Al-Mustaqbal University
College of Sciences
Intelligent Medical System Department

جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

# كلية العلوم
# قسم الانظمة الطبية الذكية

# Lecture: (6)

**Classification: Decision Trees & Bayesian Classification Subject:**
**Clinical Data Mining**
**Level: Four**
**Lecturer: Dr. Maytham Nabeel Meqdad**

## Classification: Decision Trees & Bayesian Classification

**Classification** is a supervised learning technique used to build models that can predict the class label of new data based on previously labeled training data. Two of the most widely used classification methods are **Decision Trees** and **Bayesian Classification**.

---

# 1. Decision Trees

A **Decision Tree** is a tree-like predictive model used to classify data by recursively splitting it based on the most informative attributes.

### How Decision Trees Work

- Each **internal node** represents a test on an attribute.
- Each **branch** represents the outcome of that test.
- Each **leaf node** represents a class label.
- The goal is to partition the dataset into smaller, purer subsets.

### Attribute Selection Measures

To choose the best attribute for splitting, different metrics are used:

- **Entropy** – measures disorder or impurity in the dataset.
- **Information Gain** – the reduction in entropy after a split.
- **Gain Ratio** – adjusts information gain to reduce bias.
- **Gini Index** – used in CART to measure node purity.

### Common Algorithms

- **ID3** – uses Information Gain.
- **C4.5** – uses Gain Ratio and handles continuous attributes.
- **CART** – uses the Gini Index and produces binary trees.

### Advantages

- Easy to understand and interpret.
- Can handle both numerical and categorical data.
- Requires little data preparation.

### Disadvantages

- Prone to overfitting.
- Sensitive to noisy data.
- Choosing the optimal tree size can be difficult

# . Bayesian Classification

Bayesian Classification is based on **Bayes' Theorem**, which calculates the probability of a class given a set of features.

### Bayes' Theorem

P(C|X)=P(X|C)·P(C)P(X)P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}P(C|X)=P(X)P(X|C)·P(C)

Where:

- **P(C|X)**: Posterior probability of class *C* given data *X*
- **P(X|C)**: Likelihood of data *X* within class *C*
- **P(C)**: Prior probability of class *C*
- **P(X)**: Probability of the data

### Naïve Bayes Classifier

The most common Bayesian method.
It assumes **feature independence**, which simplifies probability calculations but still produces strong performance in many applications, especially text classification.

### Steps in Naïve Bayes

1. Calculate prior probabilities for each class.
2. Compute likelihoods for each feature value given each class.
3. Apply Bayes' theorem to obtain posterior probabilities.
4. Assign the class with the highest posterior probability.

### Advantages

- Very fast and efficient.
- Performs well with large datasets.
- Excellent performance in text classification and spam filtering.

### Disadvantages

- Assumes independence between features, which is often unrealistic.
- Sensitive to zero-frequency problems without smoothing.

---

# 3. Comparison

| Feature | Decision Trees | Bayesian Classifier |
|---|---|---|
| Speed | Fast | Very fast |
| Interpretability | Highly interpretable | Less interpretable |
| Handling noise | More sensitive | More robust |
| Text classification | Moderate | Excellent |
| Risk of overfitting | High | Low |

- Decision Trees and Bayesian Classification are two fundamental techniques in data mining. Decision Trees are ideal when interpretability is important, while Bayesian methods—especially Naïve Bayes—are highly efficient and perform exceptionally well with large or text-based datasets. Both methods provide powerful tools for building accurate and reliable classification models

