Chapter - 3

Classification

ta mining

Grouping

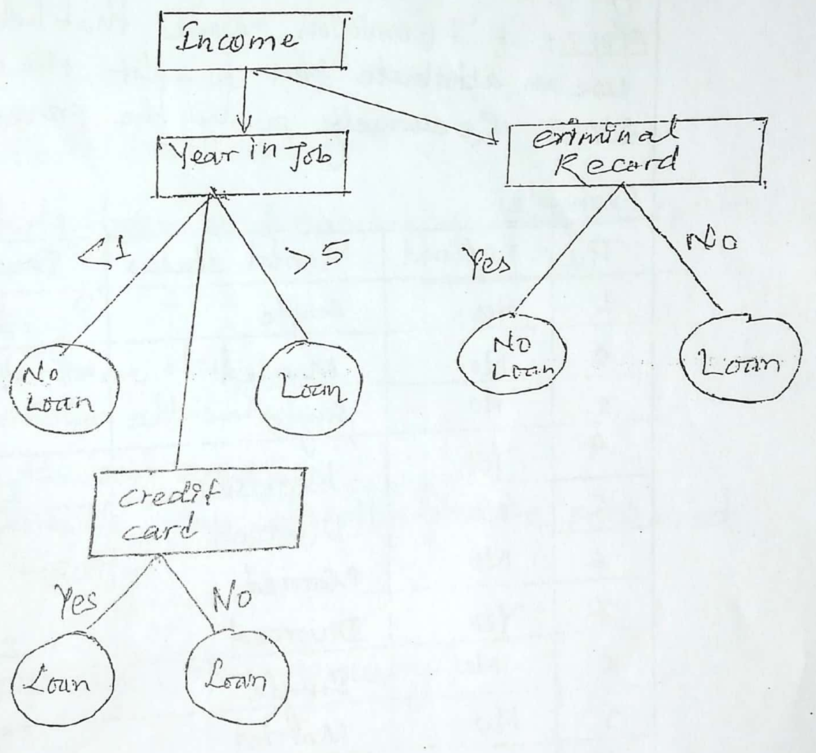
(1) **Decision Tree:**

→ A decision tree is a structure in which each branch n represents a choice between a number of alternatives and ea leaf node represent a classification or decision

→ Decision tree is a classifier in the form of a tree struc where each node is either a leaf node, indicating a cla: instances or a decision node that represents some test to carried out on a single attribute value with one branch subtree for each possible outcome of tree.

→ It can be used to classify an instance by starting a of the tree and moving through it until leaf node, which provides the classification of the instances.

Example :>



Advantage:-

→ Extremely fast for classifying unknown records

→ Easy to interprete

→ High accuracy for simple data.

Some Common Decision Tree Algorithms Are:

- Hunts Algorithm
- CART
- ID3, J48, C45
- SLIQ, SPRINT

Hunts Algorithm:

Let D_t be the set of training records that reach at node T.

Algorithm:

Step 1: If 'D_t' contains records that belongs to the same class then the node 'T' is a leaf node

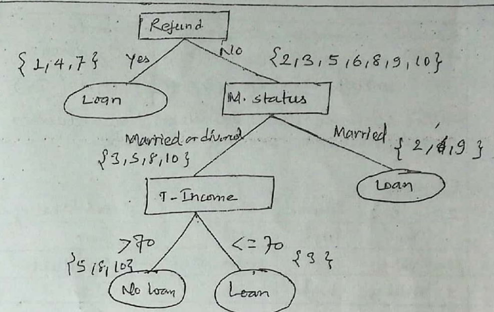
Step 2: If 'D_t' is an empty set then 'T' is a leaf node labelled by default class

Step 3: If 'D_t' contains records that belongs to more than one class use an attribute test to split the data into smaller subsets

Step 4: Recursively apply the procedure to each subset.

Example:-

Tid	Refund	Marital status	Taxable income	class
1	Yes	Single	125K	Loan
2	No	Married	100K	Loan.
3	No	Single	70K	Loan
4	Yes	Married	120K	Loan
5	No	Divorced	95K	no loan
6	No	Married	60K	Loan
7	Yes	Divorced	220K	Loan
8	No	Single	85K	No loan
9	No	Married	75K	loan
10	No	Single.	90K	No loan.



