cse634
DATA MINING

Lecture Notes
Chapter 6: CLASSIFICATION by DECISION TREES

Introduction
BASIC ALGORITHM
Examples

Professor Anita Wasilewska
Computer Science Department
Stony Brook University
Classification Learning ALGORITHMS
Different Classifiers

- **DESCRIPTIVE:**
  - Decision Trees (ID3, C4.5)
  - Rough Sets
  - Genetic Algorithms
  - Classification by Association

- **STATISTICAL:**
  - Neural Networks
  - Bayesian Networks
Classification Data

- **Data format:** a data table with key attribute removed. Special attribute- class attribute must be distinguished.

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit_rating</th>
<th>buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31...40</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31...40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31...40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31...40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>
### Classification (Training) Data with objects

<table>
<thead>
<tr>
<th>rec</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_rating</th>
<th>Buys_computer (CLASS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r2</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r3</td>
<td>31...40</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r4</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r6</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r7</td>
<td>31...40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r8</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r9</td>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r10</td>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r11</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r12</td>
<td>31...40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r13</td>
<td>31...40</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r14</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
</tbody>
</table>
Classification by Decision Tree Induction

- Decision tree is a flow-chart-like tree structure;
  - Internal node denotes an attribute;
  - Branch represents the values of the node attribute;
  - Leaf nodes represent class labels or class distribution
DECISION TREE
An Example

- age
  - <=30: student (Bbuys=no)
  - 31..40: Buys=yes
  - >40: credit rating
    - excellent: Buys=no
    - fair: Buys=yes
Classification by Decision Tree Induction

Basic Algorithm

• The basic algorithm for decision tree construction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner.

• Given a training set $D$ of classification data, i.e.
  • a data table with a distinguished class attribute

• This training set is recursively partitioned into smaller subsets (data tables) as the tree is being built.
Classification by Decision Tree Induction

Basic Algorithm

- **Tree STARTS** as a single node (root) representing all training dataset D (samples)

- We choose a **root attribute** from D
- It is called a **SPLIT** attribute

- A branch is created for each value as defined in D of the **node attribute** and is labeled by this values and the samples (it means the data table) are partitioned accordingly

- The **algorithm** uses the same process recursively to form a **decision tree** at each partition

- Once an attribute has occurred at a node, it need not be considered in any other of the node’s descendants
Classification by Decision Tree Induction

Basic Algorithm

• The **recursive partitioning** **STOPS** only when any **one** of the following conditions is **true**

  1. **All the samples** (records) in the partition are of the **same class**, then the node becomes **the leaf labeled with that class**

  2. **There is no remaining attributes** on which the data may be further **partitioned**, i.e. **we have only class attribute left**. In this case, we apply **MAJORITY VOTING** to **classify** the node. **MAJORITY VOTING** involves converting the node into a **leaf** and labeling it with the **most common class** in the **training data set**

  3. **There is no records** (samples) left – a **LEAF** is created with **majority vote** for **training data set**
Classification by Decision Tree Induction

**Crucial point**

**Good choice** of the root attribute and internal nodes attributes is a crucial point.

**Bad choice** may result, in the worst case in a just another knowledge representation: a relational table re-written as a tree with class attributes (decision attributes) as the leaves.

- **Decision Tree Algorithms** differ on methods of evaluating and choosing the root and internal nodes attributes.
Consider our TRAING Dataset (next slide)

We START building the Decision Tree by choosing the attribute age as the root of the tree
## Training Data with objects

<table>
<thead>
<tr>
<th>rec</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_rating</th>
<th>Buys_computer(CLASS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r2</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r3</td>
<td>31...40</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r4</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r6</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r7</td>
<td>31...40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r8</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r9</td>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r10</td>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r11</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r12</td>
<td>31...40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r13</td>
<td>31...40</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r14</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
</tbody>
</table>
Building The Tree: we choose “age” as a root

```
<table>
<thead>
<tr>
<th>income</th>
<th>student</th>
<th>credit</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

<=30

>40

31…40

<table>
<thead>
<tr>
<th>income</th>
<th>student</th>
<th>credit</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
</tbody>
</table>
```
Building The Tree: “age” as the root

- **age**
  - <=30
    - income | student | credit | class
      - high | no | fair | no
      - high | no | excellent | no
      - medium | no | fair | no
      - low | yes | fair | yes
      - medium | yes | excellent | yes
  - >40
    - income | student | credit | class
      - medium | no | fair | yes
      - low | yes | fair | yes
      - low | yes | excellent | no
      - medium | yes | fair | yes
      - medium | no | excellent | no

