Unit 2 :-

Databases are rich with hidden information that can be used as intelligent decision making.
Classification and prediction are two forms of data analysis
- Used to extract models describing important data classes or to predict future data trends.
- Whereas classification predicts categorical (discrete, unordered) labels.
- prediction models continuous valued function.
- For example, we can build a classification model to categorize bank loan applications as either safe or risky.
- Prediction methods have been proposed by researchers to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.
- Classification and prediction have numerous applications including fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.

What is Classification?
- A bank loans officer needs analysis of her data in order to learn which loan applicants are "safe" and which are "risky" for the bank.
- A marketing manager at AllElectronics needs data analysis to help guess whether a customer with a given profile will buy a new computer.

5757511 55

- In each of these examples, the data analysis task is classification.
- Where a model or classifier is constructed to predict categorical labels such as "safe" or "risky" for the loan application data
- "yes" or "no" for the marketing data.
- "treatment A", "treatment B", or "treatment C" for the medical data.

⇒ Suppose that the marketing manager would like to predict how much a given customer will spend during a sale at AllElectronics?
- This data analysis task is an example of numeric prediction, where the model constructed predicts a continuous valued function, or ordered value, as opposed to a categorical label.
- This model is a predictor.
- Regression analysis is a statistical methodology that is most often used for numeric prediction.

How does classification work?
- Data classification is a two-step process.
- In the first step, a classifier is built describing a predetermined set of data classes or concepts.
- This is the learning step (or training phase).
- where, a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels.

classification process. (1) Learning: Training data analyze by classification algorithm. Here the class label attribute is loan_decision, and learned model is represented in the form of classification rules.

