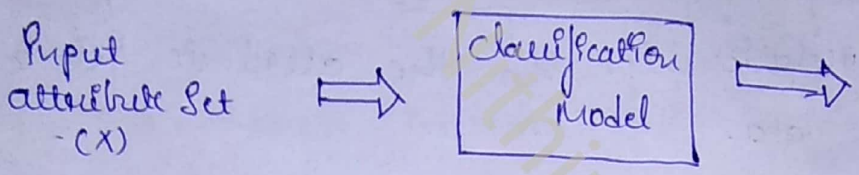
* Classification :-

Classification is the task of learning a target function "f" that maps each attribute set "x" to one of predi-fined class label "y"



The target function is known informally as "classification Model"

* descriptive Modelling :- A classification Model can serve as an Explanatory tool to distinguish btw objects of diff class.

→ for ex it would be useful for both biologists & others to have a descriptive Model that summarizes data shown

Name	Body temp	Skin cover	Gives Birth	Aquatic creature	Aerial creature	has legs	class label
Human	warm	hair	yes	NO	NO	yes	Mammals
python	cold	Scales	NO	NO	NO	NO	reptiles
frog	cold	none	no	Semi	no	yes	Amphibians
whale	warm	hair	yes	yes	NO	NO	Mammal

* predictive Modelling :- A classification Model can also be used to predict the class label of unknown records.

→ A classification Model can be treated as a "black-box" that automatically assigns a class label when presented with attribute set of an unknown record.

Name	Body temp	Skin cover	Gives Birth	Aquatic creature	Aerial creature	has legs	class label
gila Monster	cold	Scales	NO	NO	NO	yes	Reptile

we use a classification Model built from data set to determine the class to which the creature belongs.

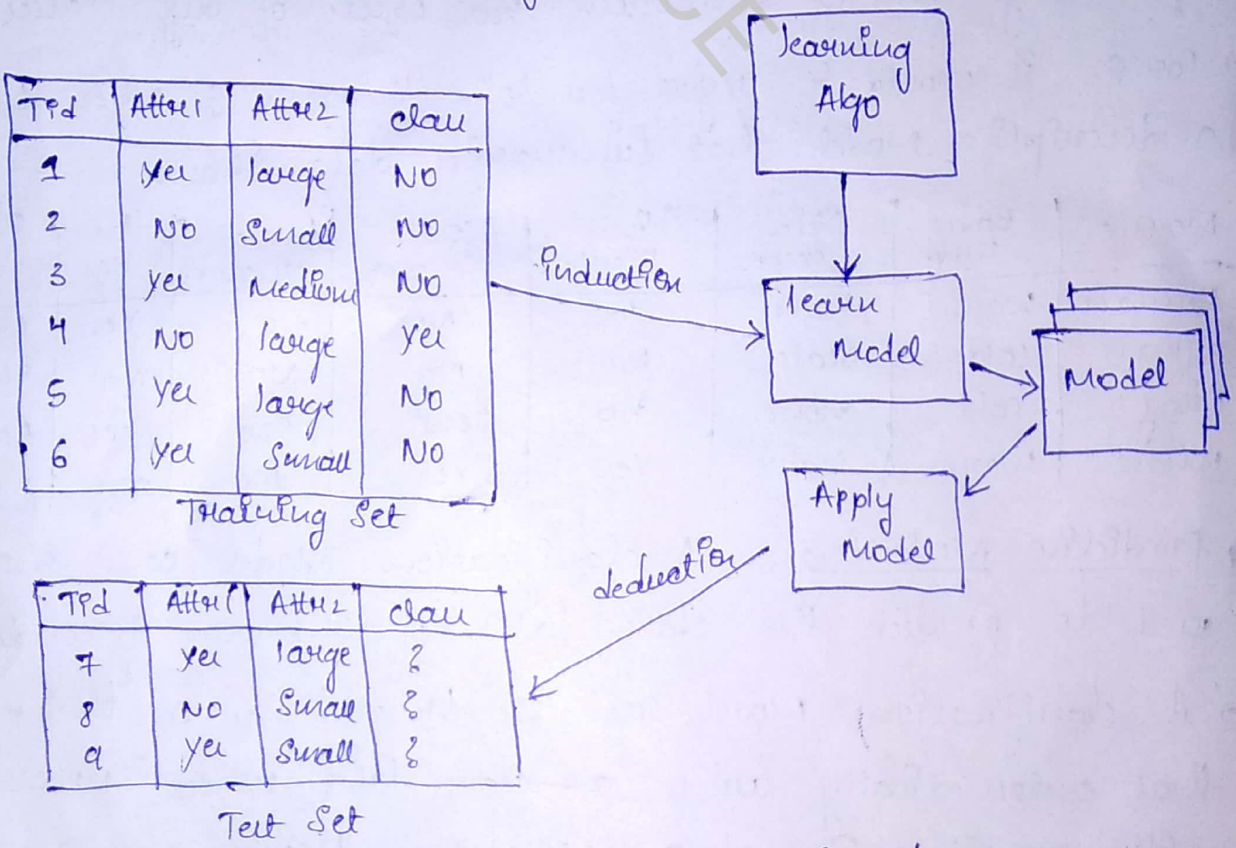
*General Approach to Solving a classification problem

A classification technique (or classifier) is a systematic approach to building classification model from an input set data.

Ex: decision tree classifier, Rule-based classifier Etc

→ Each technique employs a "learning Algo" to identify a model that best fits the relationships b/w the attribute set & class label of input data.

→ the model generated by learning Algo should both fit the input data well & accurately predict the class labels of records it has never before.



The Evaluation of performance of a classification model is based on the counts of test records correctly & incorrectly predicted by the model.

