UNIT –V

Classification

Objective:

- To gain knowledge on designing of Classification models to predict categorical class labels; and prediction models predict continuous valued functions.

Syllabus:

Basic Concepts, What is Classification, General approach to classification, decision tree induction, Attribute selection measures: Information gain, Bayes classification methods: Bayes' theorem, Naïve Bayesian classification.

Learning Outcomes:

At the end of the unit, students will be able to:

1. Understand the necessity of Classification and prediction models.
2. Implement classification techniques like decision tree induction, Bayesian classification.

Learning Material

5.1 Basic Concepts

Introduction

Databases are rich with hidden information that can be used for intelligent decision making.Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends.

Classification predicts categorical (discrete, unordered) labels, **Prediction** models continuous valued functions.

Ex: Build a classification model to **categorize bank loan applications** as either safe or risky.

Build prediction model to **predict the expenditures** in dollars of potential customers on computer equipment given their income and occupation.

Classification Techniques :

Basic classification techniques are decision tree classifiers, Bayesian classifiers, Bayesian belief networks, and rule based classifiers, Back propagation etc.,

Methods for prediction:

Linear regression, nonlinear regression etc.,

Applications :

- Detecting spam email messages.
- Target marketing
- Medical diagnosis
- Credit approval

Supervised learning:

In which the class label of each training tuple is known, and the number or set of classes to be learned known in advance.

Un Supervised learning:

In which the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance.

Classification is supervised learning (i.e., the learning of the classifier is "supervised" in that it is told to which class each training tuple belongs)

Training data :

Consisting of records or tuples whose class labels are known must be provided. The training set is used to build a classification model

Test data:

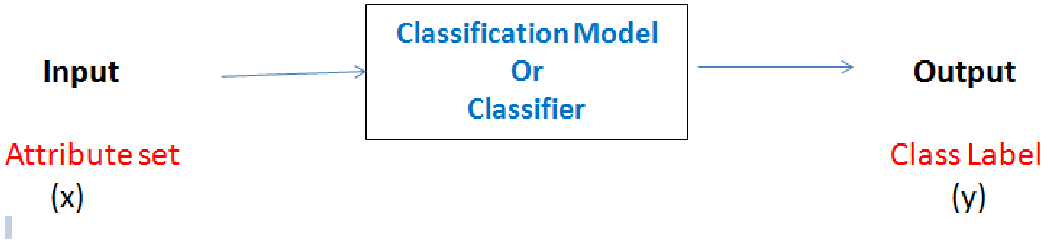
Consisting of records with unknown class labels.

Class label attribute:

The class label attribute is discrete-valued and unordered. It is *categorical in that each* value serves as a category or class.

5.1.1. What Is Classification?

Classification is one form of data analysis, 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. **i.e.** classifying future or unknown objects



Model representation:

classification rules, decision trees, or mathematical formulae

5.1.2. General Approach to Classification

Data classification is a two-step process.

First step:(Learning)

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**.

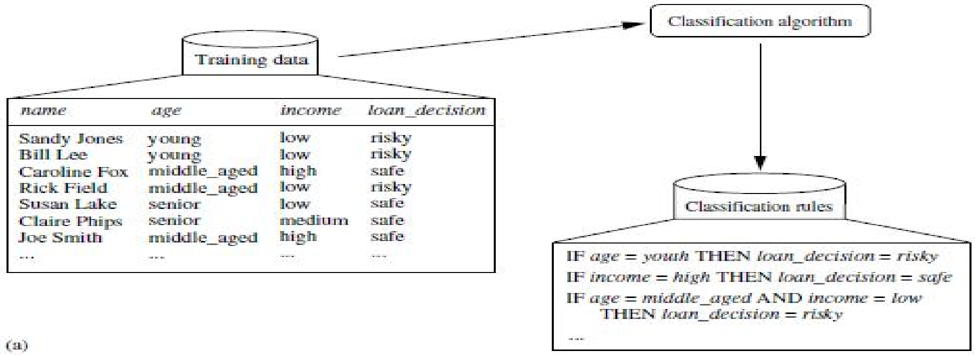
Second step :(Classification)

The model is used for classification. Use the **test set** to measure the accuracy of the classifier. If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known.