- 31…40
  - class=yes
Building The Tree: we chose “student” on <=30 branch

<table>
<thead>
<tr>
<th>income</th>
<th>student</th>
<th>credit</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>

- **<=30**
  - **student**
    - **no**
      - **in cr cl**
        - h f n
    - **yes**
      - **in cr cl**
        - l f y
        - m e y

- **>40**
  - 31…40
  - **class=yes**
Building The Tree: we chose “student” on <=30 branch

- age
  - <=30
    - student
      - no (class= no)
      - yes (class= yes)
  - >40
    - income | student | credit | class
      - medium no fair yes
      - low yes fair yes
      - low yes excellent no
      - medium yes fair yes
      - medium no excellent no
    - 31…40
      - class= yes
Building The Tree: we chose “credit” on >40 branch

```
- student
  - no
    - class= no
  - yes
    - class= yes

- age
  - <=30
    - student
    - no
      - class= no
    - yes
      - class= yes
  - >40

- credit
  - excellent
    - in st cl
      - l y n
      - m n n
  - fair
    - in st cl
      - m n y
      - l y y
      - m y y

```

31…40

```
- class= yes
```

Finished Tree for class="buys"

- Age <= 30
  - Student
    - No: buys = no
    - Yes: buys = yes
  - Yes: buys = yes
- Age > 40
  - Credit
    - Excellent: buys = no
    - Fair: buys = yes
- Age 31...40: buys = yes
Extracting **Classification Rules** from Trees

- **Goal**: Represent the knowledge in the form of **IF-THEN discriminant** rules
- **One rule** is created for **each path** from the **root** to a **leaf**;
- **Each attribute-value** pair along a **path** forms a conjunction;
- The **leaf node** holds the **class prediction**

- **Rules are easier to understand**
Discriminant **RULES** extracted from our TREE

- The rules are:

  IF age = “<=30” AND student = “no” THEN
  \[ \text{buys\_computer} = “no” \]
  IF age = “<=30” AND student = “yes” THEN
  \[ \text{buys\_computer} = “yes” \]
  IF age = “31…40” THEN
  \[ \text{buys\_computer} = “yes” \]
  IF age = “>40” AND credit\_rating = “excellent” THEN
  \[ \text{buys\_computer} = “no” \]
  IF age = “>40” AND credit\_rating = “fair” THEN
  \[ \text{buys\_computer} = “yes” \]
In order to use rules for testing, and later when testing is done and predictive accuracy is acceptable we write rules in a **predicate form**:

```
IF age(x, <=30) AND student(x, no)  THEN buys_computer (x, no)
IF age(x, <=30) AND student(x, yes) THEN buys_computer (x, yes)
```

- Attributes and their values of the **new record x** are **matched** with the **IF** part of the rule and the **record x is classified** accordingly to the **THEN** part of the rule.
Exercise

Calculate the **predictive accuracy** of our set of rules with respect to the TEST data given by the next slide.

R1: IF \( \text{age} = \text{"\(<=30\)" AND student = \"no\} \) THEN \( \text{buys\_computer} = \text{"no\} \)

R2: IF \( \text{age} = \text{"\(<=30\)" AND student = \"yes\} \) THEN \( \text{buys\_computer} = \text{"yes\} \)

R3: IF \( \text{age} = \text{"\(31…40\)"} \\) THEN \( \text{buys\_computer} = \text{"yes\} \)

R4: IF \( \text{age} = \text{"\(>40\)" AND credit\_rating = \"excellent\} \) THEN \( \text{buys\_computer} = \text{"no\} \)

R5: IF \( \text{age} = \text{"\(>40\)" AND credit\_rating = \"fair\} \) THEN \( \text{buys\_computer} = \text{"yes\} \)
TEST Data
for predictive accuracy evaluation

<table>
<thead>
<tr>
<th>rec</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_rating</th>
<th>Buys_computer(CLASS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&lt;=30</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>yes</td>
</tr>
<tr>
<td>r2</td>
<td>&lt;=30</td>
<td>High</td>
<td>yes</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r3</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r4</td>
<td>31…40</td>
<td>Medium</td>
<td>yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r6</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>yes</td>
</tr>
<tr>
<td>r7</td>
<td>31…40</td>
<td>High</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r8</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r9</td>
<td>31…40</td>
<td>Low</td>
<td>no</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r10</td>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Basic Idea of ID3/C4.5 Algorithm

- **The basic algorithm** for decision tree induction is a greedy algorithm that constructs decision trees in a top-down recursive divide-and-conquer manner.

