Classification

Classification is the process of building a model that predicts the class label for unseen example/ tuple / sample.

It involves two steps

<1> Learning (Training)

<2> Testing (prediction)

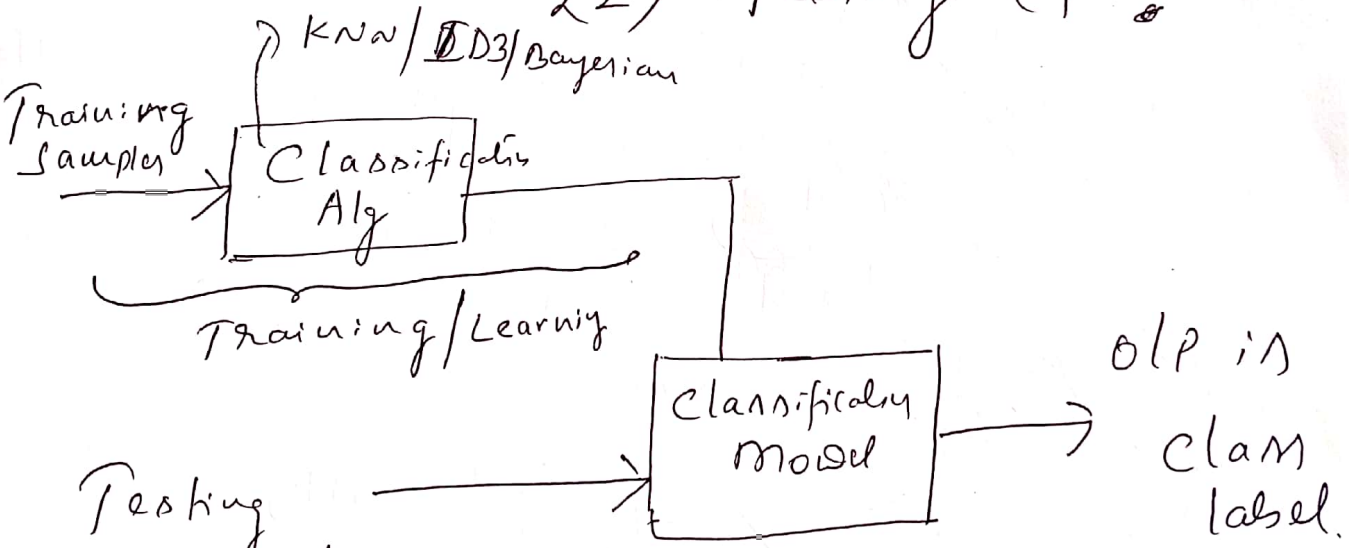


Fig:- A General Classification System.

Classification Methods

(1) Decision Tree Induction

(2) Bayes Algorithm

(3) Rule-based Classification Algorithm

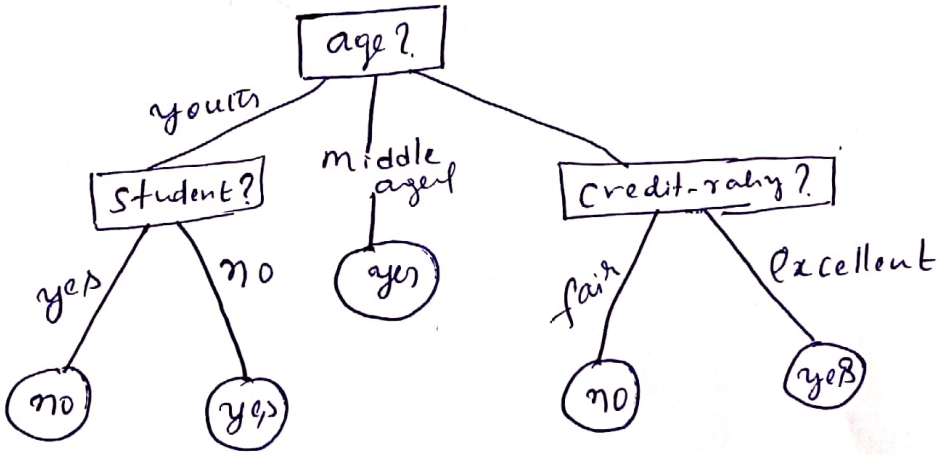
(4) K-Nearest Neighbour Algorithm

(5) Neural Network.

