# Example 3: Gini index (CART)

| *TID* | *age* | *income* | *student* | *credit_rating* | *Class: buy_computer* |
|---|---|---|---|---|---|
| 9 | youth | low | yes | fair | yes |
| 11 | youth | medium | yes | excellent | yes |
| 1 | youth | high | no | fair | no |
| 2 | youth | high | no | excellent | no |
| 8 | youth | medium | no | fair | no |
| 7 | middle_aged | low | yes | excellent | yes |
| 13 | middle_aged | high | yes | fair | yes |
| 3 | middle_aged | high | no | fair | yes |
| 12 | middle_aged | medium | no | excellent | yes |
| 5 | senior | low | yes | fair | yes |
| 10 | senior | medium | yes | fair | yes |
| 4 | senior | medium | no | fair | yes |
| 6 | senior | low | yes | excellent | no |
| 14 | senior | medium | no | excellent | no |

## 2. Naïve Bayesian Classification

- **Example 4 Predicting a class label using naïve Bayesian classification**. We want to predict class label of a tuple using NBC, given training data *D* shown in Table 8.1. Data tuples are described by attributes *age*, *income*, *student*, and *credit rating*. Class label attribute, *buys_computer*, has two distinct values (namely, {*yes*, *no*}). Let $C_1$ correspond to class *buys_computer* = *yes* and $C_2$ correspond to *buys_computer* = *no*. The tuple we want to classify is

## 2. Naïve Bayesian Classification

$$\boldsymbol{X} = (age = youth, income = medium,$$
$$student = yes, credit\_rating = fair)$$
$$\text{i.e., } \boldsymbol{X} = (x_1 = youth, x_2 = medium, x_3 = yes,$$
$$x_4 = fair)$$

- We need to maximize $P(\boldsymbol{X} \mid C_i)P(C_i)$, for $i = 1, 2$. $P(C_i)$, the prior probability of each class, can be computed based on training tuples in $D$.

## 2. Naïve Bayesian Classification

• Remainder: $P(C_i) = |C_{i,D}|/|D|$, where $|C_{i,D}|$ is the number of training tuples of class $C_i$ in $D$.

$P(C_1) = |C_{1,D}|/|D| = P(buys\_computer = yes)$

$= 9/14 = 0.643$

$P(C_2) = |C_{2,D}|/|D| = P(buys\_computer = no)$

$= 5/14 = 0.357$

• To compute $P(\boldsymbol{X} \mid C_i)$, for $i = 1, 2$, we compute the following conditional probabilities.

## 2. Naïve Bayesian Classification

- Remainder: $P(\boldsymbol{X} \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)$ from the training tuples $D$, where $x_k$ is the value of attribute $A_k$ for the given tuple $\boldsymbol{X}$ and $P(x_k \mid C_i) = |C_{i,x_k}| / |C_{i,D}|$.

## 2. Naïve Bayesian Classification

• Compute $P(x_k \mid C_1) = \mid C_{1,x_k} \mid / \mid C_{1,D} \mid$,

where $x_1 = youth$, $x_2 = medium$, $x_3 = yes$, $x_4 = fair$

$P(age = youth \mid buys\_computer = yes) = 2/9 = 0.222$

$P(income = medium \mid buys\_computer = yes) = 4/9 = 0.444$

$P(student = yes \mid buys\_computer = yes) = 6/9 = 0.667$

$P(credit\_rating = fair \mid buys\_computer = yes) = 6/9 = 0.667$

# 2. Naïve Bayesian Classification

- Compute $P(x_k \mid C_2) = |C_{2,x_k}| / |C_{2,D}|$,

where $x_1 = youth$, $x_2 = medium$, $x_3 = yes$, $x_4 = fair$

$P(age = youth \mid buys\_computer = no) = 3/5 = 0.600$

$P(income = medium \mid buys\_computer = no) = 2/5 = 0.400$

$P(student = yes \mid buys\_computer = no) = 1/5 = 0.200$

$P(credit\_rating = fair \mid buys\_computer = no) = 2/5 = 0.400$

## 2. Naïve Bayesian Classification

- Using the probabilities $P(x_k \mid C_i)$ computed above, we obtain $P(\mathbf{X} \mid C_1)$

$$P(\mathbf{X} \mid buys\_computer = yes) = P(age = youth \mid buys\_computer = yes)$$
$$\times P(income = medium \mid buys\_computer = yes)$$
$$\times P(student = yes \mid buys\_computer = yes)$$
$$\times P(credit\_rating = fair \mid buys\_computer = yes)$$
$$= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044.$$

## 2. Naïve Bayesian Classification

- Similarly, we obtain $P(\boldsymbol{X} \mid C_2)$

$P(\boldsymbol{X} \mid buys\_computer = no) = 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019.$

## 2. Naïve Bayesian Classification

- To find the class $C_i$ that maximizes $P(\boldsymbol{X} \mid C_i)P(C_i)$, we compute

$$P(\boldsymbol{X} \mid C_1)P(C_1) =$$

$$P(\mathbf{X} \mid buys\_computer = yes)P(buys\_computer = yes)$$

$$= 0.044 \times 0.643 = 0.028$$

$$P(\boldsymbol{X} \mid C_2)P(C_2) =$$

$$P(\mathbf{X} \mid buys\_computer = no)P(buys\_computer = no)$$

$$= 0.019 \times 0.357 = 0.007$$

## 2. Naïve Bayesian Classification

- $P(\boldsymbol{X} \mid C_1)P(C_1) = 0.028 > P(\boldsymbol{X} \mid C_2)P(C_2) = 0.007$. Thus, NBC predicts *buys_computer* = *yes* for the given tuple $\boldsymbol{X}$.

## 2. Naïve Bayesian Classification

- Recall: $P(\boldsymbol{X} \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)$
- If $P(x_k \mid C_i) = |C_{i,x_k}| / |C_{i,D}|$ = zero for some $k$, then $P(\boldsymbol{X} \mid C_i)$ is zero.
- Example: $P(student = yes \mid buys\_computer = no) = 0$

## 2. Naïve Bayesian Classification

- Solution to the problem of $P(x_k \mid C_i) = 0$ for some $k$. We can assume that training data set $D$ is so large that adding one to each count that we need would only make a negligible difference in the estimated probability value, yet would conveniently avoid the case of $P(x_k \mid C_i) = 0$.

## 2. Naïve Bayesian Classification

- The above technique to deal with $P(x_k \mid C_i) = 0$ is known as **Laplacian correction** or **Laplace estimator**. Specifically, if we have $q$ counts to which we each add one, then we must remember to add $q$ to the corresponding denominator used in the probability calculation.

# 2. Naïve Bayesian Classification

- **Example 5 Using the Laplacian correction to avoid computing probability values of zero**. Suppose that for class *buys_computer* = *yes* in some training data set *D* containing 1000 tuples, we have 0 tuples with *income* = *low*, 990 tuples with *income* = *medium*, and 10 tuples with *income* = *high*.
- The probabilities of these events, without the Laplacian correction, are 0/1000 = 0, 990/1000 = 0.990, and 10/1000 = 0.010, respectively.

## 2. Naïve Bayesian Classification

- Using Laplacian correction for the three quantities, we pretend that we have 1 more tuple for each income-value pair. In this way, we obtain the following probabilities:

1/1003 = 0.001, 991/1003 = 0.988, and 11/1003 = 0.011, respectively.

- The "corrected" probability estimates are close to their "uncorrected" counterparts, yet the zero probability value is avoided.