Tree Induction:-

Greedy Strategy:

→ split the record based on an attribute test that optimizes certai

Issues

(i) How to split record?

(ii) How to specify attribute test condition?

→ depends on attribute type and number of ways to split (i.e multiple way split.)

(iii) when to stop splitting?

→ All records belong to same class.

→ All records have similar attributes.

(iv). How to determine the best split?

→ Nodes with homogeneous class distribution are preferred

→ Measure of node impurity

Measure of Node Impurity (Homogeneity)

- Gini Index
- Entropy calculation.

Gini Index:

the Gini Index measures the impurity of data set (D) as,

$$Gini(D) = 1 - \sum_{i=1}^{n} P_i^2$$

where, P_i → is the probability that a touple in D belongs to cl C_i.

Scanned by CamScanner

→ Consider a binary split for each attribute
→ when D is partitioned into D_1 and D_2 then,

$$Gini(D) = \frac{D_1}{D} Gini(D_1) + \frac{D_2}{D} Gini(D_2)$$

→ The attribute that maximizes the reduction in impurity is selected as the splitting attribute.

Example:-

ID	Age	Income	student	credit_rating	class (buy-comp
1	Youth	high	No	fair	No
2	Youth	high	No	Excellent	No
3	Middle age	high	No	Fair	Yes
4	Senior	medium	No	fair	Yes
5	Senior	low	Yes	Fair	Yes
6	senior	low	Yes	Excellent ~~fair~~	No
7	M.age	Low	Yes	Excellent	Yes
8	Youth	medium	Yes	fair ~~Excellent~~	No
9	Youth	low	Yes	fair	Yes
10	Senior	medium	Yes	Fair	Yes
11	Youth	medium	Yes	Excellent	Yes
12	M-age	medium	No	Excellent	Yes
13	M-age	high	Yes	fair	Yes
14	senior	medium	No	Excellent	No

Solution:-

(buys-computer = Yes) = 9
(buys-computer = No) = 5

Impurity D, $Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = \underline{0.459}$

Now calculating Gini Index for each attribute

Attribute: Income;

Subset 1, $D_1 = \{$low, medium$\}$, $D_2 = \{$High$\}$

$$Gini(Income) = \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2)$$

$$= \frac{10}{14}\left(1 - (7/10)^2 - (3/10)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - (2/4)^2\right)$$

[Here, 7 buys computer from set D_1 & 3 dont buys computer]

$= 0.556$

Subset 2: $D_1 = \{$low, high$\}$, $D_2 = \{$medium$\}$

$$Gini(Income) = \frac{8}{14}\left(1 - (5/8)^2 - (3/8)^2\right) + \frac{6}{14}\{1 - (4/6)^2 - (2/6)^2$$

$= 0.4582$

Subset 3: $D_1 = \{$medium, high$\}$, $D_2 = \{$low$\}$

$$Gini(income) = \frac{10}{14}\{1 - (6/10)^2 - (4/10)^2\} + \frac{4}{14}\{1 - \left(\frac{1}{4}\right)^2 - \left(\frac{3}{4}\right.$$

$= 0.450$

Since, subset 3 has minimum value so selected as bina split for income.

Attribute: Age;

Subset 1: $D_1 = \{$youth, senior$\}$, $D_2 = \{$M. aged$\}$

$$Gini(age) = \frac{10}{14}\{1 - \left(\frac{5}{10}\right)^2 - \left(\frac{5}{10}\right)^2\} + \frac{4}{14}\{1 - \left(\frac{4}{4}\right)^2\}$$

$= 0.357$

Subset 2: $D_1 = \{$youth, m-aged$\}$, $D_2 = \{$senior$\}$

$$Gini(age) = \frac{9}{14}\{1 - (3/9)^2 - \left(\frac{6}{9}\right)^2\} + \frac{5}{14}\{1 - (3/5)^2 - (2/5)^2\}$$

$= 0.3678$

Subset 3: $D_1 = \{$youth$\}$, $D_2 = \{$mage, senior$\}$

$$Gini(age) = \frac{5}{14}\{1 - (3/5)^2 - (2/5)^2\} + \frac{9}{14}\{1 - (7/9)^2 - (2/9)^2\}$$

$= 0.3936$

Attribute: Student;