- The **basic strategy** is as follows.

- Tree **STARTS** as a single **node** representing all training dataset (data table with records called **samples**)

- **IF** the samples (records in the data table) are all in the same class, **THEN** the node becomes a **leaf** and is labeled with **that class**
Basic Idea of ID3/C4.5 Algorithm

• OTHERWISE
  • the algorithm uses an entropy-based measure known as information gain as a heuristic for selecting the attribute that will best separate the samples: split the data table into individual classes

• This attribute becomes the node-name: test, or tree split decision attribute

• A branch is created for each value of the node-attribute (as defined by the training data) and is labeled by this value and the samples (data table at the node) are partitioned accordingly
**Basic Idea of ID3/C4.5 Algorithm Revisited**

- The algorithm uses the same process recursively to form a decision tree at each partition.
- Once an attribute has occurred at a node, it need not be considered in any other of the node’s descendants.
- The recursive partitioning stops only when any one of the following conditions is TRUE.
Basic Idea of ID3/C4.5 Algorithm

Termination conditions:

1. All records (samples) for the given node belong to the same class

2. There are no remaining attributes left on which the samples (records in the data table) may be further partitioned. In this case we convert the given node into a LEAF and label it with the class in majority among original training samples.
   • This is called a majority voting
   • OR

3. There is no records (samples) left – a LEAF is created with majority vote for training sample
Heuristics: Attribute Selection Measures

• **Construction** of the tree **depends** on the **order** in which root attributes are **selected**

• **Different choices** produce **different trees**; some better, some worse

• **Shallower** trees are **better**; they are the ones in which classification is **reached** in **fewer levels**

• These trees are said to be **more efficient** and hence **termination** is reached **quickly**
Attribute Selection Measures

- Given a training data set (set of training samples) there are many ways to choose the root and nodes attributes while constructing the decision tree.

- **Some possible choices:**
  - Random
  - Attribute with smallest/largest number of values
  - Following certain order of attributes
  - We present here a special order: information gain as a measure of goodness of the split
  - The attribute with the highest information gain is always chosen as the split decision attribute for the current node while building the tree.
Information Gain Computation (ID3/C4.5): Case of Two Classes

• Assume there are two classes, $P$ (positive) and $N$ (negative)

Let $S$ be a training data set consisting of $s$ examples (records):

$|S| = s$

And $S$ contains $p$ elements of class $P$ and $n$ elements of class $N$

The amount of information, needed to decide if an arbitrary example in $S$ belongs to $P$ or $N$ is defined as

$$I(p, n) = - \frac{p}{p + n} \log_2 \frac{p}{p + n} - \frac{n}{p + n} \log_2 \frac{n}{p + n}$$

• We use $\log_2$ because the information is encoded in bits
Information Gain Measure

- Assume that using attribute \( A \) a set \( S \) will be partitioned into sets \( S_1, S_2, \ldots, S_v \) (\( v \) is number of values of the attribute \( A \))

  If \( S_i \) contains \( p_i \) examples of \( P \) and \( n_i \) examples of \( N \) the entropy \( E(A) \), or the **expected information** needed to classify objects in all sub-trees \( S_i \) is

  \[
  E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)
  \]

- The encoding information that would be **gained** by branching on \( A \)

  \[
  Gain(A) = I(p, n) - E(A)
  \]
Attribute Selection: Information Gain

Data Mining Book slide

- **Class P**: buys_computer = "yes"
- **Class N**: buys_computer = "no"

\[ Info(D) = I(9,5) = -\frac{9}{14} \log_2\left(\frac{9}{14}\right) - \frac{5}{14} \log_2\left(\frac{5}{14}\right) = 0.940 \]

<table>
<thead>
<tr>
<th>age</th>
<th>( p_i )</th>
<th>( n_i )</th>
<th>( I(p_i, n_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>2</td>
<td>3</td>
<td>0.971</td>
</tr>
<tr>
<td>31...40</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&gt;40</td>
<td>3</td>
<td>2</td>
<td>0.971</td>
</tr>
</tbody>
</table>

\[ Info_{age}(D) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694 \]

\( I(2,3) \) means "age <=30" has 5 out of 14 samples, with 2 yes’es and 3 no’、“. Hence

\[ Gain(age) = Info(D) - Info_{age}(D) = 0.246 \]

Similarly,

\[ Gain(income) = 0.029 \]
\[ Gain(student) = 0.151 \]
\[ Gain(credit\_rating) = 0.048 \]
Attribute Selection by Information Gain Computation