- For finding the accuracy of the model test data is used, It is made up of test tuples and their associated class labels. They are independent of the training tuples.
- The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.
- The associated class label of each test tuple is compared with the learned classifier’s class prediction for that tuple.

❖ If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known.

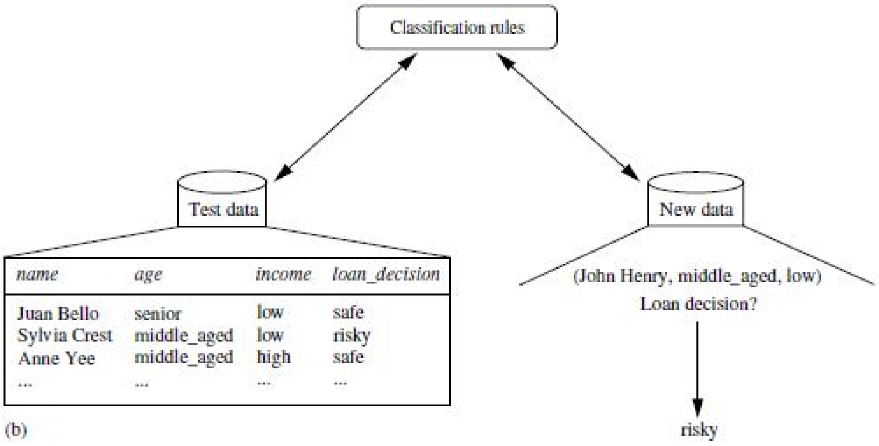
Ex :A bank loans officer needs analysis of her data in order to learn which loan applicants are “safe”and whichare “risky” for the bank.



name	age	income	loan_decision
Sandy Jones	young	low	risky
Bill Lee	young	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

IF *age* = *youth* THEN *loan_decision* = *risky*
IF *income* = *high* THEN *loan_decision* = *safe*
IF *age* = *middle_aged* AND *income* = *low*
THEN *loan_decision* = *risky*
...

(a) **Learning:** Training data are analyzed by a classification algorithm. Here, the class label attribute is loan decision, and the learned model or classifier is represented in the form of classification rules



name	age	income	loan_decision
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

(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.

.

5.2. Decision Tree Induction

- A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a **test** on an attribute, each branch represents an **outcome** of the test, and leaf nodes (or terminal node) represent **classes** or class distribution holds a class label. The topmost node in a tree is the root node.

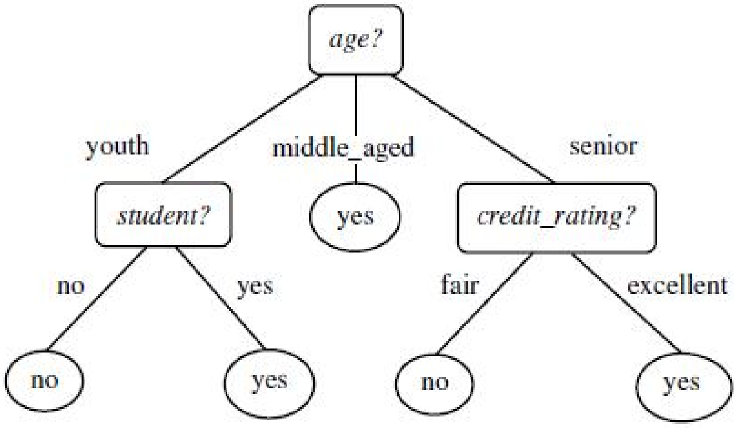


Fig : A decision tree for the concept *buys computer, indicating whether a customer at AllElectronics* is likely to purchase a computer. Each internal (nonleaf) node represents a test on an attribute. Each leaf node represents a class (either *buys computer = yes or buys computer = no).*

- It predicts whether a customer at AllElectronics is likely to purchase a computer.