$D_1 = \{$Yes$\}$, $D = \{$No$\}$

$$Gini(Student) = 7/14\{1 - (6/7)^2 - \left(\frac{1}{7}\right)^2\} + \frac{7}{14}\{1 - (4/7)^2 - (3/7)^2\}$$

$= 0.3775$

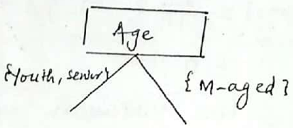
Attribute: credit rating; $D_1 = \{$fair$\}$, $D_2 = \{$Excellent$\}$

$$Gini(credit\ rating) = \frac{8}{14}\{1 - (2/8)^2 - (6/8)^2\} + \frac{6}{14}\{1 - (3/6)^2 - (3/6)^2$$

$= 0.4285$

Impurity reduction for age = $0.459 - 0.357 = 0.102$
Impurity reduction for income = $0.459 - 0.450 = 0.009$
Impurity reduction for student = $0.459 - 0.3775 = 0.0815$
Impurity reduction for credit-rating = $0.459 - 0.4285 = 0.0305$

Since, Age has maximum impurity reduction it is selected as root n



Nearest Neighbour classifier:

→ It uses K-closest points for performing classification. K neare of record X' are data points that have the K-smallest to X.

→ Classification based on learning by analogy i.e by comp given test tuple with training tuple that are similar

→ Training tuples are described by an attribute

→ When given an unknown tuple, a nearest neighbour cla searches the pattern space for K-training tuples - are closest to the unknown tuple.

→ Nearest neighbour classifier requires;

(i) Set of stored record.

(ii) distance matrix to compute the distance between reco ~~(iii)~~ Such as euclidean distance.

(iii) The value of K i.e. the number of nearest neighbour t

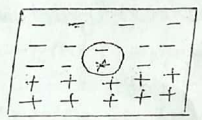
→ To classify an unknown records;

(i) Compute the distance to other training record

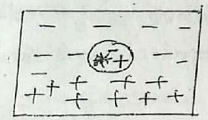
(ii) Identify the K-nearest neighbour

(iii) Use class labels of nearest neighbour to determine cla of unknown record by using majority vote.

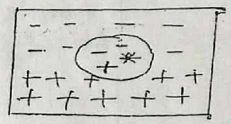
Labels



K = 1
Class = -



K = 2
class = -/+



K = 3
Class = -

classification Issue;

(a) Choosing the value of K:

If K is too small it is sensitive to noise point. If K is too larg neighbour may include points from other class.

(b) Scaling Issue:

Attributes may have to be scale to prevent distance measure fro being dominated by one of attributes
eg. Height, weight.

(c) Distance computing for non-numeric data:

Use distance as 0 (minimum) for same data 1 (maximum) for different data set.

6) missing values:

use minimum possible distance.

Disadvantages:

→ poor accuracy when data have noise and irrelevent attribut

→ classifying unknown records are relatively expensive.

→ slow when classifying test tuple.

Rule Based classifier:

It classify record by using a collection of if---then--- rule. i.e Rule based classifier uses a set of 'If ...then" rules of classification, the 'If' part or lefthand side of rule is known as the rule anticident where as the 'then' part or right hand side is known as rule consequent. In rule anticident the condition consists of one or more attribute test.

eg. R_1: {(Age= Youth) ∧ (student = yes)}= (buys_computer = yes)

If the condition in the rule anticident holds true for a given tuple the rule anticident is satisfied and the rule covers the touple i.e. converge rule. If the rule fraction of record that satisfy the anticident of rule,

$$\text{Coverage} = \frac{n_{covers}}{n_{\text{total dataset}}} \times 100\%$$

The accuracy of rule is the fraction of record that satisfy both the anticident and consequent of rule. i.e.

$$\text{Accuracy} = \frac{n_{correct}}{n_{covers}} \times 100\%$$

How does rule based classifier work?

1. If a rule is satisfied by a tuple the rule is said to be trigger
2. If only one rule is satisfied, then the rule fire by returning th class prediction for the tuple.
3. Triggering doesnt always means firing because there may be mor than one rule that can be satisfied.
4. If more than one rule is satisfied then the rule fires by retur ng the class by applying conflict resolution strategy to fin which rule is fired.