- **Class P**: buys_computer = “yes”
- **Class N**: buys_computer = “no”
- \( I(p, n) = I(9, 5) = 0.940 \)
- Compute the entropy for

<table>
<thead>
<tr>
<th>age</th>
<th>( p_i )</th>
<th>( n_i )</th>
<th>( I(p_i, n_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \leq 30 )</td>
<td>2</td>
<td>3</td>
<td>0.971</td>
</tr>
<tr>
<td>31...40</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&gt;40</td>
<td>3</td>
<td>2</td>
<td>0.971</td>
</tr>
</tbody>
</table>

\[
E(age) = \frac{5}{14} I(2,3) + \frac{4}{14} I(4,0) + \frac{5}{14} I(3,2) = 0.694
\]

Hence

\[
Gain(age) = I(p, n) - E(age)
\]

Gain\( (age) = 0.246 \)

Similarly

\[
Gain(income) = 0.029
\]

\[
Gain(student) = 0.151
\]

\[
Gain(credit\_rating) = 0.048
\]

The attribute “age” becomes the root.
Decision Tree Induction, Predictive Accuracy and Information Gain

EXAMPLES
TASK: Use Decision Tree Induction algorithm and use different choices of the root and nodes attributes to find discriminant rules that determine whether a person buys a computer or not.

Compute Information gain for all nodes of the tree.

1. We choose attribute *buys_computer* as the class attribute.
2. We perform DT algorithm “by hand” using different choices of the root attribute, and different “by hand” choices of the following nodes.
3. We build two trees with attributes: *Income* and *Credit Rating* respectively, as the root attribute to derive rules.
### Training Data

<table>
<thead>
<tr>
<th>age</th>
<th>income</th>
<th>student</th>
<th>credit_rating</th>
<th>buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>high</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31…40</td>
<td>high</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>no</td>
</tr>
<tr>
<td>31…40</td>
<td>low</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>no</td>
<td>fair</td>
<td>no</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>low</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>medium</td>
<td>yes</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31…40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>yes</td>
</tr>
<tr>
<td>31…40</td>
<td>high</td>
<td>yes</td>
<td>fair</td>
<td>yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>medium</td>
<td>no</td>
<td>excellent</td>
<td>no</td>
</tr>
</tbody>
</table>
# Training Data with objects

<table>
<thead>
<tr>
<th>rec</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_rating</th>
<th>Buys_computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r2</td>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r3</td>
<td>31...40</td>
<td>High</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r4</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r6</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>No</td>
</tr>
<tr>
<td>r7</td>
<td>31...40</td>
<td>Low</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r8</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>r9</td>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r10</td>
<td>&gt;40</td>
<td>Medium</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r11</td>
<td>&lt;=30</td>
<td>Medium</td>
<td>Yes</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r12</td>
<td>31...40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>Yes</td>
</tr>
<tr>
<td>r13</td>
<td>31...40</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>r14</td>
<td>&gt;40</td>
<td>Medium</td>
<td>No</td>
<td>Excellent</td>
<td>No</td>
</tr>
</tbody>
</table>
EXAMPLE 2 Incorrect Solutions

• BOTH TREES of the following Example 2 Solutions ARE NOT CORRECT !!!

• FIND STEPS where the construction didn’t follow the ALGORITHM and CORRECT THEM

• Write the CORRECT Solutions for the EXAMPLE 2

• Perform Exercises 1 and 2 for the corrected trees
### Decision Tree

**Income**

**Index: 1**

**Gain=0.027**

#### Low

<table>
<thead>
<tr>
<th>Age</th>
<th>Student</th>
<th>Credit</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;40</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>31-40</td>
<td>Yes</td>
<td>Exc</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
</tbody>
</table>

#### Med

<table>
<thead>
<tr>
<th>Age</th>
<th>Student</th>
<th>Credit</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;40</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Yes</td>
<td>Exc</td>
<td>Yes</td>
</tr>
<tr>
<td>31-40</td>
<td>No</td>
<td>Exc</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>No</td>
<td>Exc</td>
<td>No</td>
</tr>
</tbody>
</table>

#### High

<table>
<thead>
<tr>
<th>Age</th>
<th>Student</th>
<th>Credit</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>No</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>31-40</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>31-40</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**CORRECT? – INCORRECT?**
**CORRECT? – INCORRECT?**