These counts are tabulated in a table known as "confusion Matrix"

		Predict class	
		class = 1	class = 0
Actual class	class = 1	F_{11}	F_{10}
	class = 0	F_{01}	F_{00}

Table depicts the confusion Matrix for a binary classification problem, Entry F_{ij} in this table denotes the no of records from class i predicted to be of class j

→ The performance Metric such as accuracy which is defined as follows

* Accuracy = $\frac{\text{No of correct predictions}}{\text{total no of predictions}} = \frac{F_{11} + F_{00}}{F_{10} + F_{11} + F_{00} + F_{01}}$

The performance of a Model can be Expressed in terms of the Error-rate which is given by

* Error_rate = $\frac{\text{No of wrong predictions}}{\text{total no of predictions}} = \frac{F_{10} + F_{01}}{F_{01} + F_{10} + F_{11} + F_{00}}$

Most classification Algo seeks Models that attain the "highest accuracy or Equivalently lowest Error rate" when applied to test Set.

* decision tree Induction :- which is a simple yet widely used classification technique.

→ The tree has three types of nodes

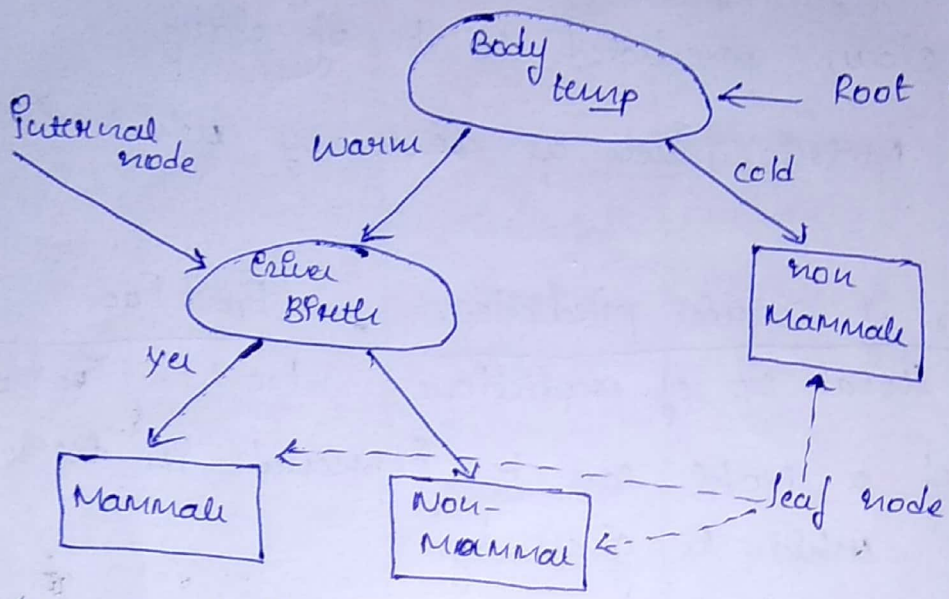
* root node → that has no incoming Edges & Zero or more outgoing Edges

* Internal node → Exactly one incoming Edge & two or more outgoing Edges.

* leaf or terminal → nodes Each of which has Exactly one incoming Edge & no outgoing Edge.

In a decision tree Each leaf node is assigned to class label, the non-terminal node which include the root & other internal node contain attribute test condition to separate records that have diff characteristics.

→ consider the Ex as shown below



* How to Build a decision tree :- One such Algo is "Hunts Algo" which is the basis of many existing decision tree induction Algo including ID3, C4.5, & CART

* Hunts Algo :- In hunts Algo, a decision tree is grown in a recursive fashion by partitioning the training records into successively purer subsets

→ let D_t be the set of training records that are associated with node t & $y = \{y_1, y_2, \ldots y_c\}$ be class labels the foll is recursive defn of Hunts Algo

* Step1 :- If all records in data belong to some class y_t then t is the leaf node labeled as y_t

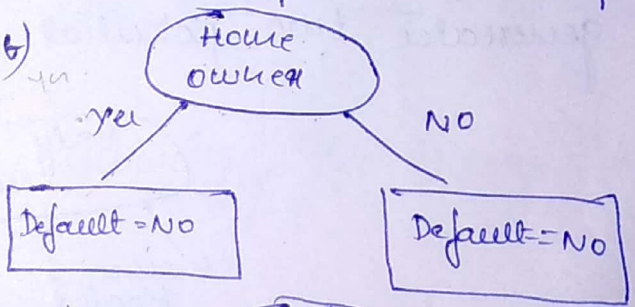
* Step2 :- If data contains records that belong to more than one class, an attribute test condition is selected

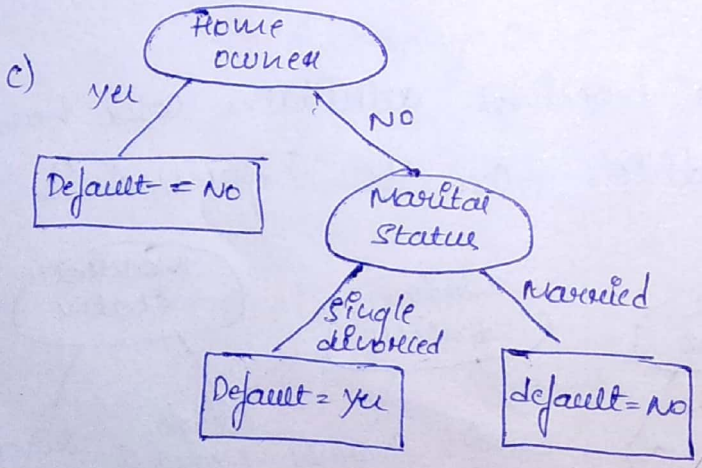
to partition the records into smaller subsets.
A child node is created for each outcome of test condition & records in Dt are distributed to children based on outcomes.