**EBook: Data Mining: Concepts and Techniques, Second Edition"Jiawei Han and Micheline Kamber"**

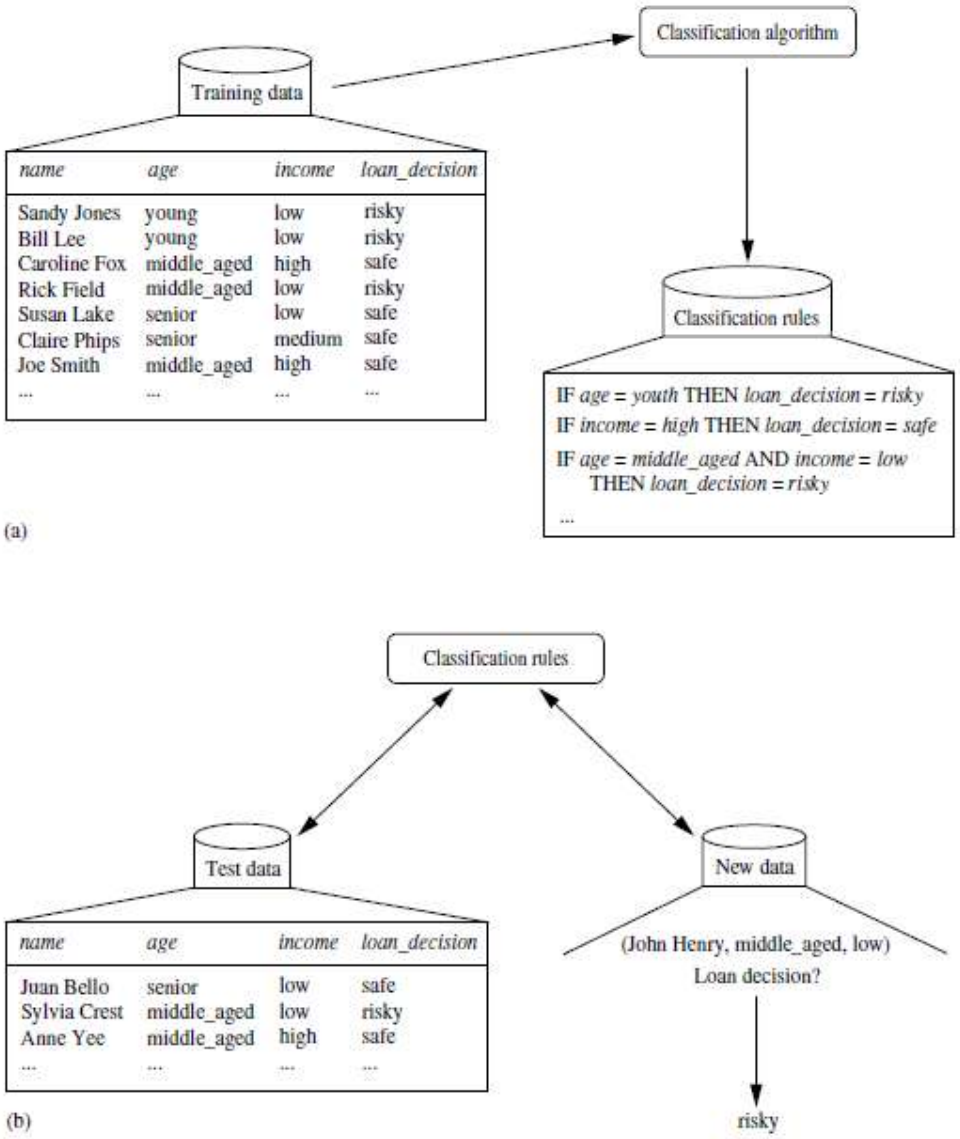


**Figure 6.1** The data classification process: (a) *Learning*: Training data are analyzed by a classification algorithm. Here, the class label attribute is *loan_decision*, and the learned model or classifier is represented in the form of classification rules. (b) *Classification*: Test data are used 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.