**Income**
- Low
- Med
  - High

**Credit**
- Fair
- Exec

**Age**
- >40
- <=30

**Student**
- Yes

**Class**
- No
- Yes

- **Ind:2**
  - **0.01**
  - **Ind:3**
  - **Ind:4**

**Age**
- >40
  - No
  - Yes
- <=30
  - No
  - Yes

**Student**
- Yes

**Credit**
- Fair
- Exc

**Class**
- No
- Yes

---

**Income**
- 0.027

**Credit**
- 0.01

**Age**
- 0.59
  - <=30
  - >40

**Student**
- 0.316
  - Yes
  - No

---

**Age**
- 31-40

**Credit**
- Fair
- Exc

**Class**
- Yes
- No

---

<table>
<thead>
<tr>
<th>Age</th>
<th>Student</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;40</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Age</th>
<th>Credit</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-40</td>
<td>Fair</td>
<td>Yes</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Stud</th>
<th>Credit</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>Yes</td>
<td>Exc</td>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
<td>Exc</td>
<td>No</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Stud</th>
<th>Credit</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>Exc</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Tree 1 with root attribute Income

- Income
  - Low
    - Credit
      - Fair
        - YES
          - Age
            - Yes
              - Credit
                - Yes
                  - Age
                    - <=30
                      - YES
                      - NO
                    - 31-40
                      - NO
                    - >40
                      - YES
                      - NO
                - Exc
                  - NO
  - Med
    - High
      - Age
        - 31-40
          - Yes
            - Credit
              - Yes
                - Credit
                  - Fair
                    - Yes
                      - NO
                      - NO
                    - <=30
                      - NO
                    - 31-40
                      - NO
                  - Exc
                    - NO
                  - Fair
                    - Yes
                      - YES
                      - NO
                    - Exc
                      - NO
          - NO
  - NO
Rules derived from tree 1 (predicate form for testing)

1. Income(x, Low) ^ Credit(x, Fair) -> buysComputer(x, Yes).
2. Income(x, Low) ^ Credit(x, Exc) ^ Age(x, 31-40) -> buysComputer(x, Yes).
3. Income(x, Low) ^ Credit(x, Exc) ^ Age(>40) -> buysComputer(x, No).
4. Income(x, High) ^ Student(x, Yes) -> buysComputer(x, Yes).
5. Income(x, High) ^ Student(x, No) ^ Age(x, <=30) -> buysComputer(x, No).
6. Income(x, High) ^ Student(x, No) ^ Age(x, 31-40) -> buysComputer(x, Yes).
7. Income(x, Medium) ^ Age(x, 31-40) -> buysComputer(x, Yes).
8. Income(x, Medium) ^ Age(x, <=30) ^ Credit(x, Fair) -> buysComputer(x, No).
9. Income(x, Medium) ^ Age(x, <=30) ^ Credit(x, Exc) -> buysComputer(x, Yes).
10. Income(x, Medium) ^ Age(x, >40) ^ Credit(x, Fair) -> buysComputer(x, Yes).
11. Income(x, Medium) ^ Age(x, >40) ^ Credit(x, Exc) -> buysComputer(x, No).
Tree 2 with root attribute Credit Rating

CORRECT? – INCORRECT?
Tree 2 with next level attributes Income and Student
CORRECT? – INCORRECT?

Tree 2 with root attribute Credit Rating
Final Tree 2 with root attribute Credit Rating
The Decision tree with root attribute \textit{Credit\_Rating} has produced 13 rules, two more than with root attribute \textit{Income}

1. \( \text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Low}) \rightarrow \text{buysComp}(x, \text{Yes}) \).
2. \( \text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{High}) \land \text{Student}(x, \text{Yes}) \rightarrow \text{buysComp}(x, \text{Yes}) \).
3. \( \text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{High}) \land \text{Student}(x, \text{No}) \land \text{Age}(<30) \rightarrow \text{buysComp}(x, \text{No}) \).
4. \( \text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{High}) \land \text{Student}(x, \text{No}) \land \text{Age}(31-40) \rightarrow \text{buysComp}(x, \text{Yes}) \).
5. \( \text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Med}) \land \text{Student}(x, \text{Yes}) \rightarrow \text{buysComp}(x, \text{Yes}) \).
6. \( \text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Med}) \land \text{Student}(x, \text{No}) \land \text{Age}(>40) \rightarrow \text{buysComp}(x, \text{Yes}) \).
7. \( \text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Med}) \land \text{Student}(x, \text{No}) \land \text{Age}(<30) \rightarrow \text{buysComp}(x, \text{No}) \).
8. \( \text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{Low}) \land \text{Age}(31-40) \rightarrow \text{buysComp}(x, \text{Yes}) \).
9. \( \text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{Low}) \land \text{Age}(>40) \rightarrow \text{buysComp}(x, \text{No}) \).
10. \( \text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{Med}) \rightarrow \text{buysComp}(x, \text{Yes}) \).
11. \( \text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{Med}) \land \text{Age}(31-40) \rightarrow \text{buysComp}(x, \text{Yes}) \).
12. \( \text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{Med}) \land \text{Age}(>40) \rightarrow \text{buysComp}(x, \text{No}) \).
13. \( \text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{High}) \rightarrow \text{buysComp}(x, \text{No}) \).
EXERCISE 1