Conflict Resolution Strategy:

(1) when more than one rule is triggered rule ordering or ranking is ap the rule odering may be class based or rule based. when rul is used the rule set is given as a decision less.

(2) when no rule is satisfied by tuple-X (unknown) then, a def rule can be setup to specify a default class.

Example:

Name	Blood type	Give birth	can fly	Lives in water	clas
Lemor	warm	Yes	no	no	mam
Turtle	cold	No	no	Sometimes	Rep Amphi
shark	cold	Yes	no	Yes	acqu

Rules:

R_1: (Give birth = yes) ∧ (blood type = warm) = mammals

R_2: (Give birth = no) ∧ (can fly = no) = Reptile

R_3: (Lives in water = sometime) = Amphibian

→ A lemor triggers R_1. so, its a mammal

→ A turtle triggers R_2 and R_3 i.e conflict

→ A shark triggers non of the rule, use default class

→ If conflict; apply rule ranking.

Characteristics of Rule based classifier.

1. Mutually Exclusive rule: (ordering)

Classifier contains mutually exclusive rule. If rules are independe to each other.

(i) Every record is covered by almost one rule.

(ii) Rules are no longer mutually exclusive if record may trigger r than one rule.

(iii) To make mutually exclusive we apply ordering.

2. Exhaustive rule: (default)

→ Classifier has exhaustive coverage if it accounts for every possible combinations of attribute values.

→ Each record is covered by almost one rule

→ Rules are no longer exhaustive if a record may not trigger any r

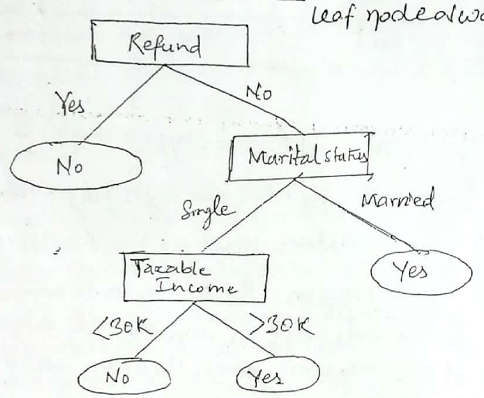
→ To make exhaustive we apply default class.

Building Classification Rules:

1. Direct Method: (Sequential & Inductive Approach)
   → Extract rule direct from data.
   eg. RIPPER, CN2
2. Indirect Method:
   → Extract rule from other classification.
   Eg. Decision Tree, ANN.

Rule extraction from Decision tree.

Leaf node always class



Rules:
R1: (Refund = Yes) = No
R2: (Refund = No) ∧ (Marital status = Married) = Yes

Rule simplification:-
R2 can be simplified as
(Marital status = Married) = Yes.

Advantages of Rule based classifier:
→ Highly expressive
→ Easy to generate and interprete
→ can classify new instances rapidly.
→ High performance.

Rules simplification

$$P(A/B) = \frac{P(B/A) * P(A)}{P(B)}$$

posterior, likelihood, preposition, evidence

Bayesian Classifier:
→ It is statistical classifier which ~~provides~~ predicts class membership probabilities for a membership class.
→ It has high accuracy and speed for large databases.
→ It has minimum error rate in comparison to other clas
→ Based on Baye's Law.

$$P(A/B) = \frac{P(A \cap B)}{P(B)}, \quad P(B/A) = \frac{P(A \cap B)}{P(A)}$$

Types:
(1) Bayesian Belief Networks (Graphical Models)
→ It allows dependencies among subset of attribute i.e has conditional dependencies among the attribute.
→ Has complex computational complexity.

(2) Naive Bayesian Classifier
→ It assumes that the effect on an attribute value on a class is independent of the value of other attributes i.e conditional independence
→ It has simple computational complexity.
→ Let 'D' be the training setup tuples and $C_1, C_2, \ldots, C_n$ their associated classes.
Given a tuple 'x', the classifier will predict that x ... to the class having highest posterior probability conditioned ... i.e. the native Bayesian classifier predicts that the tuple ~~belong~~ belong to the class C_i only if

$$P(C_i/x) > P(C_j/x) \text{ for } 1 < i \leq m, j \neq i$$

$$\text{i.e } P(C_i/x) = \frac{P(x/C_i) \cdot P(C_i)}{P(x)} \quad \text{(maximum)}$$

$P(C_i/x) = \frac{P(x/C_i) \times P(C_i)}{P(x)}$

$P(x^{k}/C_i)$

Here, $P(x)$ = Constant
$P(C_i) = P(C_1) = P(C_2) = \ldots = P(C_m)$

So, we need to maximize $P(X/C_i)$

Since, for native assumption is class conditional impedance

$$P(X/C_i) = \prod_{k=1}^{n} D(x_k/C_i)$$

$$= D(x_1/C_i) * D(x_2/C_i) * \ldots * D(x_n/C_i)$$

These probability can be calculated from training tuple.