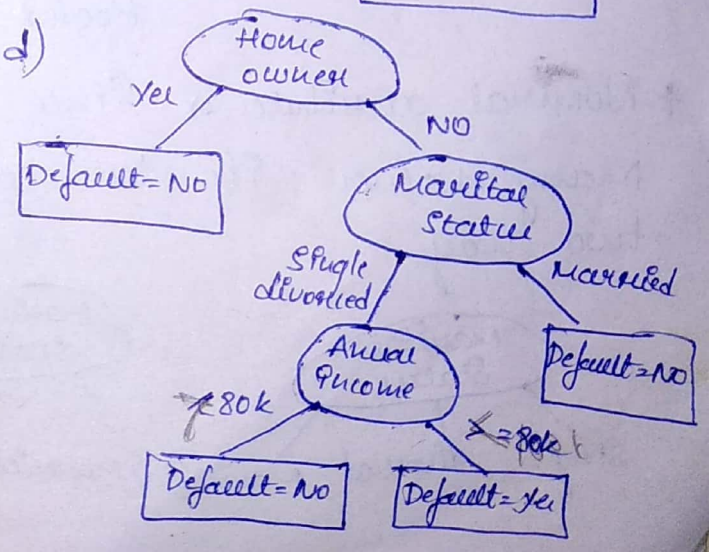
→ consider the following example problem of predicting whether loan Applicant will repay the loan

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	70k	Yes

a) Default = No

b)


c)


d)


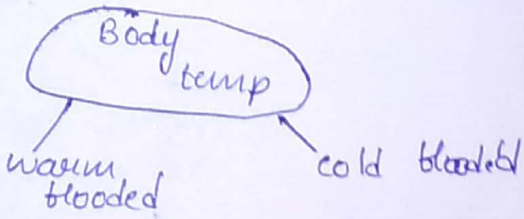
Additional condition are needed to handle the fall cases.

* It is possible for some of child nodes created in step 2 to be empty
  i.e there are no records associated with these nodes.
* In step 2 if all the records associated with Dt have identical attribute values then it is not possible to split them furthers.

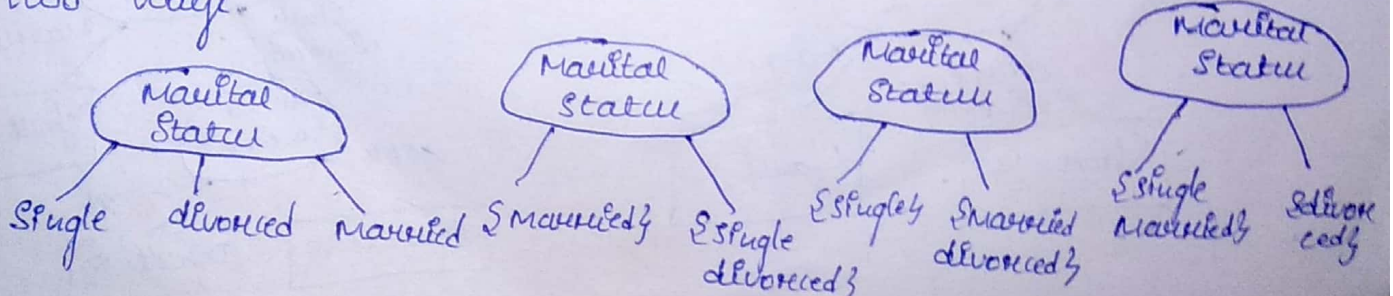
* <u>design issues of decision tree induction</u>- A learning Algo for inducing decision tree must address the fall two issues.
  * How should the training records be split?
    → Attribute test condition
  * How should the splitting procedure stop?
    → Stopping condition.

→ <u>Methods for Expressing Attr test condition</u>

* Binary attributes → the test condition for binary attr generates two potential out comes

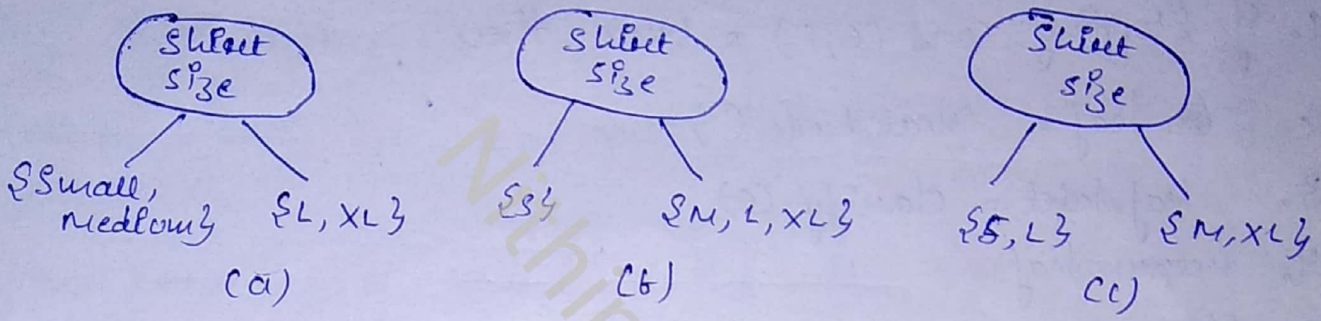


* Nominal attributes → Since a Nominal attribute can have many values the test condition can be expressed in two ways.