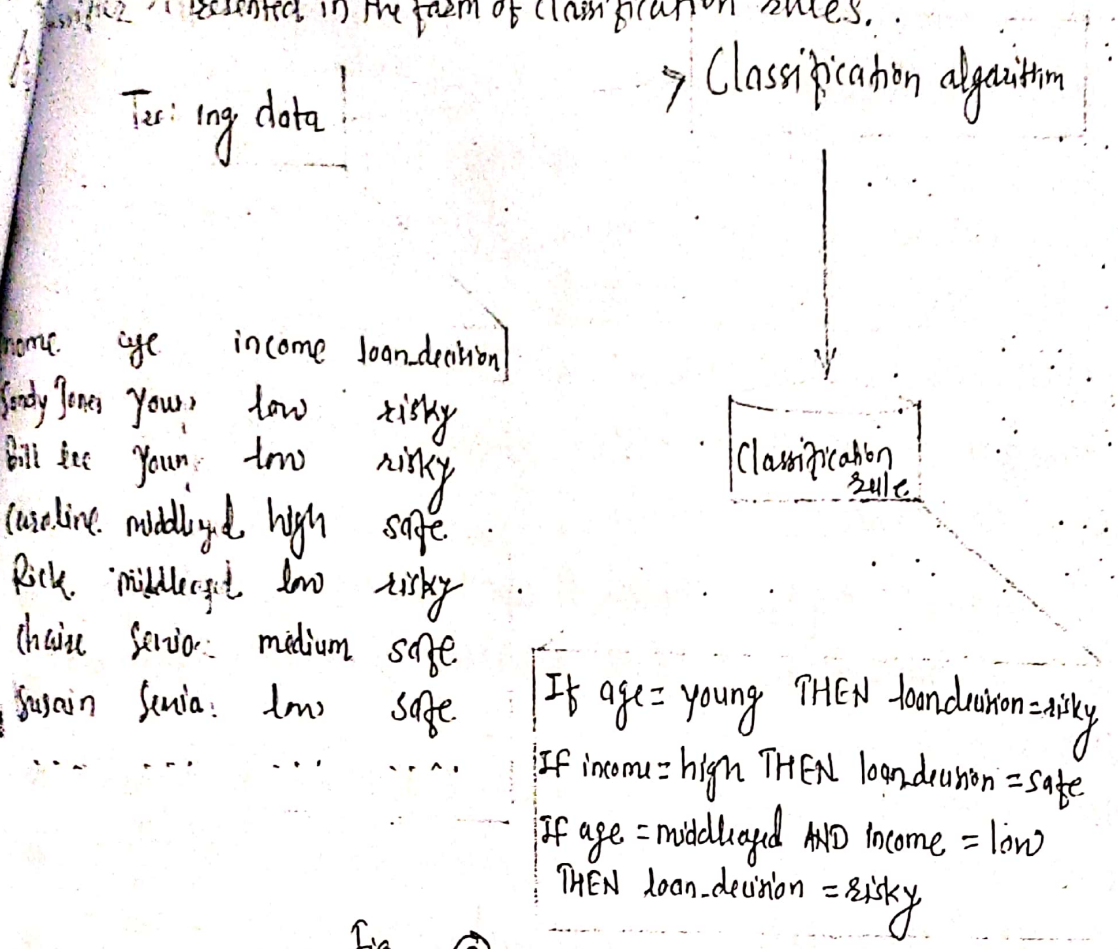


Fig. @

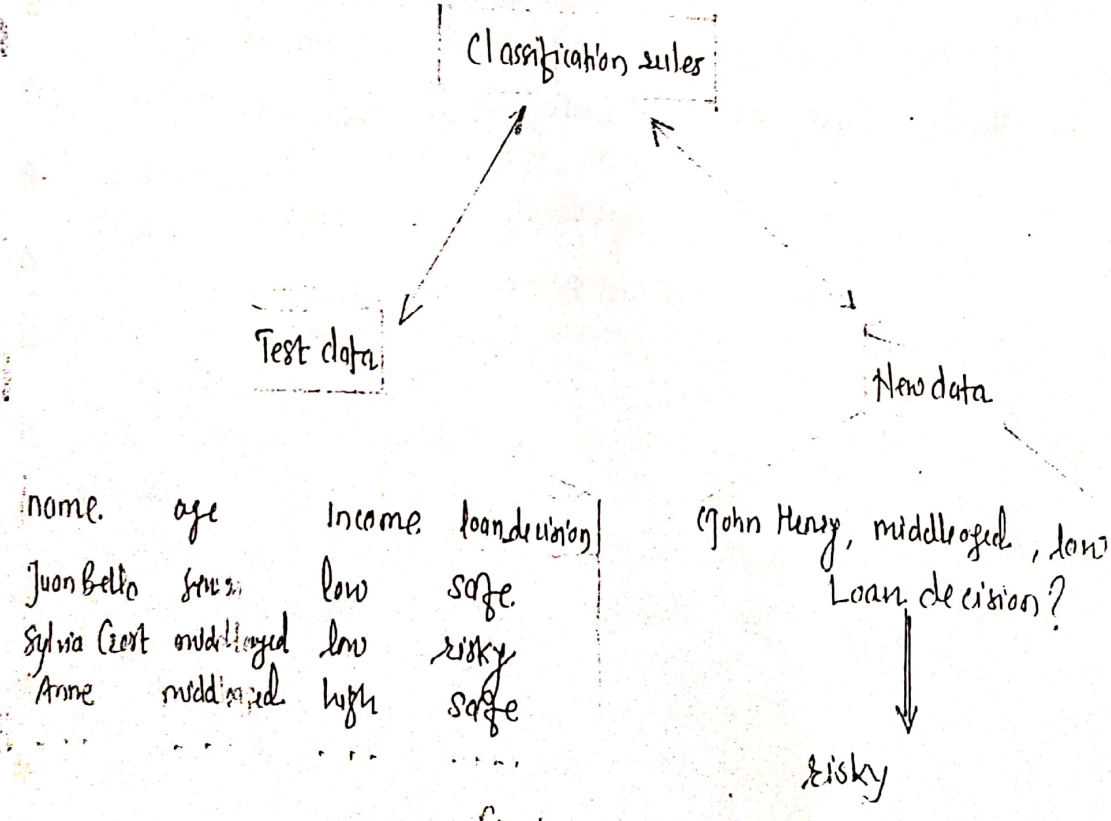


Fig.b

Fig.b Classification: Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

- A tuple X is represented by an n-dimensional attribute vector $X = (x_1, x_2, \ldots x_n)$ depicting n measurements made on the tuple from n-database attributes respectively. $A_1, A_2, \ldots A_n$.
- Each tuple, X is assumed to belong to a predefined class as determined by another database attribute called the <u>class Label attribute</u>.
- The class Label attribute is discrete-valued and unordered.
- It is categorical in that each value serves as a category or class.
- The individual tuples making up the training set are referred as training tuples and selected from the database under analysis.
- In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.
- Because the class label of each training tuple is provided, this step is also known as <u>supervised learning</u> (i.e the learning of the classifier is "supervised" in that it is told to which class each training tuple belongs).