• We use some random records (tuples) to calculate the Predictive Accuracy of the set of rules from the Example 2

Predictive Accuracy is the % of well classified records not from training set for which the class attribute is known
Random Tuples to Check Predictive Accuracy based on three sets of rules

<table>
<thead>
<tr>
<th>Obj</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_R</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;=30</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>31-40</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>31-40</td>
<td>High</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>&lt;=30</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
</tbody>
</table>

Predictive accuracy:
1. Against Lecture Notes: 4/6 = \textbf{66.66\%}
2. Against Tree 1 rules with root att. \textit{Income}: 3/6 = \textbf{50\%}
3. Against Tree 2 rules with root att. \textit{Credit}: 5/6 = \textbf{83.33\%}
EXERCISE 2

• Predictive accuracy depends heavily on a choice of the test and training data.

• Find a small set of TEST records such that they would give a predictive accuracy 100% for rules From the Lecture Tree and Trees 1 and 2 from Example 1
1. TEST DATA applied against rules in Lecture Notes that gives predictive accuracy 100%

<table>
<thead>
<tr>
<th>No</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_R</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;=30</td>
<td>Med</td>
<td>No</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>&lt;=30</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>31-40</td>
<td>Low</td>
<td>No</td>
<td>Exc</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>&gt;40</td>
<td>High</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>&lt;=30</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>31-40</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
</tbody>
</table>
2. **TEST DATA** that applied against the rules with root attribute *Income* give **predictive accuracy 100%**

<table>
<thead>
<tr>
<th>No</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_R</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31-40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>&gt;40</td>
<td>Low</td>
<td>No</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>&lt;=30</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>31-40</td>
<td>High</td>
<td>No</td>
<td>Exc</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>31-40</td>
<td>Med</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>&gt;40</td>
<td>Med</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
</tbody>
</table>
3. TEST DATA that applied against the rules with root attribute *Credit Rating* gives predictive accuracy 100%

<table>
<thead>
<tr>
<th>No</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_R</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31-40</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>&lt;=30</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>&lt;=30</td>
<td>Med</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>31-40</td>
<td>High</td>
<td>Yes</td>
<td>Exc</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>&gt;40</td>
<td>Med</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>&gt;40</td>
<td>Med</td>
<td>No</td>
<td>Exc</td>
<td>No</td>
</tr>
</tbody>
</table>
Exercise 2  Corrections

We FIXED the following two points of the Tree construction:

1. We choose recursively internal nodes (attributes) with all of their proper values as branches.

Mistake: NOT ALL attributes values were always used.

2. there is no more samples (records) left.
   In this case we apply Majority Voting to classify the node, where the Majority Voting involves converting the node into a leaf and labeling it with the most common class in the training set.

Mistake: NO MAJORITY Voting was used.
CORRECT Tree 1 with root attribute Income

Income
- Low
  - Credit
    - Fair
      - YES
    - Exc
      - YES
      - Age
        - 31-40
          - YES
          - NO
          - >=30
            - YES
        - >40
          - NO
          - >40
            - NO
            - >=30
              - NO
              - 31-40
                - Credit
                  - Fair
                    - YES
                  - Exc
                    - NO
                - >40
                  - Fair
                    - YES
                  - Exc
                    - NO
        - High
          - Student
            - Yes
              - Age
                - 31-40
                  - Credit
                    - Fair
                      - YES
                    - Exc
                      - NO
                - >40
                  - Credit
                    - Fair
                      - YES
                    - Exc
                      - NO
          - No
            - Age
              - 31-40
                - Credit
                  - Fair
                    - YES
                  - Exc
                    - NO
              - >40
                - Credit
                  - Fair
                    - YES
                  - Exc
                    - NO
Rules derived from Tree 1 (predicate form for testing)