Decision Tree Induction

→ Decision tree induction is the learning of Decision Trees from class-labelled training examples (tuples).

→ A Decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, & each leaf node holds a class label. The topmost node in tree is the root node.



Example :- Decision tree for the concept buys_computer

→ The learning (training) and classificatin (testing) Steps of Decision tree inducting are Simple & fast.

→ It is an example for Nonparameteric classifier.

→ Decision tree classifiers have good accuracy.

→ Decision Tree algorithms have been used for classificating in many applicating areas Such as Medcine, manufacturing & proddctn, financial analysis, & Molecular bidogy.

- In 1980, J Ross Quinlan Developed a Decision tree algorithm known as ID3.

- Later, he presented C4.5 (a Successor of ID3), which become benchmark to compare with newer ~~Supp~~ Supervised algorithms.

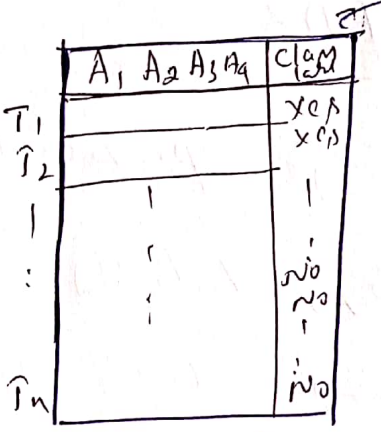
- In 1984, a group of Statisticans published the book Classification and Regression Trees (CART), which described the generation of binary Decision Trees.

- ID3, C4.5 and CART adopt a greedy approach in which Decision trees are Constructed in top-Down recursive manner.

for Example

Consider D - Data partition → D)

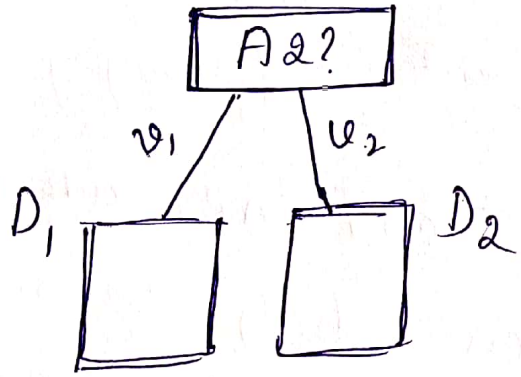
Let
Attribute list
Say $\{A_1, A_2, A_3, A_4\}$



Let
No. of
Classes - 2
& yes, no

Then Select the best splitting attribute such that we Except resultant partition are pure

Let A_2 is selected as best Splitty attribute



Then Algorithm will check first D_1 is pure or impure ?

~~Partition~~ partition

Assume resultant D_1 is impure.

Then Again Split D_1

⇓ SP

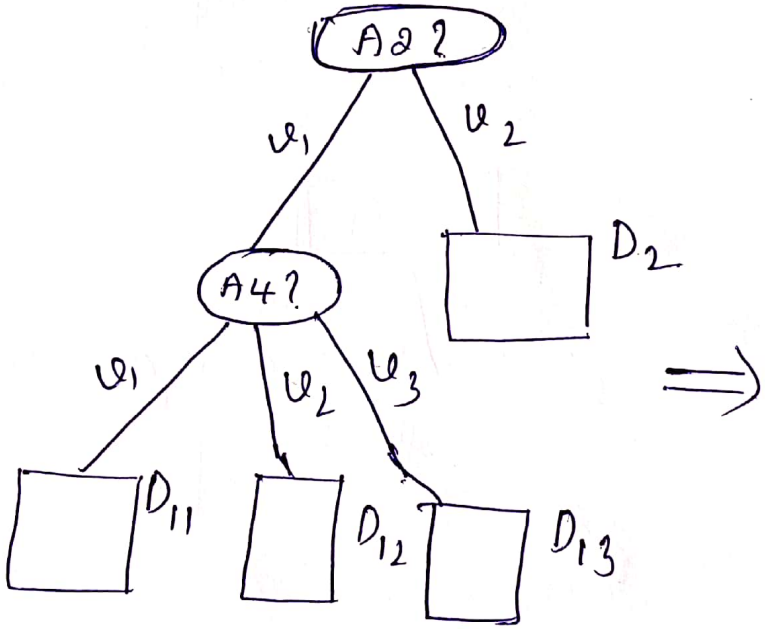
Therefore ~~Select~~ Select best split attribute from $\{A_1, A_2, A_3, A_4\} - A_2$