Data for Classification and Prediction

Following preprocessing steps may be applied to the data to help improve the accuracy, efficiency, and scalability of the classification or prediction process.

Data Cleaning: This refers to the preprocessing of data in order to remove or reduce noise (by applying smoothing techniques) and the treatment of missing values (e.g. by replacing a missing value with the most commonly occurring value for that attribute, or with the most probable value based on statistics).

- This step can help reduce confusion during learning.

ii) Relevance analysis: Many of the attributes in the data may be redundant. Correlation analysis can be used to identify whether any two given attributes are statistically related.

- Relevance analysis, in the form of correlation analysis and attribute subset selection, can be used to detect attributes that do not contribute to the classification or prediction task.
- Including such attributes may otherwise slow down, and possibly mislead, the learning step.

iii) Data transformation and reduction:-

- The data may be transformed by normalization.
- When neural networks or methods involving distance measurements are used in learning step.
- Normalization involves scaling all values for a given attribute so that they fall within a small specified range, such as -1.0 to 1.0 or 0.0 to 1.0.

Classification by Decision Tree Induction ⇒

- Decision tree induction is the learning of decision trees from class-labeled training tuples.
- A decision tree is a flow chart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test.
- Each leaf (or terminal node) holds a class label.
- The topmost node in a tree is the root node.

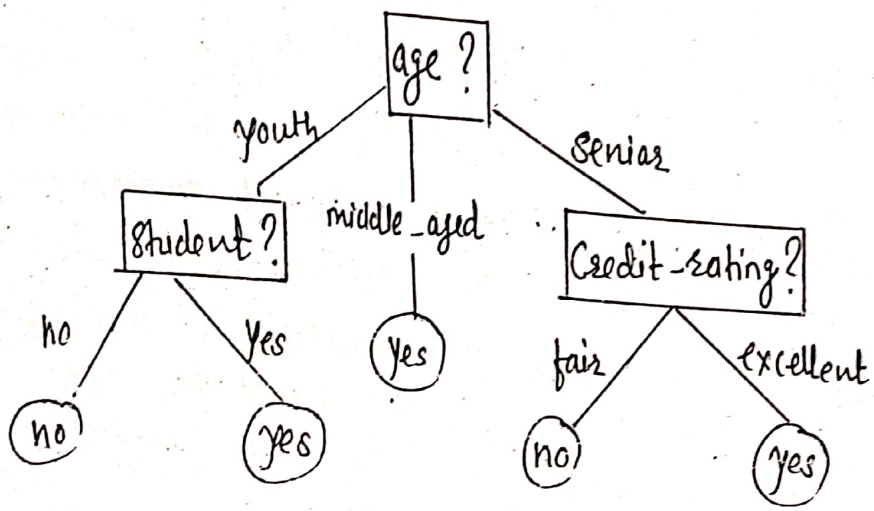


Fig. A decision tree for the concept buys_computer, indicating whether a customer at AllElectronics is likely to purchase a computer.

- Each leaf node represents a class (either buys_computer = yes or buys_computer = no).
- Figure represents the concept buys_computer, that is, it predicts whether a customer at AllElectronics is likely to purchase a computer.
- Internal nodes are denoted by rectangles, leaf nodes are denoted by ovals.
- Some decision trees algorithms produce only binary trees (where each internal node branches have exactly two other nodes).

Decision Tree Induction ⇒

- for learning decision trees
- tree construction, attribute selection measures are used to select the attribute that best partitions the tuples into distinct classes.

Algorithm: Generate_decision_tree ⇒ Generate a decision tree from the training tuples of data partition D.

