach the tree pruning step will be performed by trimming a fully grown tree in a bottom up fashion.
   This trimming can be done by replacing a sub tree with
   1- A new leaf node whose class label is determined from the majority class of records affiliated with sub tree.
   2- The most frequently used branch of the sub tree.

   Note- Post pruning is better than pre pruning because it takes the pruning decisions based on fully grown tree unlike pre pruning, which can suffer from pre mature termination of the tree growing process.

Evaluating the performance of a classifier

To compare the relative performance of different classifiers, the accuracy or error rate computed from the test set can be used.

There are some methods commonly used to evaluate the performance of a classifier.

1. Holdout Method
2. Random Sampling
3. Cross Validation
4. Boot Strap

Holdout Method- In this method the original data is partitioned into two disjoint sets called the training and the test sets. A classification model is then induced from the training set and its performance is evaluated on the test set.

The accuracy of the classifier can be estimated based on the accuracy of the induced model on the test set.

Limitations of holdout method-

1. Fewer records are available for training because some of the records are held for testing.
2. The model is highly dependent on composition of training and test sets.

Note- If the training set size is small then the variance of model is large. If the training set size is large then the accuracy computed from smaller test set is less reliable. Therefore the training and test sets are no longer independent to each other, because they are subsets of original data.

Random Sub sampling- The repetition of holdout method several times to improve the estimation of a classifier's performance is known as random sampling.

Limitations-

1. This method does not utilize as much data as possible for training.
2. It also has no control over the number of times each record is used for testing and training.

Cross-Validation-

In this approach each record is used same number of times for training and exactly once for testing.

In this method suppose we partition the data into two equal sized subsets then we choose one of the subsets is for training and the other for testing and the swap the roles so that training set becomes test set and vice versa. This approach is called a two-fold cross validation. The total error is obtained by summing up the errors for both runs.

So far assume that the training records are sampled without replacement as a result there are no duplicate records in the training and test sets.

Bootstrap- In this approach the training records are sampled with replacement i.e., a record already chosen for training is put back into the original pool of records so that it is equally likely to be redrawn. Records that are not included in the bootstrap sample become part of the test set. The model induced from the training set is then applied to the test set to obtain an estimate of the accuracy of the bootstrap sample.

Please refer text book for detailed explanation