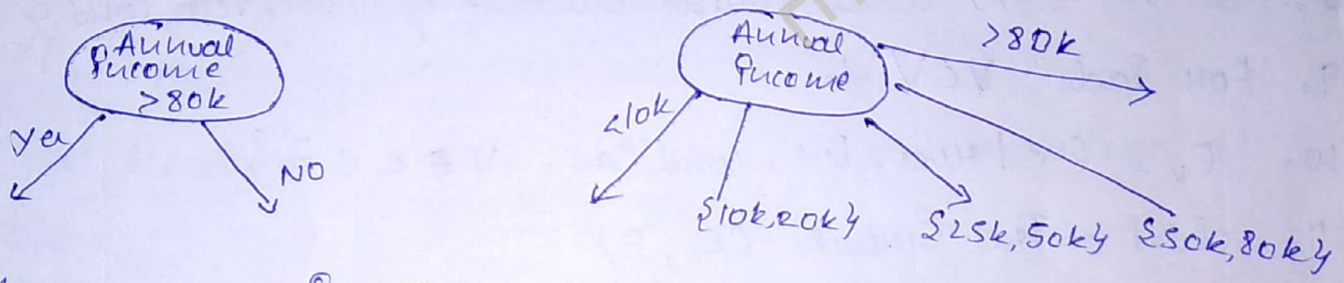


* Ordinary attributes → can also produce binary or Multiple Splits.

Ordinal attribute values can be grouped as long as the grouping does not violate the order property of attribute values



* Continuous attribute → For continuous attributes, the test condition can be expressed as comparison test ($A < V$) or ($A \geq V$) with binary outcomes



→ Measures for Selecting the Best Split :- Measures developed for selecting the best split are often based on "degree of impurity of child nodes"

→ The smaller the degree of impurity, the more skewed the class distribution

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

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

* Classification Error $= 1 - \max_i [p(i/t)]$

* Algorithm for decision tree Induction

The Input to this Algo consists of training records "E" & attribute set "F"

Tree Growth (E, F)

1. If Stopping_cond (E, F) = true then
2. leaf = Create Node ()
3. leaf.label = classify (E)
4. return leaf
5. Else
6. root = Create Node ()
7. root.test_cond = Find_best_Split (E, F)
8. let V = {v/v is a possible outcome of root.test_cond}
9. For Each $v \in V$ do
10. E_v = {e | root.test_cond (e) = v & e ∈ E}
11. child = Tree Growth (E_v, F)
12. add child as descendant of root & label the edge as v
13. End For
14. End IF
15. return Root.

After building the decision tree "Tree-pruning" step can be performed to reduce the size of decision tree.

→ decision tree that are too large are susceptible to phenomenon known as "overfitting"

* Characteristics of decision tree Induction

* It is non-parametric Approach for building classification Model.

* decision tree are computationally inexpensive making it possible to quickly construct even when training set size is very large.
* decision trees, especially smaller-sized tree are relatively easy to interpret.
* decision tree Algo are quite robust to presence of Noise.
* The presence of redundant attr does not adversely affect the accuracy of decision tree.
* Since decision tree use "top-down" recursive Approach the no of records become smaller.
* "Sub tree" can be replicated many times.
* "test condition" involve using only one attribute at a time.
* Model overfitting :- The Errors committed by a classification Model are generally divided into two types.
  * Training Errors
  * generalization Errors.
* "Training Error" also known as "resubstitution Error" or "Apparent Error" is the no of misclassification Errors committed on training records.
* "generalization Error" is the Expected Error of the model on previously unseen records.

A good Model Must have low training Error as well as low generalization Error.
This is important Because a Model that fits the training

* Boot Strap :- Boot Strap approach, the training records are sampled with replacement, i.e a record already chosen from training is put back into the original pool of records so that it is equally likely to be redrawn.

→ A boot Strap sample size N contains about 63.2% of records in orig data.

This Approximation follows from the fact that probability a record chosen by boot Strap sample is

$$1 - (1 - 1/N)^N$$

$$1 - \varepsilon^{-1} = 0.632$$

The sampling procedure repeated b times to generate b boot Strap samples.

* Methods for comparing classifiers

* Estimating a confidence interval for accuracy :- To determine the confidence interval, we need to Establish the probability distribution that governs the accuracy measure.

→ Following is a list of characteristics of binomial Exp

* Exp consists of N independent trails, where trail has two possible outcomes → Success
→ failure

* the probability of Success p, in Each trail is constant.

→ An Ex of a binomial Exp in counting no of heads that turn up when coin flipped N times

$$P(X = v) = \binom{N}{p} p^v (1-p)^{N-v}$$

where X is no of Success observed in N trails.

→ Based on Normal distribution, the following confidence interval for acc can be derived

$$P\left(-Z_{\alpha/2} \leq \frac{acc - p}{\sqrt{p(1-p)/N}} \leq Z_{1-\alpha/2}\right) = 1 - \alpha$$

* Comparing the performance of two Models :- Consider a pair of Models M_1 & M_2 that are Evaluated on two Independent test Sets D_1 & D_2

→ let n_1 denote no of records in D_1 & n_2 denote no of records in D_2

Suppose the Error rate for M_1 on D_1 is e_1

the Error rate for M_2 on D_2 is e_2

→ Therefore the Variance of d can be computed as follows

$$\sigma_d^2 \approx \hat{\sigma}_d^2 = \frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}$$

where $e_1(1-e_1)/n_1$ & $e_2(1-e_2)/n_2$ are the Variance of Error rates finally at $(1-\alpha)$ % confidence level.

* Comparing performance of two classifiers :- Suppose we want to compute the performance of two classifiers using k-fold cross-validation approach.

