SUPERVISED LEARNING

ROAD MAP

- **Basic concepts**
- Decision tree induction
- Evaluation of classifiers
- Rule induction
- Classification using association rules
- K-nearest neighbor

AN EXAMPLE APPLICATION

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - Marital status
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: to decide whether an application should be approved, or to classify applications into two categories, approved and not approved.

AN EXAMPLE APPLICATION

- An emergency room in a hospital measures 15 variables (e.g., blood pressure, age, heart rate, etc) of newly admitted patients.
- A decision is needed: whether to send a new patient to an intensive-care unit based on the mortality risk.
- Problem: to predict high-risk patients and distinguish them from low-risk patients.

MACHINE LEARNING AND OUR FOCUS

- A computer system learns from data
- Our focus:
  - learn a target function
  - Use the learned function to predict the values of a discrete class attribute
    - e.g., approve or not-approved, and high-risk or low risk.
- The task is commonly called: Supervised learning, classification, or inductive learning.
  - Classification (discrete); Regression (numeric; continuous)

THE DATA AND THE GOAL

- Data: A set of data examples / instances / cases described by
  - k attributes: $A_1, A_2, \ldots A_k$.
  - a class: Each example is labelled with a pre-defined class.
    - e.g., approved or not approved
- Goal:
  - learn a classification model from the data
  - Use the model to predict the classes of new instances.

AN EXAMPLE: DATA (LOAN APPLICATION)

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Approved or not

AN EXAMPLE: DATA (LOAN APPLICATION)

- Learn a classification model from the data
- Use the model to classify future loan applications into
  - Yes (approved) and
  - No (not approved)
- What is the class for following case/instance?

Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

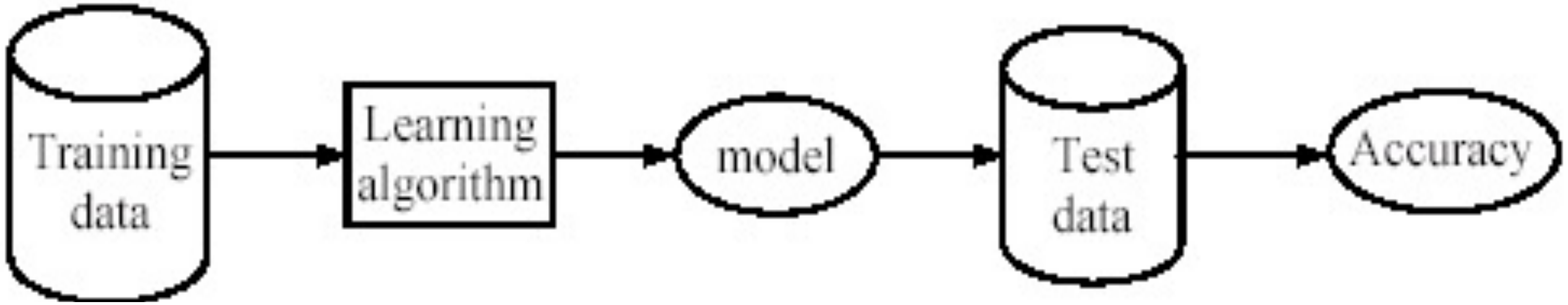
SUPERVISED VS. UNSUPERVISED LEARNING

- Supervised learning:
- Supervision: data are labeled with pre-defined classes.
  - Predict the test data into the classes.
- Unsupervised learning (clustering)
  - Class labels of the data are unknown
  - Given a set of data, the task is to establish the existence of classes or clusters in the data

SUPERVISED LEARNING PROCESS: TWO STEPS

- Learning (training): learn a model via the training data
- Testing: test the model via test data and evaluate the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



AN EXAMPLE

- Data: loan application data
- Task: predict whether a loan should be approved or not.
- Performance measure: accuracy

No learning: put all test data to the majority class (i.e., Yes):

Accuracy = 8/15 = 53%

- With the learned model, we can do better than 53%.

FUNDAMENTAL ASSUMPTION OF LEARNING

Assumption: the distribution of training data is identical to the distribution of test data.

- To achieve good accuracy on the test data, training data must be sufficiently large.