- Internal nodes are denoted by **rectangles**, and leaf nodes are denoted by **ovals**.

- Some decision tree algorithms produce only binary trees (where each internal node branches to exactly two other nodes), whereas others can produce non binary trees.

Decision trees used for classification to classify an unknown sample i.e whose class label is unknown.

- Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple.
- Decision trees can easily be converted to classification rules.
- The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.
- Decision trees can handle high dimensional data.
- Decision tree induction algorithms have been used for classification in many application areas, such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.

Algorithm: **Generate_ decision_tree**. Generate a decision tree from the given training data.

Input: The training samples, samples, represented by discrete-valued attributes; the set of candidate attributes , attribute_list.

Output: A decision tree.

Method:

```
(1) create a node N;
(2) If samples are all of the same class, C then
(3)       return N as a leaf node labeled with the class C;
(4) if attribute _list is empty then
```

```
(5) return N as a leaf node labeled with the most common class in samples.
(6) Select test-attribute, the attribute among attribute_list with the highest information gain.
(7) label node N with test-attribute;
(8) for each known value $a_i$ of test-attribute
(9  grow a branch from node N for the condition test-attribute = $a_i$;
 (10 let $s_i$ be the set of samples in samples for which test-attribute = $a_i$;
(11) If $s_i$ is empty then
(12) Attach a leaf labeled with the most common class in samples;
(13) else  attach the node returned by Generate_decision_tree($s_i$, attribute-list-test-attribute);
```

5.2.1 Decision tree induction

- Decision tree induction is the learning of decision trees from class-labeled training tuples.
- The basic algorithm for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner.
- A well-known decision tree induction algorithm is ID3(Iterative Dichotomiser).

The basic strategy for decision tree induction is

- The tree starts as a single node, N, representing the training samples(**step 1**)

- If the samples all of the same class, then the node becomes a leaf and is labeled with that class (**steps 2 and 3**).

- Otherwise, the algorithm uses an entropy-based measure known as information gain a heuristic for selecting the attribute that will best separate the samples in to individual classes (**step 6**). This attribute becomes the "test" or "decision" attribute at the node (**step 7**). All attributes are categorical, that is discrete-valued. Continuous-valued attributes must be discretized.

- A branch is created for each known value of the test attribute, and the samples are partitioned accordingly (**steps 8-10**)

- The algorithm uses the same process recursively to form a decision tree for the samples at each partition. Once an attribute has occurred at a node, it need not be considered in any of the node's descendents(**step 13**)

- The recursive partitioning stops only when any one of the following conditions is true:

  a) All samples for a given node belong to the same class (**steps 2 and 3**), or

  b) There are no remaining attributes on which the samples may be further partitioned (**step 4**). In this case, majority voting is employed (**step 5**). This involves converting the given node into a leaf and labeling it with the class in majority among samples. Alternatively, the class distribution of the node samples may be stored.

c) There are no sample for the branch test-attribute = a_i (**step 11**). In this case, a leaf is created with the majority class in samples (**step 12**).

5.2.2. Attribute Selection Measure : Information Gain

- The information gain measure is used to select the test attribute at each node in the tree. Such a measure is referred to as an attribute selection measure or a measure of the goodness of split.
- The attribute with the highest information gain (or greatest entropy reduction) is chosen as the test attribute for the current node.
- Let **S** be a set consisting of **s** data samples.
- Suppose the class label attribute has 'm' distinct values defining **m** distinct classes, **C_i(for i=1,...m).**
- Let **s_i** be the number of samples of **S** in class **C_i**.
- The expected information needed to classify a given sample is given by

$$\mathbf{I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} P_i \log_2 (P_i)}$$

- **P_i** is the probability that an arbitrary sample belongs to class **Ci** and is estimated by **s_i/s**