Input:
- Data Partition, D, which is a set of training tuples and their associated class labels;
- attribute_list, the set of candidate attributes;
- Attribute_selection_method, a procedure to determine the splitting criteria that "best" partition the data tuples into a individual class. Criterion consists of a splitting_attribute

Output: A decision tree.

Method:
1) Create a node N
2) If tuples in D are all of the same class, C then
   return N as a leaf node labeled with the class C;
   if attribute_list is empty then
   return N as a leaf node labeled with the majority class in D;
   apply Attribute_selection_method (D, attribute_list) to find the "best" splitting_criterian;
   label node N with splitting_criterion;
   if splitting_attribute is discrete_valued and
   multiway splits allowed then
   attribute_list ← attribute_list − splitting_attribute //remove splitting_attribute

for each outcome j of splitting_criterion
    // partition the tuples and grow subtrees for each partition
    let Dj be the set of data tuples in D satisfying outcome j;
                                    // a partition
    if Dj is empty then
        attach a leaf labeled with the majority class in D
        to node N;
    else attach the node returned by Generate_decision_tree(Dj,
        attribute_list) to node N;
    end for
    return N.

⇒ fig. Basic algorithm for inducing a decision tree from training tuples.

- The algorithm is called with three parameters: D, attribute_list, Attribute_selection_method
- Initially D refers as a data partition, it is the complete set of training tuples and their associated class labels.
- The parameter attribute_list is a list of attributes describing the tuples.
- Attribute_selection_method specifies a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class.
- The procedure employs an attribute selection measure, such as information gain or the gini index.

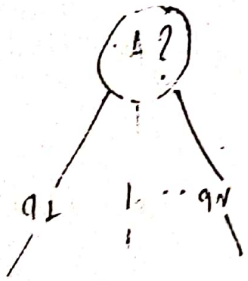
... exists as a single node, N, representing the training ...

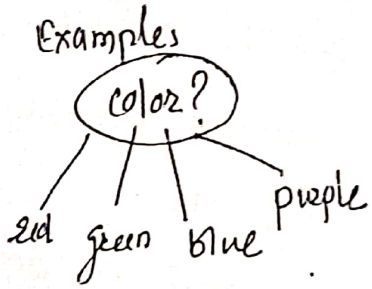
... tuples in D are all of the same class, then node ... becomes leaf and is labeled with that class.

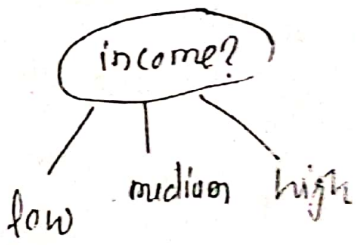
... otherwise, the algorithm calls Attribute_selection_method ... to determine the splitting criterion.

- The splitting criterion tells us which branches to grow from node N with respect to the outcomes of the chosen test.
- The node N is labeled with splitting criterion, which serves as a test at the node.
- A branch is grown from node N for each of the outcomes of the splitting criterion.
- Let A be the splitting attribute. A has v distinct values, $\{a_1, a_2, \ldots a_v\}$, based on the training data.