→ Initially data set D is divided into k- Equal Sized partitions, we then apply Each classifier to Construct a Model from k-1 of partitions & test it on remaining partition. This Step repeated k times.

$$\hat{\sigma}_{dv}^2 = \frac{\sum_{j=1}^{k} (d_j - \bar{d})^2}{k(k-1)}$$

& the observed difference.

where $\bar{d}$ is average difference, For this approach we need to use a t-distribution to compute the confidence interval for d_t^{cv}

$$d_t^{cv} = \bar{d} \pm t_{(1-\alpha),k-1} \hat{\sigma}_{d^{cv}}.$$

* Rule-Based classifier :-

A rule-based classifier is a technique for classifying records using a collection of "if... then..." rules.

→ The rules for the Model are represented in a disjunctive normal form $R = (r_1 \vee r_2 \vee \ldots r_k)$ where R is known as "rule set" & "r_i" are classification rules or disjuncts.

→ Example of rule set for vertebrate classification problem

r_1 : (Gives Birth = no) $\wedge$ (Aerial Creature = yes) → Birds

r_2 : (Gives Birth = no) $\wedge$ (Aquatic Creature = yes) → Fishes

r_3 : (Gives Birth = yes) $\wedge$ (Body temp = warm) → Mammals

r_4 : (Gives Birth = No) $\wedge$ (Aerial Creature = No) → Reptiles

→ Each classification rule can be Expressed in following way

$$r_i : (\text{Condition}_i) \rightarrow y_i$$

The left-hand side of the rule is called "rule antecedent" or "precondition" it contains a conjunction of attribute tests

$$\text{condition}_i = (A_1 \text{ op } V_1) \wedge (A_2 \text{ op } V_2) \wedge \ldots (A_k \text{ op } V_k)$$

where (A_j, V_j) is an attribute-value pair & "op" is a logical operator chosen from set $\{=, \neq, <, >, \leq, \geq\}$

Each attribute test $(A_j \text{ op } V_j)$ is known as "conjunct".

→ The right-hand side of the rule is called the "rule consequent" which contains predicted class y_i.

→ A rule r covers a record x if the precondition matches the attribute of x.
r is also said to be fired or triggered whenever it covers a given record.

→ The quality of a classification rule can be evaluated using measures such as
* coverage
* accuracy

* The coverage of rule is defined as fraction of records in D that triggers rule r.

$$\text{coverage}(r) = \frac{|A|}{|D|}$$ → the fraction of records that satisfy the antecedent

* The accuracy or confidence factor is defined as the fraction of records triggered by r whose class labels are equal to y.

$$\text{Accuracy}(r) = \frac{|A \cap y|}{|A|}$$ fraction of records that satisfy both antecedent & consequent

where |A| → no of records that satisfy rule antecedent
|A ∩ y| → no of records that satisfy both antecedent & consequent
|D| → total no of records.

* How a rule-based classifier works:-
A rule-based classifier classifies a test record based on rule triggered by the record.

→ Consider the rule set for all vertebrates

Name	Body temp	Skin cover	Gives birth	Aquatic creature	Aerial creature	Has legs	Hibernates
lemur	warm	Fur	yes	No	No	yes	yes
turtle	cold	Scales	no	Semi	no	yes	no
dogfish shark	cold	Scales	yes	yes	No	No	No

* The first vertebrate which is lemur, is warm blooded & gives birth to its young, it triggers rule r3 & thus

is classified as a Mammal.

* The second vertebrate, which is turtle, triggers rules r_4 & r_5, their conflicting classes must be resolved.

* None of the rules are applicable to dogfish shark.

→ Mutually Exclusive rules :- The rules in rule set R are Mutually Exclusive if no two rules in R are triggered by same record.

This property ensures that every record is covered by at most one rule in R.

→ Exhaustive rules :- A rule set R has Exhaustive coverage if there is a rule for each combination of attribute values This property ensures that every record is covered by at least one rule in R.

Assuming that "Body temp" & "Gives Birth" are binary variables

→ Example of Mutually Exclusive & Exhaustive rule set

r_1 : (Body temp = cold) → Non-Mammals

r_2 : (Body temp = warm) ∧ (Gives Birth = yes) → Mammals

r_3 : (Body temp = warm) ∧ (Gives Birth = No) → Non-Mamma

→ Ordered rules :- In this approach, the rules in rule set are ordered in decreasing order of their priority, which can be defined in many ways (coverage, accuracy etc)

* This avoids the problem of having conflicting classes predicted by Multiple classifier rules.

→ Unordered rules :- This approach allows a test record to trigger Multiple classification rules & considers the consequent of each rule as a vote for particular class. The record is usually assigned to class that receives the highest no of votes.

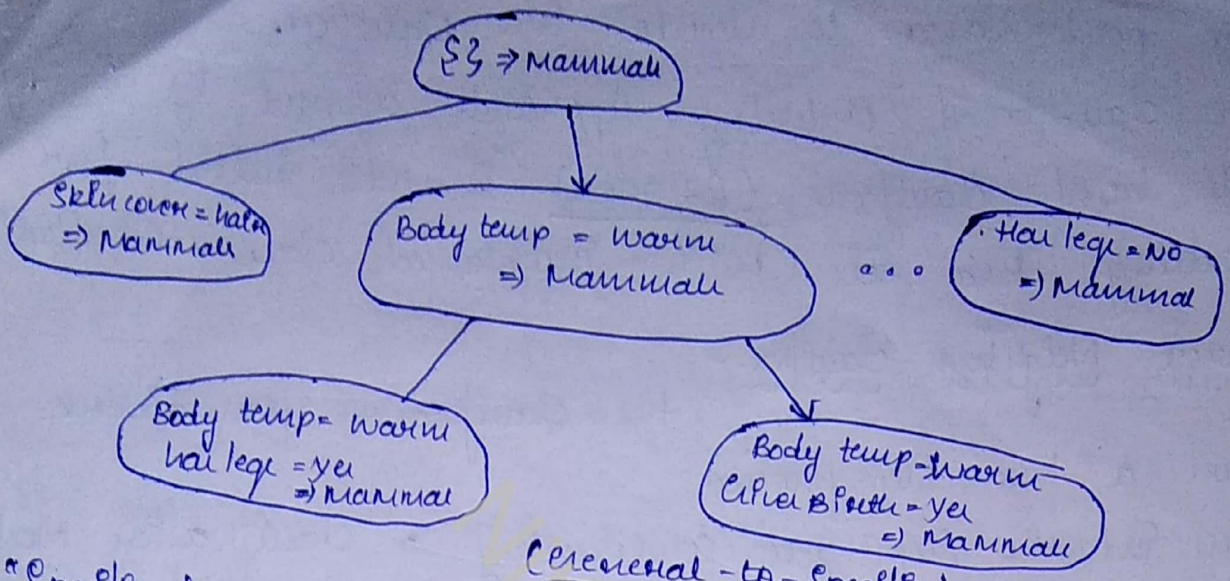
* Direct Methods for Rule Extraction :- "Sequential covering" Algorithm is often used to extract rules directly from data.
Rules are grown in a greedy fashion based on certain Evaluation Measure.