Entropy and Information Gain

S contains s_i tuples of class C_i for i = {1, …, m}

Information measures info required to classify any arbitrary tuple

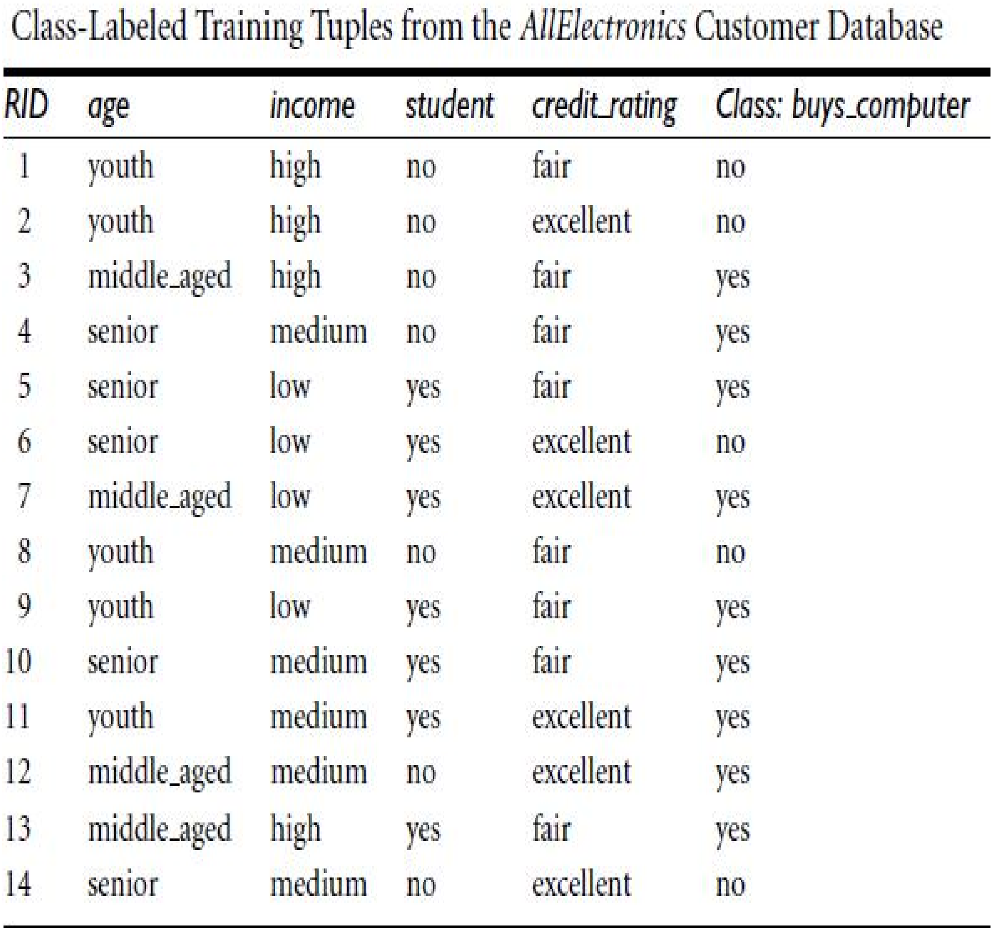
$$I(s_1,s_2,\ldots,s_m) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

Entropy (weighted average) of attribute A with values $\{a_1,a_2,\ldots,a_v\}$

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \ldots + s_{mj}}{s} I(s_{1j},\ldots,s_{mj})$$