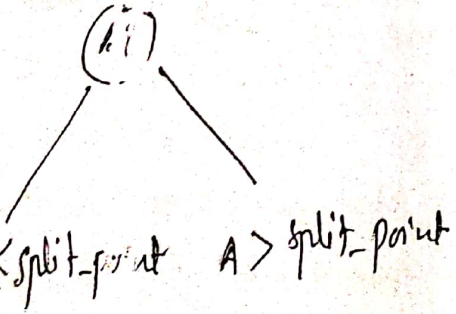
1) A is discrete-valued :-



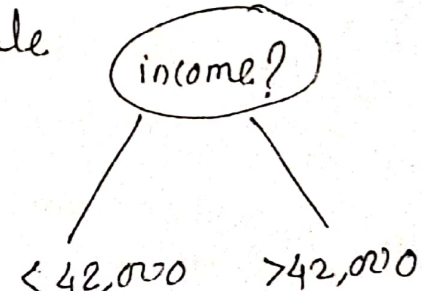




2) A is continuous-valued :-



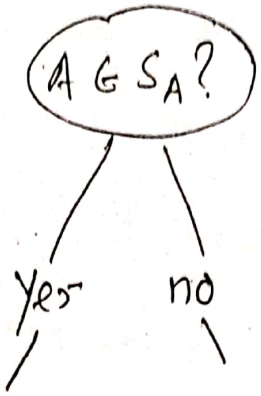
Example



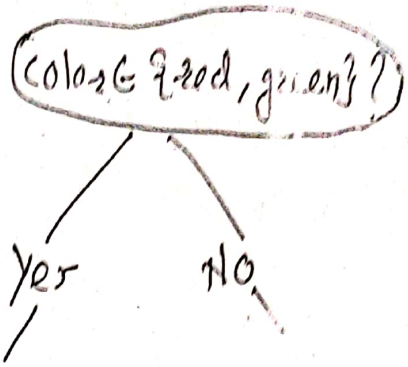
3) A is discrete-valued and binary tree must be produced
- The test at node N is of the form,

$A \in S_A$?

- S_A is the splitting subset for A, returned by Attribute_selection method as part of the splitting criterion.



Example :-



- The algorithm uses the same process recursively to form a decision tree for the tuples at each resulting partition D_j of D.
- The recursive partitioning stops only when any one of the following terminating conditions is true :-

1) All of the tuples in partition D (represented at node N) belongs to the same class, or
2) There are no remaining attributes on which the tuples may be further partitioned.
3) There are no tuples for a given branch, that is, a partition D_j is empty. In this case, a leaf is created with the majority class in D.

Attribute Selection Measures :-

- Attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, D of class-labeled training tuples into individual classes.
- If we were to split D into smaller partitions according to the outcomes of the splitting criterion, ideally each partition would be pure.
- i.e all of the tuples that fall into a given partition would belong to the same class).
- Attribute selection measures are also known as splitting rules because they determine how the tuples at given node are to be split.
- The attribute selection measure provides ranking.
- The attribute having the best score for the measure is chosen as the splitting attribute for the given tuples.
- This selection describes three popular attribute selection measures
  - i) Information gain
  - ii) Gain ratio
  - iii) Gini index

1) Information gain ⇒

- This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages.
- Let node N represent or hold the tuples of partition D.
- The attribute with the highest information gain is chosen as the splitting attribute for node N.

- The expected information needed to classify a tuple in D is given by,

$$Info(D) = -\sum_{i=1}^{m} Pi \log_2(pi),$$

- where Pi is the probability that an arbitrary tuple in D belongs to class Ci.
- and is estimated by $\frac{|Ci,D|}{|D|}$

- A log function to the base 2 is used, because the information is encoded in bits.
- Info(D) is just the average amount of information needed to identify the class label of a tuple in D.
- Info(D) is also known as the entropy of D.
- Now suppose we were to partition the tuples in D on some attribute A having v distinct values {a1, a2, ... av}, as observed from the training data.
- If A is discrete-valued, these values correspond directly to the v outcomes of a test on A.
- Attribute A can be used to split D into v partitions or subsets {D1, D2, ... Dv}.
- Where Dj contains those tuples in D that have outcome aj of A.
- These partitions would correspond to the branches grown from node N.
- How much more information would we still need (after the partitioning) in order to arrive at an exact classification?
- This amount is measured by,

$$Info_A(D) = \sum_{j=1}^{v} \frac{|Dj|}{|D|} \times info(Dj).$$

- The term $\frac{|Dj|}{|D|}$ acts as the weight of the jth partition.

$Info_A(D)$ is the expected information required to classify a tuple from D based on the partitioning by A.

The smaller the expected information (still) required, the greater the purity of the partition.

Information gain is defined as the difference between the original information requirement and the new requirement. That is;

$$Gain(A) = Info(D) - Info_A(D)$$

- The attribute A with the highest information gain, (Gain(A)), is chosen as the splitting attribute at node N.

Example:- Class-labeled training tuples from the ALLElectronics customer database.