1. Income(x, Low) $\land$ Credit(x, Fair) $\rightarrow$ buysComputer(x, Yes).
2. Income(x, Low) $\land$ Credit(x, Exc) $\land$ Age(x, 31-40) $\rightarrow$ buysComputer(x, Yes).
3. Income(x, Low) $\land$ Credit(x, Exc) $\land$ Age(>40) $\rightarrow$ buysComputer(x, No).
4. Income(x, High) $\land$ Student(x, Yes) $\rightarrow$ buysComputer(x, Yes).
5. Income(x, High) $\land$ Student(x, No) $\land$ Age(x, <=30) $\rightarrow$ buysComputer(x, No).
6. Income(x, High) $\land$ Student(x, No) $\land$ Age(x, 31-40) $\rightarrow$ buysComputer(x, Yes).
7. Income(x, Medium) $\land$ Age(x, 31-40) $\rightarrow$ buysComputer(x, Yes).
8. Income(x, Medium) $\land$ Age(x, <=30) $\land$ Credit(x, Fair) $\rightarrow$ buysComputer(x, No).
9. Income(x, Medium) $\land$ Age(x, <=30) $\land$ Credit(x, Exc) $\rightarrow$ buysComputer(x, Yes).
10. Income(x, Medium) $\land$ Age(x, >40) $\land$ Credit(x, Fair) $\rightarrow$ buysComputer(x, Yes).
11. Income(x, Medium) $\land$ Age(x, >40) $\land$ Credit(x, Exc) $\rightarrow$ buysComputer(x, No).
12. Income(x, Low) $\land$ Age(x, <=30) $\land$ Credit(x, Exc) $\rightarrow$ buysComputer(x, Yes). Majority Voting
13. Income(x, High) $\land$ Student(x, No) $\land$ Age(x>40) $\rightarrow$ buysComputer(x, Yes). Majority Voting
Tree 2 with root attribute Credit Rating

<table>
<thead>
<tr>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>31-40</td>
<td>High</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Med</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Med</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Low</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Med</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>31-40</td>
<td>High</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=30</td>
<td>High</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>31-40</td>
<td>Low</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt;=30</td>
<td>Med</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>31-40</td>
<td>Med</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>&gt;40</td>
<td>Med</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

CORRECT
Tree 2 with next level attributes Income and Student
**Tree 2 with root attribute Credit Rating**
CORRECTED
CORRECT
CORRECTED Tree 2 with root attribute Credit_Rating
The Decision tree with root attribute \textit{Credit\_Rating} has produced 13 rules, two more than with root attribute \textit{Income}.

1. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Low}) \rightarrow \text{buysComp}(x, \text{Yes}).\)
2. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{High}) \land \text{Student}(x, \text{Yes}) \rightarrow \text{buysComp}(x, \text{Yes}).\)
3. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{High}) \land \text{Student}(x, \text{No}) \land \text{Age}(\leq 30) \rightarrow \text{buysComp}(x, \text{No}).\)
4. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{High}) \land \text{Student}(x, \text{No}) \land \text{Age}(31-40) \rightarrow \text{buysComp}(x, \text{Yes}).\)
5. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Med}) \land \text{Student}(x, \text{Yes}) \rightarrow \text{buysComp}(x, \text{Yes}).\)
6. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Med}) \land \text{Student}(x, \text{No}) \land \text{Age}(>40) \rightarrow \text{buysComp}(x, \text{Yes}).\)
7. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Med}) \land \text{Student}(x, \text{No}) \land \text{Age}(\leq 30) \rightarrow \text{buysComp}(x, \text{No}).\)
8. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{Low}) \land \text{Age}(31-40) \rightarrow \text{buysComp}(x, \text{Yes}).\)
9. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{Low}) \land \text{Age}(>40) \rightarrow \text{buysComp}(x, \text{No}).\)
10. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{Med}) \rightarrow \text{buysComp}(x, \text{Yes}).\)
11. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{Med}) \land \text{Age}(x, 31-40) \rightarrow \text{buysComp}(x, \text{Yes}).\)
12. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{Med}) \land \text{Age}(x, >40) \rightarrow \text{buysComp}(x, \text{No}).\)
13. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{High}) \rightarrow \text{buysComp}(x, \text{No}).\)
14. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{High}) \land \text{Student}(x, \text{No}) \land \text{Age}(>40) \rightarrow \text{buysComp}(x, \text{Yes}).\) Majority Voting
15. \(\text{Credit}(x, \text{Fair}) \land \text{Income}(x, \text{Med}) \land \text{Student}(x, \text{No}) \land \text{Age}(31-40) \rightarrow \text{buysComp}(x, \text{Yes}).\) Majority Voting
16. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{Low}) \land \text{Age}(\leq 30) \rightarrow \text{buysComp}(x, \text{Yes}).\) Majority Voting
17. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{Med}) \land \text{Age}(x\leq30) \rightarrow \text{buysComp}(x, \text{Yes}).\) Majority Voting
18. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{Yes}) \land \text{Income}(x, \text{High}) \rightarrow \text{buysComp}(x, \text{Yes}).\) Majority Voting
19. \(\text{Credit}(x, \text{Exc}) \land \text{Student}(x, \text{No}) \land \text{Income}(x, \text{Low}) \rightarrow \text{buysComp}(x, \text{Yes}).\) Majority Voting
Random Tuples to Check Predictive Accuracy based on three sets of rules