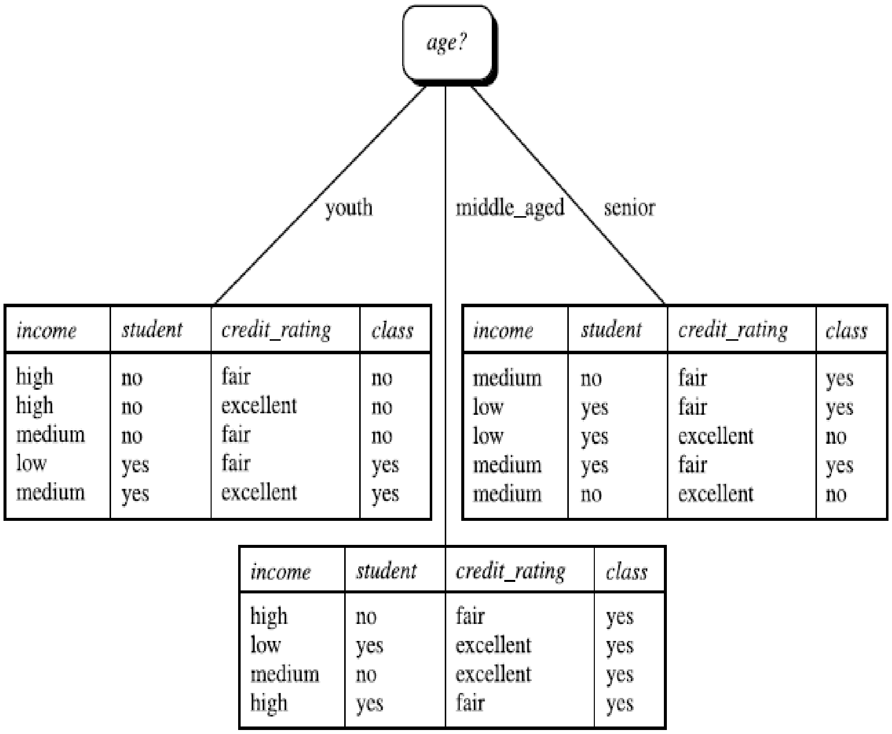
Information gained by branching on attribute A

$$\text{Gain}(A) = I(s_1, s_2,\ldots,s_m) - E(A)$$



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



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

The attribute *age* has the highest information gain and therefore becomes the splitting attribute at the root node of the decision tree. Branches are grown for each outcome of *age*. The tuples are shown partitioned accordingly.

5.3. Bayes Classification methods

- Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class.

- Bayesian classification is based on Bayes' theorem

- Simple Bayesian classifier known as the naïve Bayesian classifier to be comparable in performance with decision tree and neural network classifiers

- Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.

- Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called **class conditional independence**

5.3.1. Bayes' Theorem

Let X be a data sample whose class label is unknown. Let H be some hypothesis, such as that the data sample X belongs to a specified class C.

For classification problems, determine P (H/X), the probability that the hypothesis H holds given the observed data sample X

P (H/X) is the **posterior probability,** of H conditioned on X.

Ex : Suppose the world of data samples consists of fruits, described by their color and shape.

Suppose that X is red and round, and that H is the hypothesis that X is an apple.

Then P (H/X) reflects our confidence that X is an apple given that we have seen that X is red and round.

P (H) is the **prior probability**, of H.

Ex: This is the probability that any given data sample is an apple, regardless of how the data sample looks.

- The posterior probability, P (H/X) is based on more information (such as background knowledge) than the prior probability, P (H), which is independent of X
- P (X/H) is the posterior probability of X conditioned on H.

i.e It is the probability that X is red and round given that we know that it is true that X is an apple.

- P (X) is the prior probability of X. it is the probability that a data sample from our set of fruits is red and round.
- P (X), P (H), and P (XjH) may be estimated from the given data. Bayes theorem is useful in that it provides a way of calculating the posterior probability, P (H/X) from P (H), P (X), and P(X/H).