Select ~~from~~ → $\{A_1, A_3, A_4\}$

best split attribute

say A_4

Then



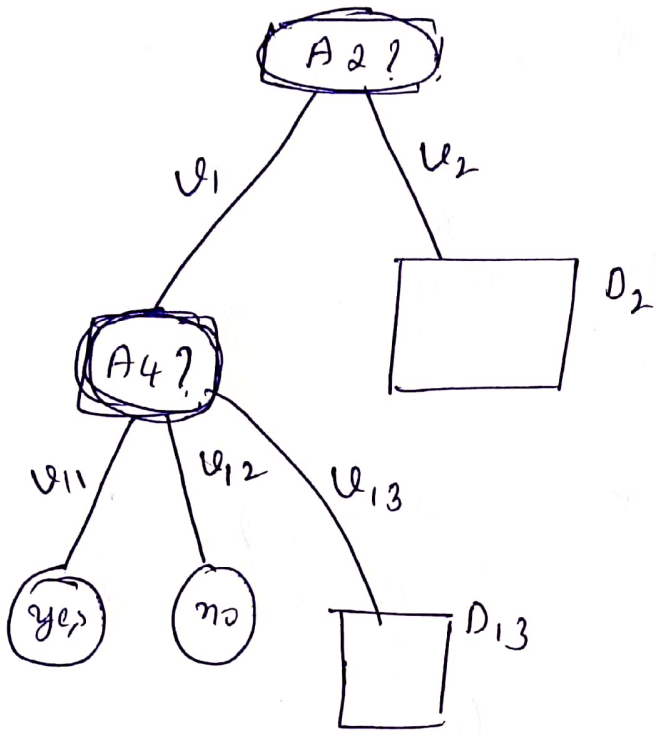
Assume D_{11}, D_{12}, D_{13} in ~~impure~~ impure

are pure

~~Say~~ & D_{11} contain only yes

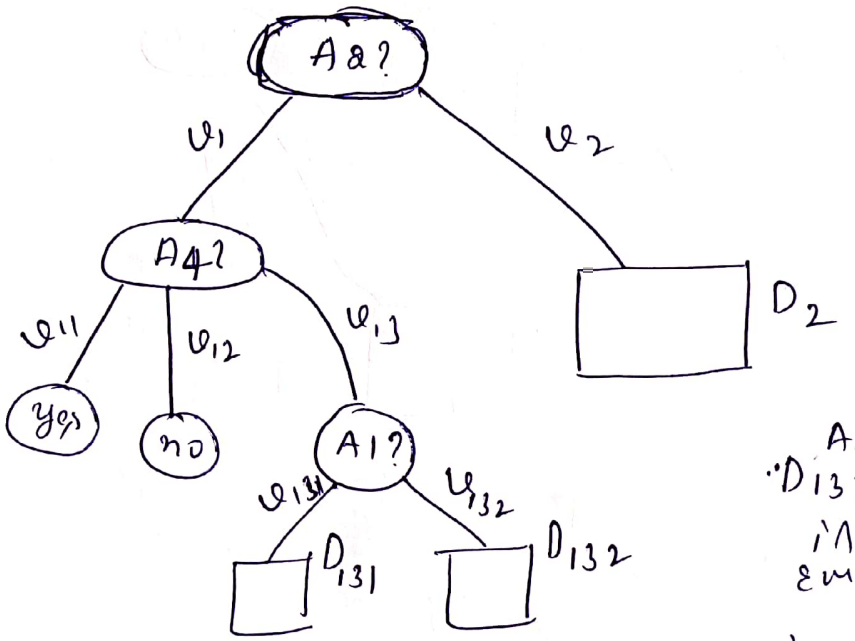
D_{12} → " — no labeled tuples

Then



Then Select the best split attribute among remaining attributes $\{A_1, A_3\}$ to split D_{13}