→ Sequential covering Algo is as shown

1. Let E be the training records & A be set of attribute value pairs $\{(A_j, v_j)\}$
2. Let Y_o be an ordered set of class $\{y_1, y_2, \ldots y_k\}$
3. Let R = { } be initial rule list
4. For Each class $y \in Y_o - \{y_k\}$ do
5. While Stopping condition is not met do
6. r ← learn-one-rule (E, A, y).
7. Remove training records from E that covered by r
8. Add r to the bottom of rule list : R → R ∨ r
9. End while
10. End For
11. Insert the default rule { } → y_k to the bottom of rule list R.

→ Rule-Growing Strategy :- There are two common Strategy for growing a classification
   * general-to-Specific
   * Specific-to-general

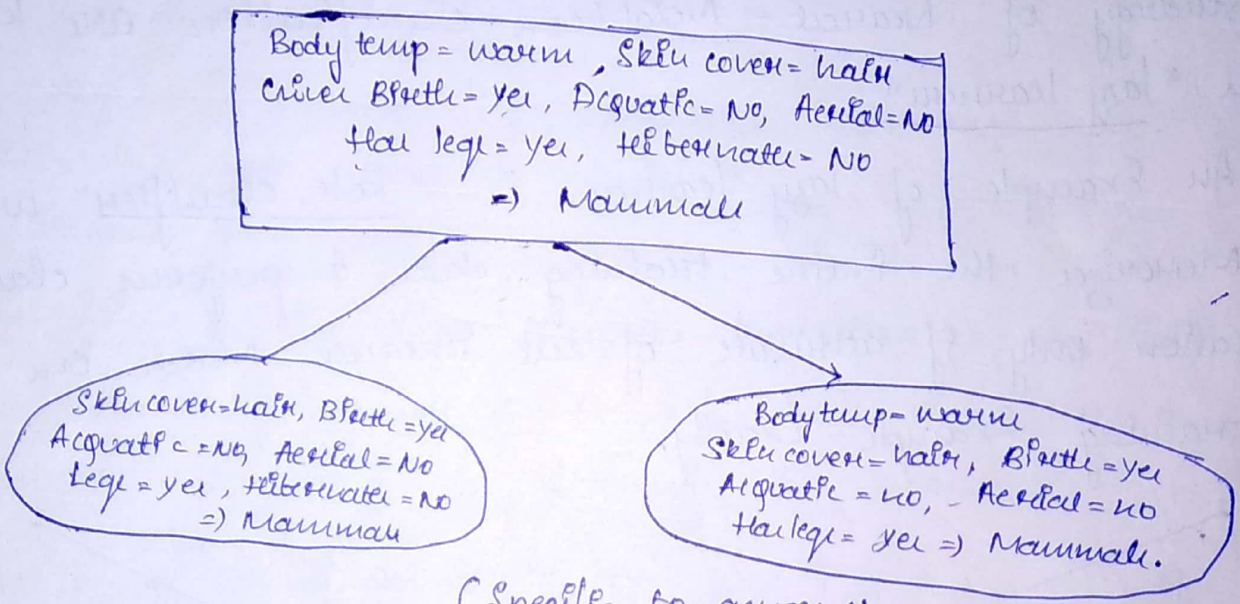
* "general-to-Specific" Strategy in which an initial rule r: { } → y is created, where the left-hand side is an Empty set & right-hand side contains the target class. The rule has poor quality because it covers all Ex in training set.



(General-to-Specific)

* "Specific-to-general" Strategy, one of the positive Examples is Randomly chosen as Initial Seed for rule-growing process.

During the refinement Step, the rule is generalized by removing one of its conjuncts so that it can cover more positive Examples.



(Specific-to-general)

* Characteristics of Rule-based classifier

* The Expressiveness of rule set is almost Equivalent to that of a decision tree

Because a decision tree can be represented by a set of Mutually Exclusive & Exhaustive rules.

* Rule-based classifiers are generally used to produce descript-ive Models that are Easier to Interpret, but give comp

-arble performance to decision tree classifiers.