<table>
<thead>
<tr>
<th>Obj</th>
<th>Age</th>
<th>Income</th>
<th>Student</th>
<th>Credit_R</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;=30</td>
<td>High</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>31-40</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>31-40</td>
<td>High</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Fair</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>&gt;40</td>
<td>Low</td>
<td>Yes</td>
<td>Exc</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>&lt;=30</td>
<td>Low</td>
<td>No</td>
<td>Fair</td>
<td>No</td>
</tr>
</tbody>
</table>

Predictive accuracy:
1. Against Lecture Notes: $4/6 = 66.66\%$
2. Against Tree 1 rules with root att. Income: $3/6 = 50\%$
3. Against Tree 2 rules with root att. Credit: $4/6 = 66.66\%$
4. Against OLD Tree 2 rules with root att. Credit: $5/6 = 83.33\%$
Calculation of Information gain at each level of tree with root attribute *Income*

1. **Original Table:**
   Class P: *buys_computer* = yes; Class N: *buys_computer* = No
   
   \[ I(P,N) = -\frac{P}{P+N} \log_2 \left( \frac{P}{P+N} \right) - \frac{N}{P+N} \log_2 \left( \frac{N}{P+N} \right) \]  
   
   \[ I(P,N) = I(9,5) = \left( -\frac{9}{9+5} \right) \log_2 \left( \frac{9}{9+5} \right) - \left( \frac{5}{9+5} \right) \log_2 \left( \frac{5}{9+5} \right) \]
   
   \[ = 0.940 \]

2. **Index:**

<table>
<thead>
<tr>
<th>Income</th>
<th>Pi</th>
<th>Ni</th>
<th>I(Pi,Ni)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>3</td>
<td>1</td>
<td>0.8111</td>
</tr>
<tr>
<td>Med</td>
<td>4</td>
<td>2</td>
<td>0.9234</td>
</tr>
<tr>
<td>High</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ E(\text{Income}) = \frac{4}{14} I(3,1) + \frac{6}{14} I(4,2) + \frac{4}{14} I(2,2) \]  

\[ I(3,1) = 0.8111 \text{ (Using equation 1)} \]
\[ I(4,2) = 0.9234 \text{ (Using equation 1)} \]
\[ I(2,2) = 1 \]

Contd…..
Information gain calculation for Index 1 contd:

Substituting the values in eq.2 we get,

\[ E(\text{Income}) = 0.2317 + 0.3957 + 0.2857 = 0.9131 \]

Gain (Income) = \( I(P,N) - E(\text{Income}) \)

\[ = 0.940 - 0.9131 = 0.027 \]

2. Index 2

<table>
<thead>
<tr>
<th>Credit</th>
<th>( P_i )</th>
<th>( N_i )</th>
<th>( I(P_i,N_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fair</td>
<td>2</td>
<td>1</td>
<td>0.913</td>
</tr>
<tr>
<td>Exc</td>
<td>2</td>
<td>1</td>
<td>0.913</td>
</tr>
</tbody>
</table>

\[ I(P,N) = I(4,2) = 0.9234 \text{ (Using equation 1)} \]

\[ E(\text{Credit}) = \frac{3}{6} I(2,1) + \frac{3}{6} I(2,1) ------(3) \]

\[ I(2,1) = 0.913 \text{ (Using equation 1)} \]

\[ E(\text{Credit}) = 0.913 \text{ (Substituting value of } I(2,1) \text{ in (3)} \]

Gain(Credit) = \( I(P,N) - E(\text{Credit}) = 0.9234 - 0.913 \)

\[ = 0.01 \]

Similarly we can calculate Information gain of tables at each stage.
EXERCISE:
Construct a correct tree of your choice of attributes and evaluate:
1. correctness of your rules, i.e. the predictive accuracy with respect to the TRAINING data
2. predictive accuracy with respect to test data from Exercise 2

- Remember
- The TERMINATION CONDITIONS!