Say best in A_1



Assume D_{132} is empty

Assume D_{131} is pure (all tuples are no labelled)

Then

Label D_{131} as "no" class label

Since D_{132} is Empty (no tuples)

Then we need to Consider

majority voting in the

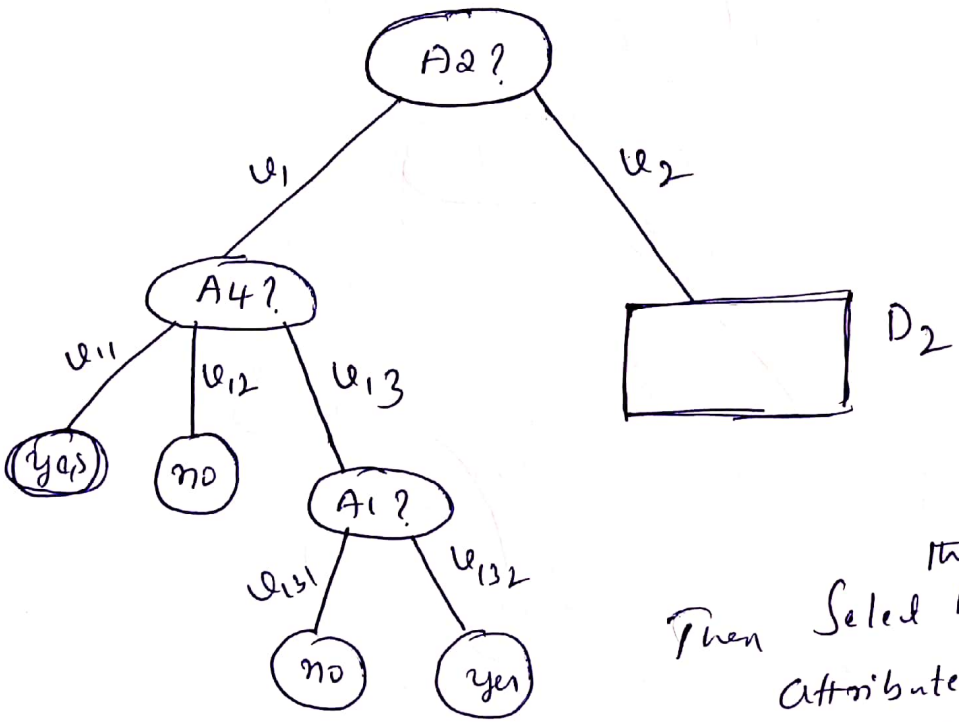
above partition D_{13}

based on That, we have