- **Bayes theorem**

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}.$$

5.3.2. Naïve Bayesian Classification

The naïve Bayesian classifier, or simple Bayesian classifier, works as follows:

1. Let D be a training set of tuples and their associated class labels. each tuple is represented by an n-dimensional attribute vector, X = (x1, x2, : : : , xn), depicting n measurements made on the tuple from n attributes, respectively, A1, A2, : : : , An.
2. Suppose that there are m classes, C1, C2, : : : , Cm. Given a tuple, X, the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X. That is, the naïve Bayesian classifier predicts that tuple X belongs to the class Ci if and only if

   P(Ci/X) > P(Cj/X) for 1≤ j ≤ m; j≠ i

   Thus we maximize P(CijX). The classCi for which P(CijX) is maximized is called the maximum posteriori hypothesis. By Bayes' theorem

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}.$$

3. As *P(X) is constant for all classes, only P(XjCi)P(Ci) need be maximized.*

4. Given data sets with many attributes, it would be extremely computationally expensive to compute *P(XjCi). In order to reduce computation in evaluating P(XjCi), the* naive assumption of **class conditional independence** i.e., that there are no dependence relationships among the attributes).

$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^{n} P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i). \end{aligned}$$

We can easily estimate the probabilities P(x1/Ci), P(x2/Ci), : : : , P(xn/Ci) fromthe training tuples. Here xk refers to the value of attribute Ak for tuple X.

To compute *P(X/Ci), we consider the following:*

a) If *Ak is categorical, then P(xk/Ci) is the number of tuples of class Ci in D having* the value *xk for Ak, divided by, the number of tuples of class Ci in D.*

b) If Ak is continuous-valued, then the attribute is typically assumed to have a Gaussian distribution

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}).$$

5. To classify an unknown sample X, P(X/Ci)P(Ci) is evaluated for each class Ci.

The classifier predicts that the class label of tuple X is the class Ci if and only if

P(X/Ci)P(Ci) > P(X/Cj)P(Cj) for 1≤ j ≤ m; j≠ I

In other words, it is assigned to the class Ci for which P(X/Ci)P(Ci) is the maximum;

UNIT-V
Assignment-Cum-Tutorial Questions
SECTION-A

Objective Questions

1. Data Classification process involves_____________, __________________.

2. Classification is a supervised learning. [T/F]

3. _______________ measure is used to select the test attribute at each node in the decision tree. [ ]

   A) Information Gain. B) Attribute Selection.

   C) Measure of the goodness of split. D) All of the above

4. Posterior probability can be calculated by ___________ theorem. [ ]

   A) Bayes. B) Apriori. C) Entropy. D) All

5. The neural network learns By adjusting the _____________ . [ ]

   A) Heights. B) Weights. C) Depths. D) All

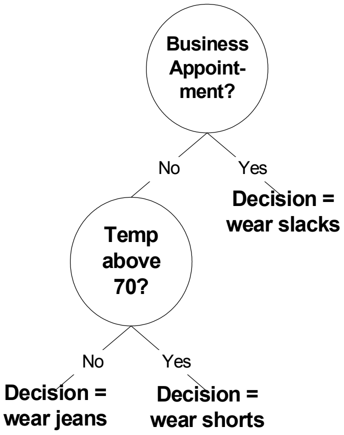
6. The process of forming general concept definitions from examples of concepts to be learned. [ ]

   A) Deduction. B) Disjunction. C) Induction. D) Conjunction.

7. Data used to build a data mining model. [ ]

   A) Validation Data. B) Hidden Data. C) Test Data. C) Training Data.

8. Which of the following is a valid production rule for the decision tree below?



A) IF Business Appointment = No & Temp above 70 = No

THEN Decision = wear slacks

B) IF Business Appointment = Yes & Temp above 70 = Yes

THEN Decision = wear shorts

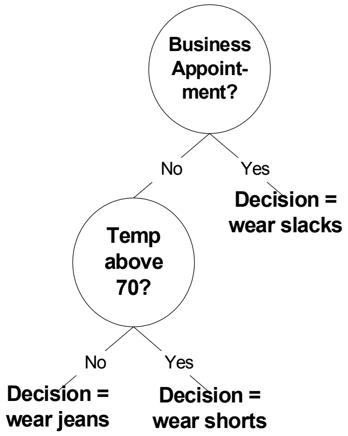
C) IF Temp above 70 = No

THEN Decision = wear shorts

D) IF Business Appointment= No & Temp above 70 = No

THEN Decision = wear jeans [ ]

9.Which of the following is a valid production rule for the decision tree below?