Example :-

ID	Age	Income	Student	Credit Rating	class buy computer
1	Youth	High	No	Fair	No
2	Youth	High	No	Excellent	No
3	M-age	High	No	fair	Yes
4	Senior	Medium	No	Fair	Yes
5	Senior	Low	Yes	fair	Yes
6	Senior	Low	Yes	Excellent	No
7	m-age	Low	Yes	Excellent	Yes
8	Youth	Medium	No	fair	No
9	Youth	Low	Yes	fair	Yes
10	Senior	Medium	Yes	Fair	Yes
11	Youth	Medium	Yes	Excellent	Yes
12	m-age	Medium	No	Excellent	Yes
13	M-age	High	Yes	fair	Yes
14	senior	Medium	No	Excellent	No

Find class of X:

X = (age = Youth, income = medium, student = yes, creditrating = fair)

Solution :-

Let, C_1 (buys_computer = Yes) = 9

C_2 (buys_computer = NO) = 5

Prior probability of $C_1 = 9/14 = 0.642$

Prior Probability of $C_2 = 5/14 = 0.357$

P (Age = youth / buys_computer = yes) = $2/9$ = 0.222

P (Age = Youth / buys_computer = No) = $3/5$ = 0.6

P (Income = medium / C_1) = $4/9$ = 0.444

P (income = medium / C_2) = $2/5$ = 0.4

P (student = Yes / C_1) = $6/9$ = 0.666

P (student = Yes / C_2) = $1/5$ = 0.2

P (credit_rating = fair / C_1) = $6/9$ = 0.666

P (credit_rating = fair / C_2) = $2/5$ = 0.4

$P(X/C_1) = 0.222 \times 0.444 \times 0.666 \times 0.666 = 0.044$

$P(X/C_2) = 0.6 \times 0.4 \times 0.2 \times 0.4 = 0.019$

Since, $P(X/C_1) > P(X/C_2)$

∴ X is classified as buys_computer = yes.

[Note: when count = 0 (Apply minimum probability) ↳ Laplace correction

In such case, suppose count = 1

Artificial Neutral Network Classifier :- (ANN)

→ ANN is a set of connected I/O units in which each connecti[on has] a weight associated with it.

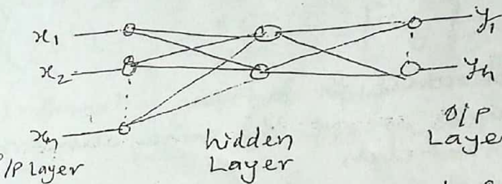
→ During the learning phase, the network learns by adjusting [the] way, so as to be able to predict the correct class label of [the] set.

→ ANN is also referred as connectionist learning due to con[nections] between different units.

→ It has long training time and poor interpretability but h[as] tolerance to noisy data

→ It can classify pattern on which they havenot been tra[ined]. It can classify pattern for continuous value input. It has parallel

→ well suited for continuous value input.



For any classification task, before training, the topology (structure) [must] be designed by specifying number of inputs units, number of h[idden] layer, number of output layer, learning rate and bias value.

Back Propagation Algorithm :-

1. Initilization :
→ set all the weights and threshold levels of the network to random numbers uniformly distributed inside a small rang[e]

2. Activation:
→ Activate the networks by applying the inputs and desired ou[tput] calculate the actual output of the neurons (nodes) in the hi[dden] layer and hence the output layers.

3. Weight Training:
→ update weight in the Network by propagating the errors associat[ed] with the output neurons by calculating error gradients a[nd]

hence to the hidden layers

4. Repeat step 2 and 3 until selected error gradient m...

Measure of Node Impurity:-

- Gini Index
- Entropy Calculation
  - → **ID3 Algorithm** (Iterative Dichotomizer 3)

ID3 is an algorithm used in decision tree to generate a decision tree. ID3 builds the tree from the top down with no back tracking.