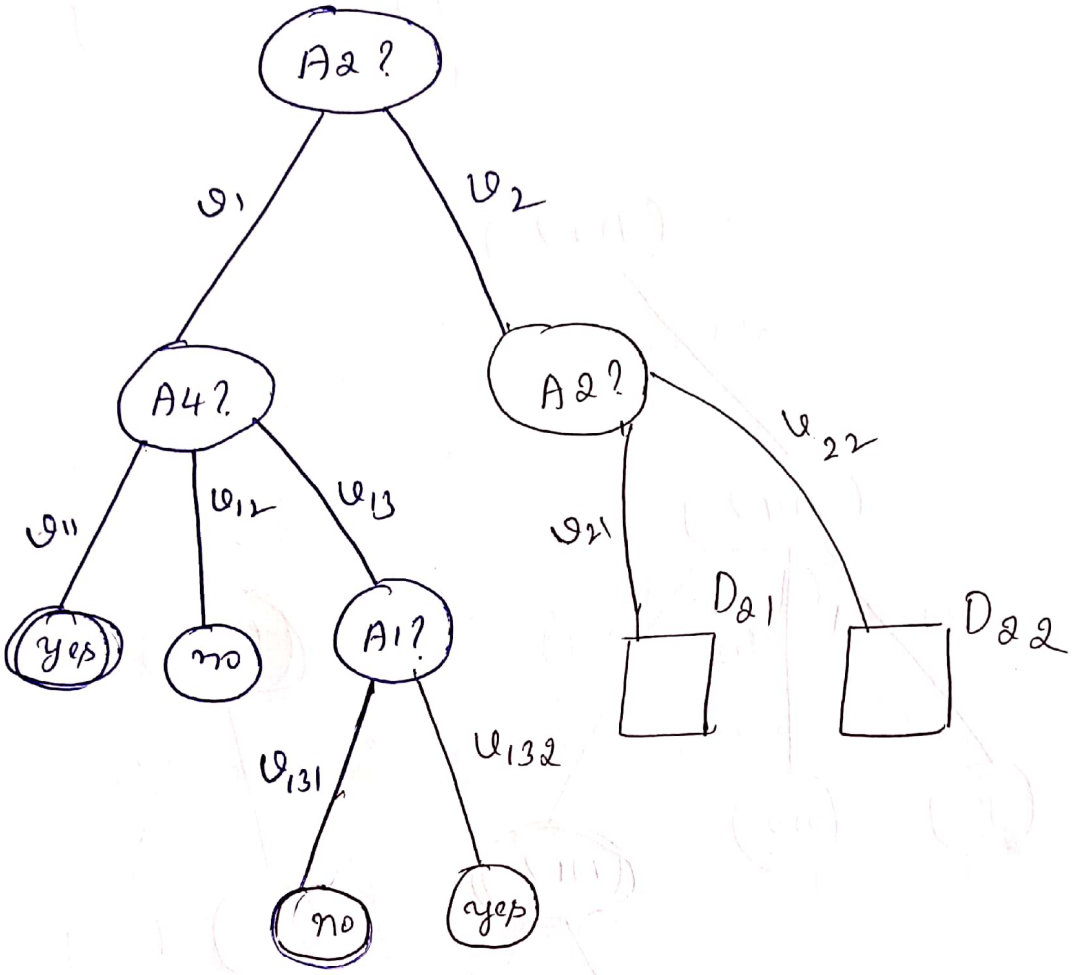
to label ~~D131~~ in D13

Assume majority voting is "yes"

Then Label D_{132} as "yes"



Then Select the best Splitting attribute to ~~Split~~ partition D_2 from list {A1}



Assume if partition D_{21} is pure
(all tuples are
Say "no" labelled)

Then label D_{21} as "no"