RID	age	income	student	credit_rating	class: buys_computer
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	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	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

- In this example, each attribute is discrete-valued.
- The class label attribute, buys_computer, has two distinct values (namely, {yes, no}).
- Therefore, there are two distinct classes that is $m = 2$.
- let class C_1 correspond to yes
  class C_2 correspond to No
- A (root) node N is created for the tuples in D.
- we must compute the information gain of each attribute to find the splitting criterion for these tuples.
- To compute the expected information needed to classify a tuple in D: we use

$$info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

yes = 9 times
no = 5 times
total tuples = 14

here $m=2$, $p_i = \frac{|C_{i,D}|}{|D|}$

$$info(D) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}$$

- Next we need to compute the expected information requirement for each attribute.
- Let's start with the attribute age.

we use,

$$info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times info(D_j)$$

Here $v=3$ i.e

	yes	no
youth (5 times)	2 times yes	3 times no
middle-aged (4 times)	4	0
senior (5 times)	3	2

$$Info_{age}(D) = \frac{5}{14} \times \left(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}\right)$$

$$+ \frac{4}{14} \times \left(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}\right)$$

$$+ \frac{5}{14} \times \left(-\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}\right)$$

$$= 0.694 \text{ bits}$$

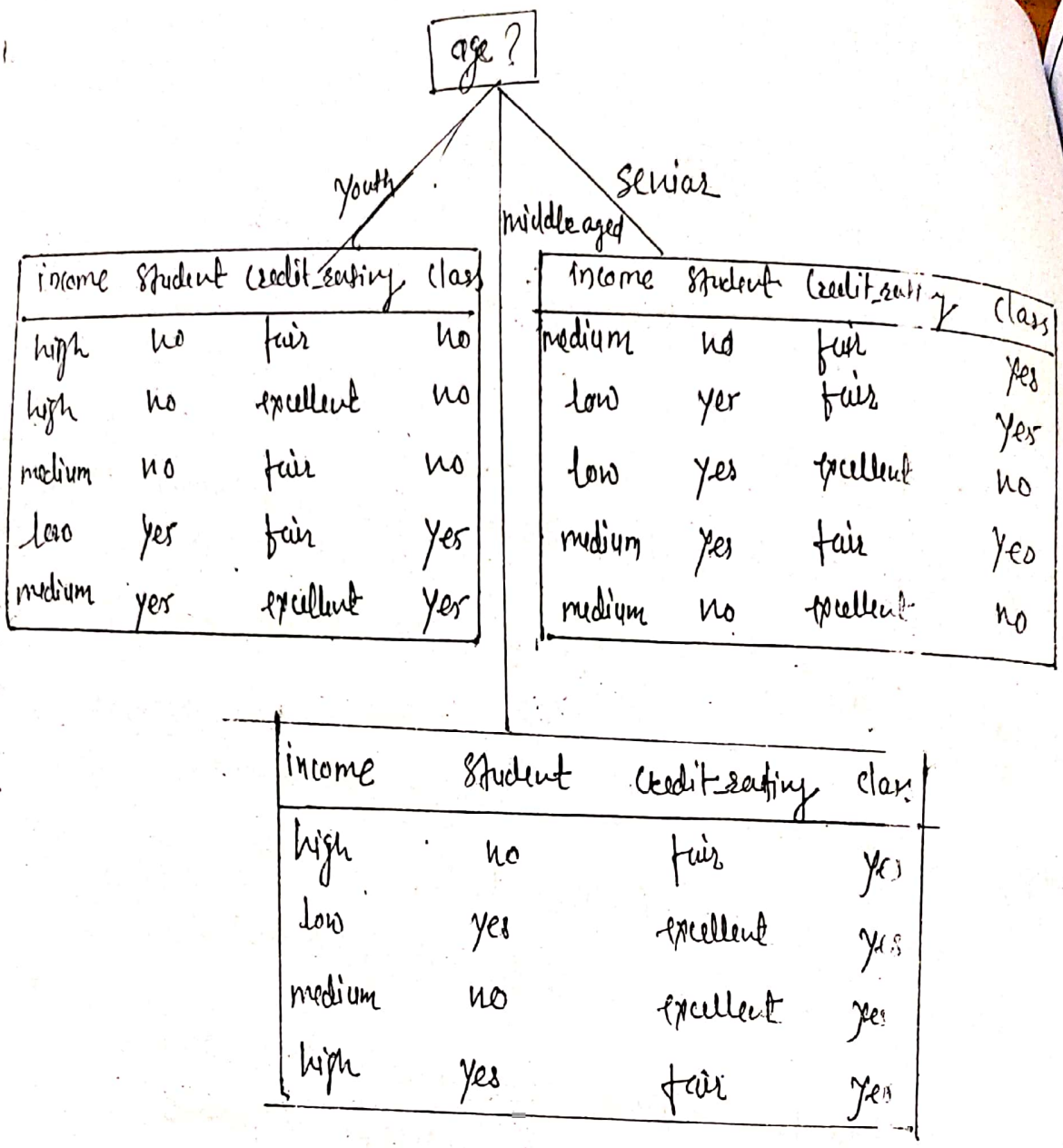
Hence, the gain in information from such a partitioning would be,