A) IF Business Appointment = No & Temp above 70 = yes

THEN Decision = wear shorts.

B) IF Business Appointment = Yes & Temp above 70 = Yes

THEN Decision = wear shorts

C) IF Temp above 70 = No

THEN Decision = wear shorts

D) IF Business Appointment= No & Temp above 70 = No

THEN Decision = wear slack. [ ]

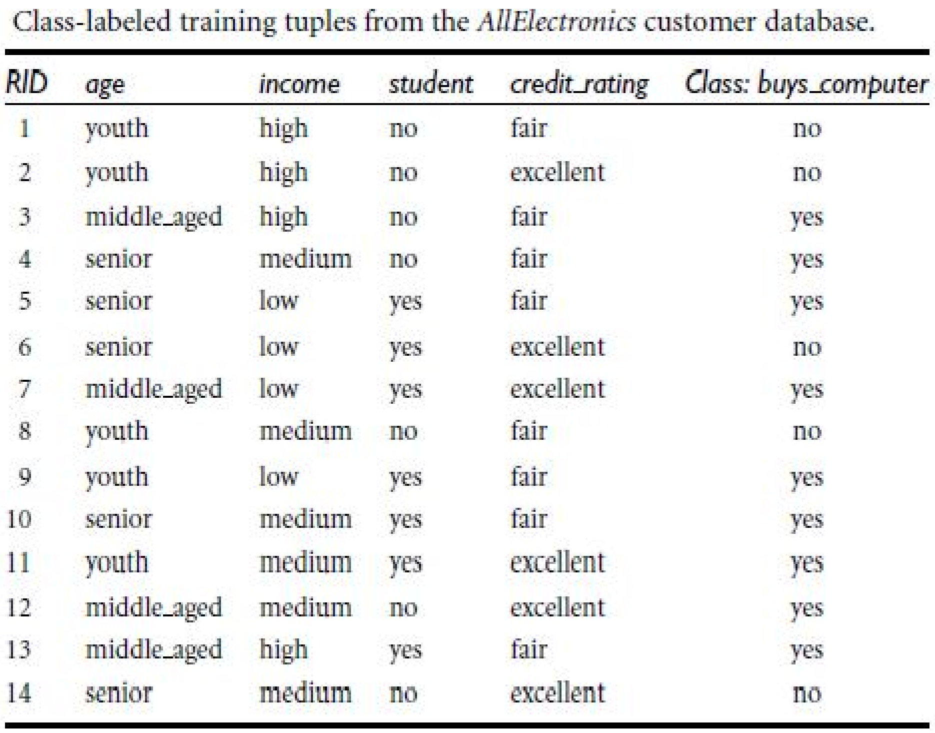
10.Decision tree is a type of _______________ algorithm. [ ]

A) Brute force approach. B) Randamized . C) Greedy D) None

SECTION-B

SUBJECTIVE QUESTIONS

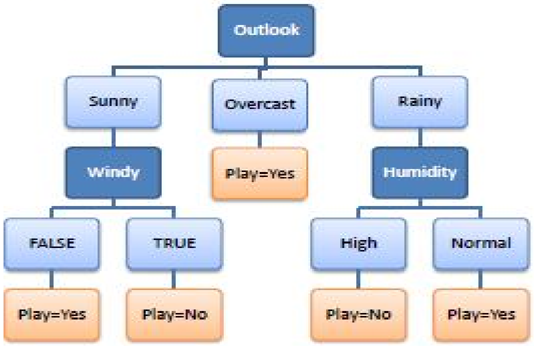
1. With a neat diagram explain Data Classification Process.
2. Elaborate the issues regarding Classification and Prediction.
3. Illustrate the process of classification by Decision Tree Induction.
4. Build a decision tree for the concept buys_computer using the below database.



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

5. What is the need for Tree Pruning?
6. Describe how classification rules are extracted from the decision tree with the following example.



7. Briefly explain about Bayesian classification.