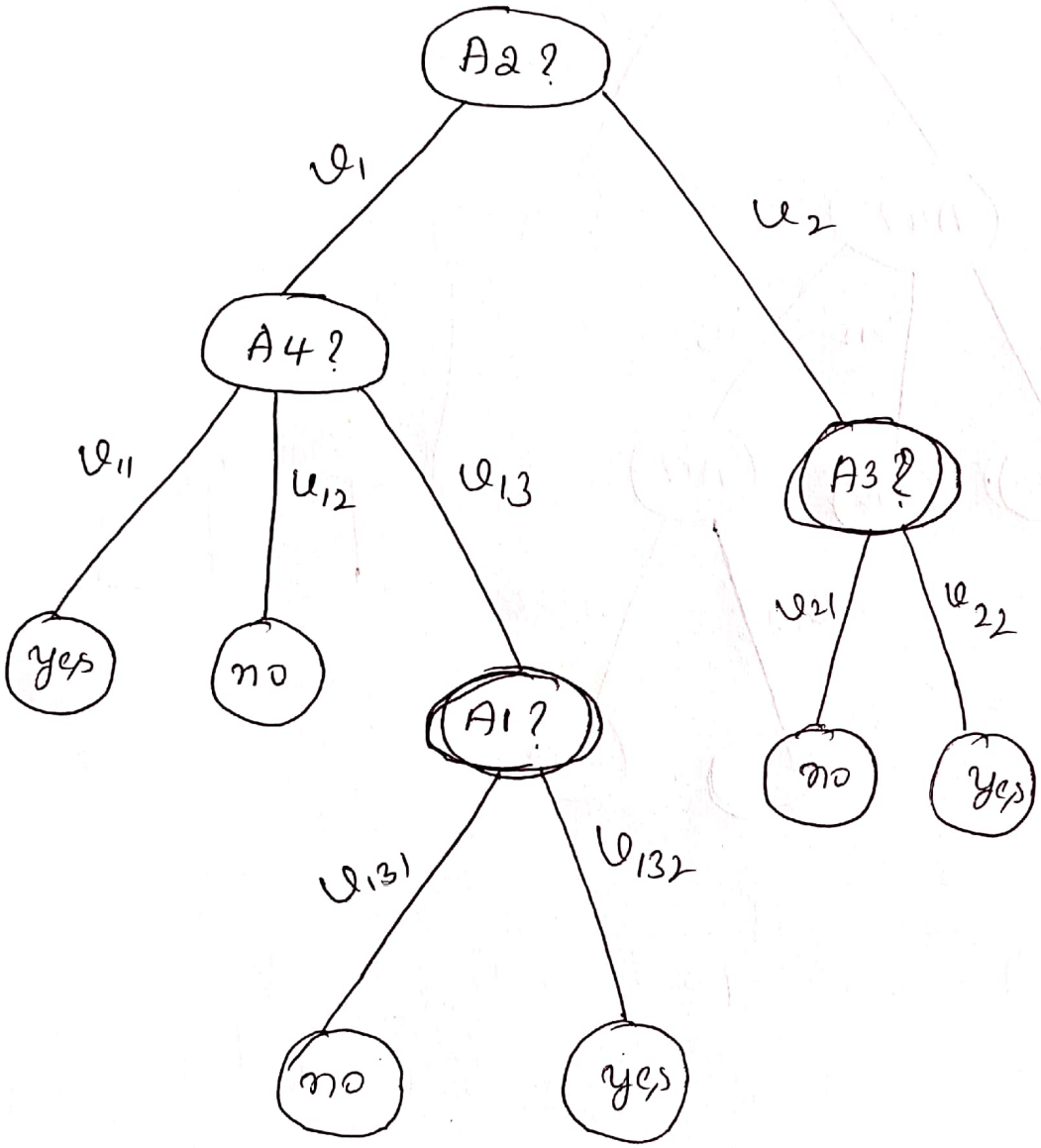
& if D_{22} in impure

Again we have to split D_{22}
but No attributes left

In this situation, we have to apply
majority voting in D_{22}

Assume say majority voting in D_{22} is "yes"

Turn label D_{22} as "yes"



Algorithm: Generate_Decision_Tree

// Generate Decision Tree from training tuples of Data partition (D)

Input:

* Data partition (D)
  // Set of Training tuples & associated class labels.
* attribute_list
  // Set of attributes.
* Attribute_Selection_method.

Output: A Decision Tree

Method

1. Create a Node N
2. if tuples in D are all of the same class C Then
3. return N as a leaf node labelled with class C
4. if attribute_list is empty then
5. return N as a leaf node with majority class in D
6.

6. apply Attribute_Selection_method(D, attribute-list)
to find the "best" Splitting_Criterion;

7. Label node N with Splitting_Criterion;

8. if Splitting_attribute is discrete-valued and
multiways splits allowed then

9. attribute_list ← attribute_list
- Splitting_attribute;

10. for each outcome j of Splitting_Criterion

11. Let Dj be the set of Data tuples in D
satisfying outcome j;

12. if Dj is empty Then

13. attach a leaf labelled with
majority class in D
to node N;

14. else attach the node returned by
Generate_decision_tree(Dj, attribut-list)
to node N;

~~End for~~
End for

15. return N;