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits}$$

Similarly, we can compute Gain(income) = 0.029 bits,
Gain(student) = 0.151 bits and
Gain(credit_rating) = 0.048 bits.

⇒ Because age has the highest information gain among the attributes, it is selected as the splitting attribute.

⇒ Node N is labeled with age, and branches are grown for each of the attributes values.



income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

⇒ Final tree

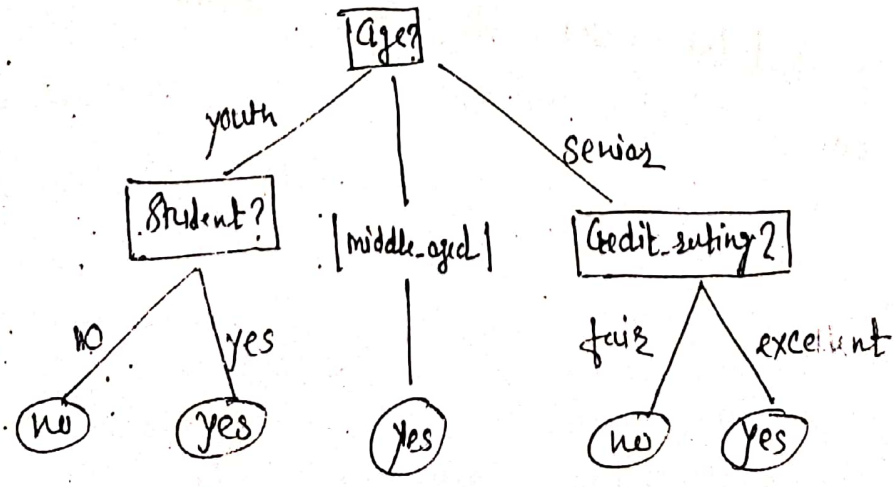


fig: A decision tree for the concept buys_computer

- the process of grouping the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another but are very dissimilar to objects in other clusters.

What is cluster Analysis?

The process of grouping a set of physical or abstract objects into classes of similar objects is called clustering.

- A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters.
- A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression.
- Cluster analysis is an important human activity.
- Cluster analysis has been widely used in numerous applications including market research, pattern recognition, data analysis and image processing.
- In business, clustering can help marketers discover distinct groups in their customer bases and characterize customer groups based on purchasing patterns.
- Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their similarity.
- Clustering can also be used for outlier detection, where outliers (values that are "far away" from any cluster) may be more interesting than common cases.
- Applications of Outlier detection include the detection of credit card fraud and the monitoring of criminal activities in electronic commerce.

Clustering is a form of learning by observation, rather than learning by examples.

Clustering is a challenging field of research in which its potential applications pose their own special requirements. The following are typical requirements of clustering in data mining.

- Scalability :- Highly scalable clustering algorithms are needed.
- Ability to deal with different types of attributes.
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters.
- Ability to deal with noisy data
- Incremental clustering and insensitivity to the order of input records.
- High dimensionality :- A database or a data warehouse can contain several dimensions or attributes. many clustering algorithms are good at handling low-dimensional data, involving two to three dimensions. Finding clusters of data objects in high dimensional space is challenging.
- Constraint-based clustering :- Real-world applications may need to perform clustering under various kinds of constraints.
- Interpretability and usability :- Users expect clustering results to be interpretable, comprehensible, and usable. That is, it is important to study how an application goal may influence the selection of clustering features and methods.