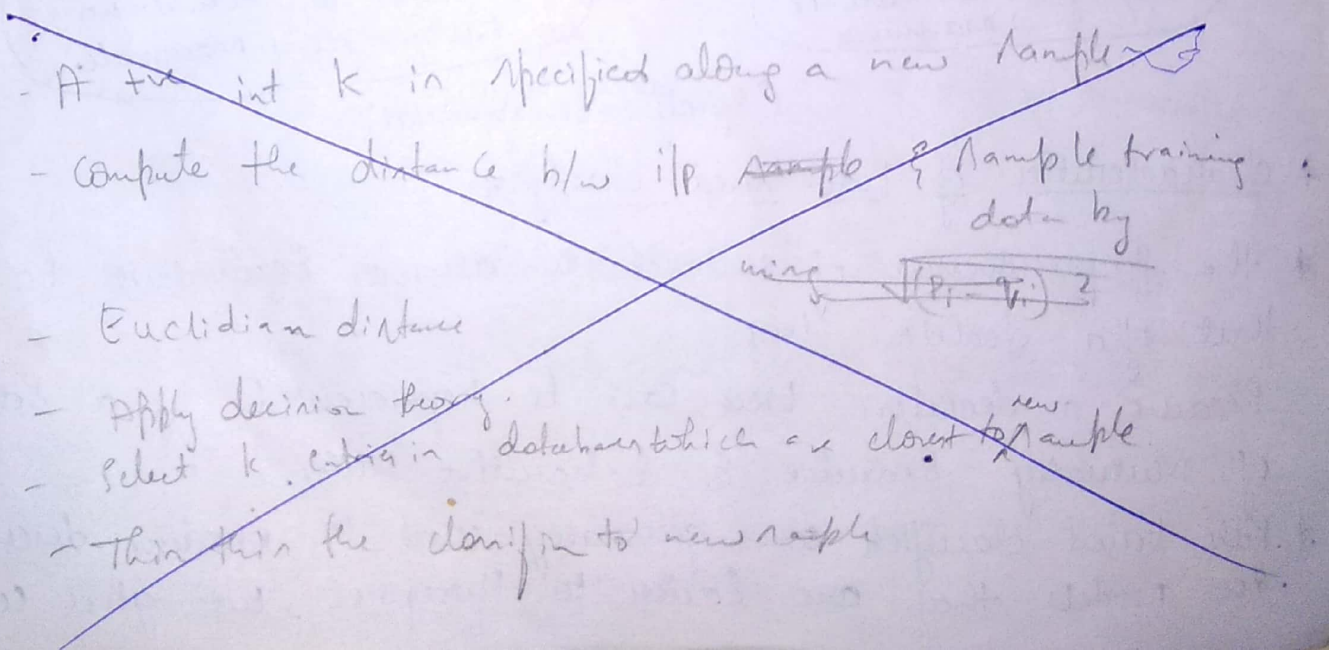
* the class-based ordering Approach adopted by many rule-based classifiers (RIPPER) is well suited for handling data sets with unbalanced class distributions.

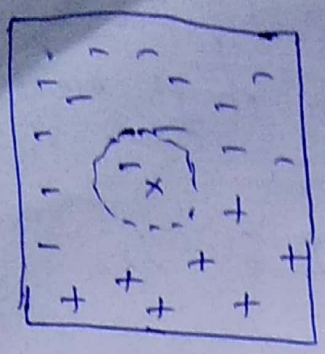
* Nearest-Neighbour classifiers:- the classification framework involves a two-step process

  * an inductive step for constructing a classification model from data.
  * a deductive step for applying model to test Example

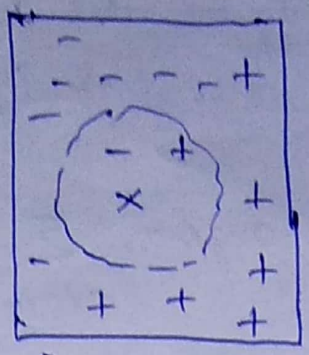
→ "decision tree classifiers" & "rule-based classifiers" are good Example of "Eager learners"
And Mean while the techniques that Employ this Strategy of Nearest-Neighbour classification are known as "lazy learners"

→ An Example of lazy learner is "Rote classifiers" which Memorizes the Entire training data & perform classification only if attribute of test instance Match one of training Example Exactly.

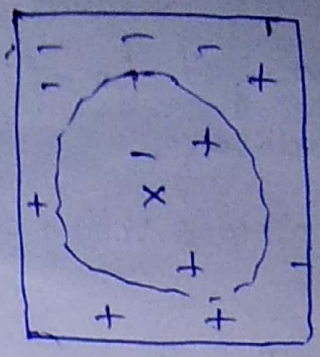




1- nearest neighbour



2-nearest neighbour



3-nearest neighbour

→ One way to make this approach more flexible is to find all training Ex that are relatively similar to the attribute of test Example, which are known as "nearest neighbours" can be used to determine the class labels of test Example.

→ the 1-, 2-, 3- nearest neighbours of data point located at the center of each circle, the data point is classified based on the class labels of its neighbours.

In the case where the neighbours have more than one label, the data point is assigned to majority class of its nearest neighbours.

* Algorithm :- the Algo computes the distance or similarity btw each test Example $z = (x', y')$ & all the training Examples $(x, y) \in D$ to determine its nearest-neighbour list.

1. Let k be no of nearest neighbours & D be set of training Ex
2. For Each test Example $z = (x', y')$ do
3. compute $d(x', x)$, the distance btw z & Every Ex $(x, y) \in D$
4. Select $D_z \subseteq D$, the set of k closest training Ex of to z
5. $y' = \arg\max_v \sum_{(x_i, y_i) \in D_z} I(v = y_i)$
6. End For.

→ Once the nearest-neighbor list is obtained, the test ex is classified based on the majority class of its nearest neighbors

* Majority voting $y' = \text{argmax} \sum_{(x_i, y_i) \in D_z} I(v = y_i)$

→ using the distance-weighted voting scheme the class label can be determined as follows

* distance-weighted voting $y' = \text{argmax} \sum_{(x_i, y_i) \in D_z} w_i \times I(v = y_i)$

* Characteristics of Nearest Neighbor classifiers

* Nearest-Neighbor classification is part of a more general technique known as "Instance-based learning" which uses specific training instances to make prediction without having to maintain an abstraction derived from data.
* lazy learners such as nearest-neighbor classifiers do not require model building.
* Nearest-Neighbor classifiers make their prediction based on local information.
* Nearest-neighbor classifiers can produce arbitrarily shaped decision boundaries.
  the decision boundaries of nearest-neighbor classifiers have "high variability" because they depend on the composition of training examples
* Nearest-Neighbor classifiers can produce wrong predictions unless the appropriate proximity measures & data preprocessing steps are taken.