→ Information gained is used to select the most useful attribute for classification.

→ Information gained is calculated from entropy calculat...

$$Entropy(H) = -\sum_{i=1}^{m} P(x_i) \cdot \log_2 P(x_i)$$

where, m = no of classes

$P(x_i)$ = Probability of 'x' in class 'i'

Algorithms:

Step 1: Creat a root node for a tree

Step 2: If all examples are of the same class return a single loud with that class

Step 3: If all examples are not of same class, calculate entropy and information gain to select root node and branch node (nodes with highest information gain is selected as root node).

Step 4: Partition the example into subsets.

Step 5: Repeat the process until all examples are classified.

Example:

Person	Hair Length	Weight	Age	class
P_1	0"	250	36	M
P_2	10"	150	34	F
P_3	2"	90	10	M
P_4	6"	78	8	F
P_5	4"	20	1	F
P_6	1"	170	70	M
P_7	8"	160	41	F
P_8	10"	180	38	M
P_9	6"	100	45	M

$\begin{bmatrix} 5- \\ 4- \end{bmatrix}$

Here, entropy of data (H) $= -\sum_{i=1}^{m} P(x_i) \log_2(P(x_i))$

$= -\left[4/9 \log_2(4/9) + 5/9 \log_2(5/9) \right] = 0.9911$

Entropy for hair length;

$(<5'') = 4 <^{1F}_{3M} = -\left[\frac{1}{4}\log_2(1/4) + 3/4 \log_2(3/4)\right] = 0.8113$

$(\geq 5'') = 5 <^{3F}_{2M} = -\left[3/5 \log_2(3/5) + 2/5 \log_2(2/5)\right] = 0.9710$

Information gain = 0.9911 − mean

$= 0.9911 - \left(\frac{4 \times 0.8113 * 5 \times 0.9710}{9} \right)$

$= 0.09107$

Entropy for weight;

$(>160) = -\left[\frac{0}{4}\log_2(0/4) + \frac{4}{4}\log_2(4/4)\right] = 0$

$(\leq 60) = -\left[\frac{4}{5}\log_2(4/5) + 1/5 \log_2(1/5)\right] = 0.7219$

Information gain = 0.9911 − (5/9 * 0.7219) = 0.59

Entropy for age;

$(<= 40) = -\left[3/6 \log_2(3/6) + 3/6 \log_2(3/6)\right] = 1$

$(>40) = -\left[\frac{1}{3}\log_2(1/3) + 2/3 \log_2(2/3)\right] = 0.9183$

Information gain $= 0.9911 - \frac{1 \times 6 + 0.9183 \times 3}{9} = 0.0183$

we choose maximum Information gain

Since, weight has maximum information gain so selected as root node.

Advantages of ID3 :

→ Easy to construct and interprete for small sized data

→ Higher accuracy for simple data.

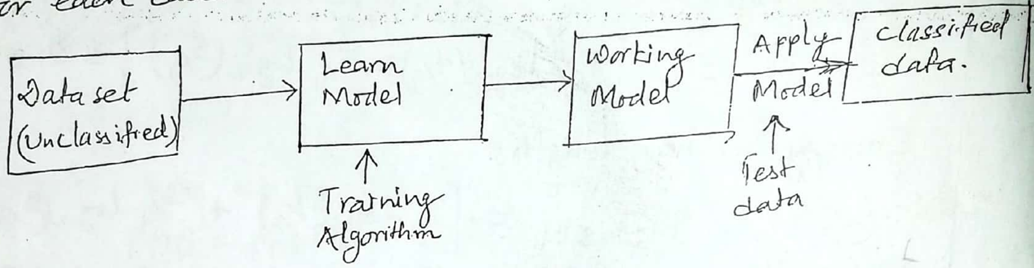
→ fast for classifying unknown records

what is classification?

It is a datamining technique used to predict group-membership of data instances.

→ classification assigns data in a collection of target category or class

→ The goal of classification is to accurately predict the target class for each case in the data.



Model comparission :

✓ (1) Confusion Matrix (contegency Table)

X (2) ROC