**Table 6.1** 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 |

**Example 6.4** Predicting a class label using naïve Bayesian classification. We wish to predict the class label of a tuple using naïve Bayesian classification, given the same training data as in Example 6.3 for decision tree induction. The training data are in Table 6.1. The data tuples are described by the attributes *age*, *income*, *student*, and *credit_rating*. The class label attribute, *buys_computer*, has two distinct values (namely, {*yes*, *no*}). Let $C_1$ correspond to the class *buys_computer* = *yes* and $C_2$ correspond to *buys_computer* = *no*. The tuple we wish to classify is

$$X = (age = youth, income = medium, student = yes, credit\_rating = fair)$$

We need to maximize $P(X|C_i)P(C_i)$, for $i = 1, 2$. $P(C_i)$, the prior probability of each class, can be computed based on the training tuples:

$$P(buys\_computer = yes) = 9/14 = 0.643$$
$$P(buys\_computer = no) = 5/14 = 0.357$$

To compute $PX|C_i)$, for $i = 1, 2$, we compute the following conditional probabilities:

$$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$$
$$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$$
$$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$$
$$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$$
$$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$$
$$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$$
$$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$$

Using the above probabilities, we obtain

$$\begin{aligned} P(X|buys\_computer = yes) &= P(age = youth \mid buys\_computer = yes) \times \\ &\quad P(income = medium \mid buys\_computer = yes) \times \\ &\quad P(student = yes \mid buys\_computer = yes) \times \\ &\quad P(credit\_rating = fair \mid buys\_computer = yes) \\ &= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044. \end{aligned}$$

Similarly,

$$P(X|buys\_computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$$

To find the class, $C_i$, that maximizes $P(X|C_i)P(C_i)$, we compute

$$P(X|buys\_computer = yes)P(buys\_computer = yes) = 0.044 \times 0.643 = 0.028$$
$$P(X|buys\_computer = no)P(buys\_computer = no) = 0.019 \times 0.357 = 0.007$$

Therefore, the naïve Bayesian classifier predicts *buys_computer* = *yes* for tuple $X$. ■

## 6.5.1 Using IF-THEN Rules for Classification

Rules are a good way of representing information or bits of knowledge. A **rule-based classifier** uses a set of IF-THEN rules for classification. An **IF-THEN** rule is an expression of the form

IF *condition* THEN *conclusion*.

An example is rule *R*1,

R1: IF *age* = *youth* AND *student* = *yes* THEN *buys_computer* = *yes*.

The "IF"-part (or left-hand side) of a rule is known as the **rule antecedent** or **precondition**. The "THEN"-part (or right-hand side) is the **rule consequent**. In the rule antecedent, the condition consists of one or more *attribute tests* (such as *age* = *youth*, and *student* = *yes*) that are logically ANDed. The rule's consequent contains a class prediction (in this case, we are predicting whether a customer will buy a computer). *R*1 can also be written as

R1: (*age* = *youth*) ∧ (*student* = *yes*) ⇒ (*buys_computer* = *yes*).

If the condition (that is, all of the attribute tests) in a rule antecedent holds true for a given tuple, we say that the rule antecedent is **satisfied** (or simply, that the rule is satisfied) and that the rule **covers** the tuple.

A rule *R* can be assessed by its coverage and accuracy. Given a tuple, *X*, from a class-labeled data set, *D*, let $n_{covers}$ be the number of tuples covered by *R*; $n_{correct}$ be the number of tuples correctly classified by *R*; and $|D|$ be the number of tuples in *D*. We can define the **coverage** and **accuracy** of *R* as

$$coverage(R) = \frac{n_{covers}}{|D|} \tag{6.19}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}. \tag{6.20}$$

That is, a rule's coverage is the percentage of tuples that are covered by the rule (i.e., whose attribute values hold true for the rule's antecedent). For a rule's accuracy, we look at the tuples that it covers and see what percentage of them the rule can correctly classify.

# References

[1] Data Mining: Concepts and Techniques, Second Edition" Jiawei Han and Micheline Kamber"

[2] .M.H. Dunham, " Data Mining Introductory and Adv anced Topics", Pear son Education Jiawei Han and Micheline Kamber