* Bayesian classifiers:-

* Bayes theorem :- It is a statistical principle for combining prior knowledge of class with new evidence gathered from data.

→ let X & Y be a pair of random variables, Their joint probability $P(X=x, Y=y)$ refers to the probability that variable X will take on the value x & variable Y take on value y.

The joint & conditional probabilities for X & Y are related in following way

$$P(X,Y) = P(Y|X) \times P(X) = P(X|Y) \times P(Y)$$

Rearranging the expression leads to formula known as "Bayes theorem"

$$P(Y|X) = \frac{P(X|Y)\,(P(Y))}{P(X)}$$

* Using Bayes theorem for classification :- let X denote the attribute set & Y denote the class variable, If the class variable has non-deterministic relationship with attributes then we can treat X & Y as random variables & capture their relationship using $P(Y|X)$.

This conditional probability is also known as "posterior probability for Y", as opposed to its "prior probability $P(Y)$"

→ during the training phase, we need to learn the posterior probability $P(Y|X)$ for every combination of X & Y based on info gathered in training data.

By knowing these probabilities, a test record X' can be classified by finding class Y' that Maximizes posterior probability

$-p(y'|x')$.

→ lets consider the following training set.

Tid	Home owner	Marital status	Annual Income	Defaulted Borrower
1	Yes	Single	125k	No
2	No	Married	100k	No
3	Yes	Single	85k	No
4	No	divorced	60k	No
5	No	Single	120k	Yes
6	No	Single	100k	No
.	.	.	.	.
.	.	.	.	.
10	No	Married	80k	Yes

Suppose we are given a test record with following attribute set X = (Home owner = No, M.status = Married, A.Income = 120k)

To classify the record we need to compute posterior probabilities $p(Yes|X)$ & $p(No|X)$ based on the info available in training data.

If $p(Yes|X) > p(No|X)$ then the record is classified as "Yes" otherwise "No"

→ Bayes theorem is useful because it allows us to express posterior probability $p(Y|X)$ in terms of prior probability $p(Y)$.

The "class-conditional" probability $p(X|Y)$ & Evidence $p(X)$

$$p(Y|X) = \frac{p(X|Y) \times p(Y)}{p(X)}$$

When comparing the posterior probabilities for diff values of X, the denominator term $p(X)$ is always constant & thus can be ignored.

* Naive Bayes classifier :- Estimate the class-conditional probability by assuming that ~~that~~ the attributes are condi-tionally independent, given the class label y.

The conditional independence assumption can be formally stated as follows

$$P(X|Y=y) = \prod_{i=1}^{d} P(X_i|Y=y)$$

where each attribute set $X = \{X_1, X_2 \ldots X_d\}$ consists of d attrib-utes.

→ Conditional Independence :- let X, Y & Z denote three set of random variables, the variables in X are said to be conditionally independent of Y, given Z, if the following condition holds

$$P(X|Y,Z) = P(X|Z).$$

→ characteristics of Naive Bayes classifiers

* they are robust to isolated noise points because such points are averaged out when estimating conditional probabilities from data
* they are robust to irrelevant attributes.
* correlated attributes can degrade the performance of classifiers because the conditional independence no longer holds for such attributes.

* Bayesian Belief Networks (BBN) :-

provides a graphical representation of the probabilistic relationships among a set of random variables

There are two key elements of Bayesian N/w

* A directed Acyclic graph (DAG) Encoding the dependence relationships among a set of variables.

* A probability table associating each node to the limited -rate parent nodes.

→ Consider three random variables A, B, C in which A & B are independent variables & each has a direct influence on third variable.

The relationship among the variables can be summarized into the directed acyclic graph as shown

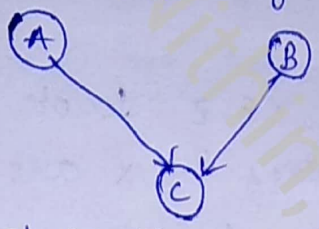


fig: Representation probabilistic relationship using directed acyclic graph.

→ Conditional Independence :- A node in a BBN is conditionally independent of its non-descendants, if its parents are known.

* If a node X does not have any parents, then the table contains only prior probability P(X).
* If a node X has only one parent Y, then the table contains conditional probability P(X|Y).
* If a node X has multiple parents {Y1, Y2, ---Yk} then table contains conditional probability P(X|Y1, Y2, -Yk).

→ Algorithm

1. Let T = {X1, X2, ---Xd} denote a total order of variables
2. for j=1 to d do
3. Let $X_{T(j)}$ denote the jth highest order variable in T
4. Let $\pi(X_{T(j)}) = \{X_{T(1)}, X_{T(2)} --- X_{T(j-1)}\}$ denote set of variables preceding $X_{T(j)}$
5. Remove variables from $\pi(X_{T(j)})$ that dont affect X_j
6. Create an arc from $X_{T(j)}$ & removed variable
7. End for.

→ Characteristics of BBN

* BBN provides an approach for capturing prior knowledge of particular domain using graphical Model.
* Construction of N/w can be time consuming & require a large amount of Effort
* BBN are well suited to dealing with in complete data
* Because the data is combined probabilistically with prior knowledge, the Method is quite robust to Model overfitting.