ROAD MAP

- Basic concepts
- **Decision tree induction**
- Evaluation of classifiers
- Rule induction
- Classification using association rules
- K-nearest neighbor

INTRODUCTION

- Decision tree learning is one of the most widely used techniques for classification.
  - its accuracy is competitive with other methods
  - it is efficient
- The classification model is a tree, called decision tree.
- C4.5 is widely used decision tree.
- (use python and weka to train and test machine learning models).

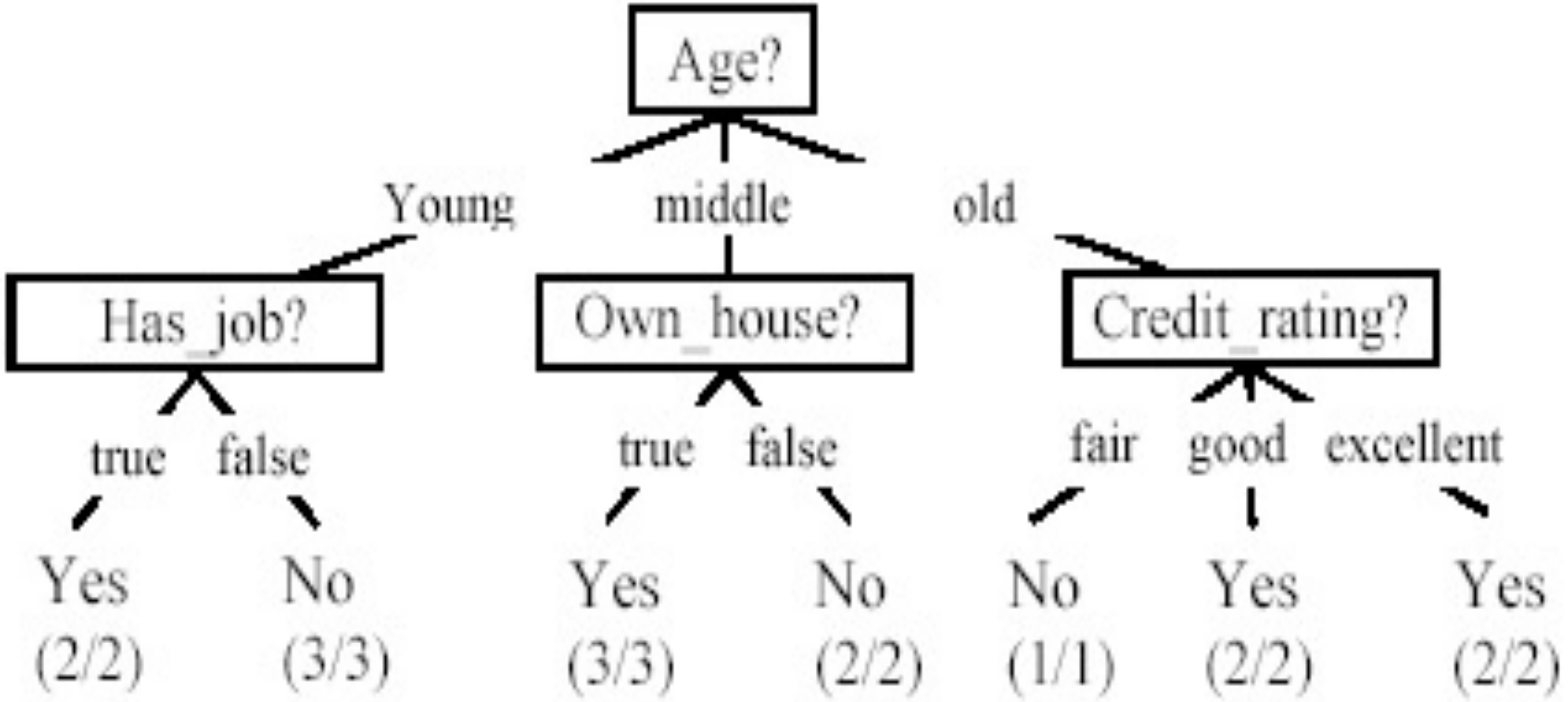
THE LOAN DATA (REPRODUCED)

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Approved or not

A DECISION TREE FROM THE LOAN DATA

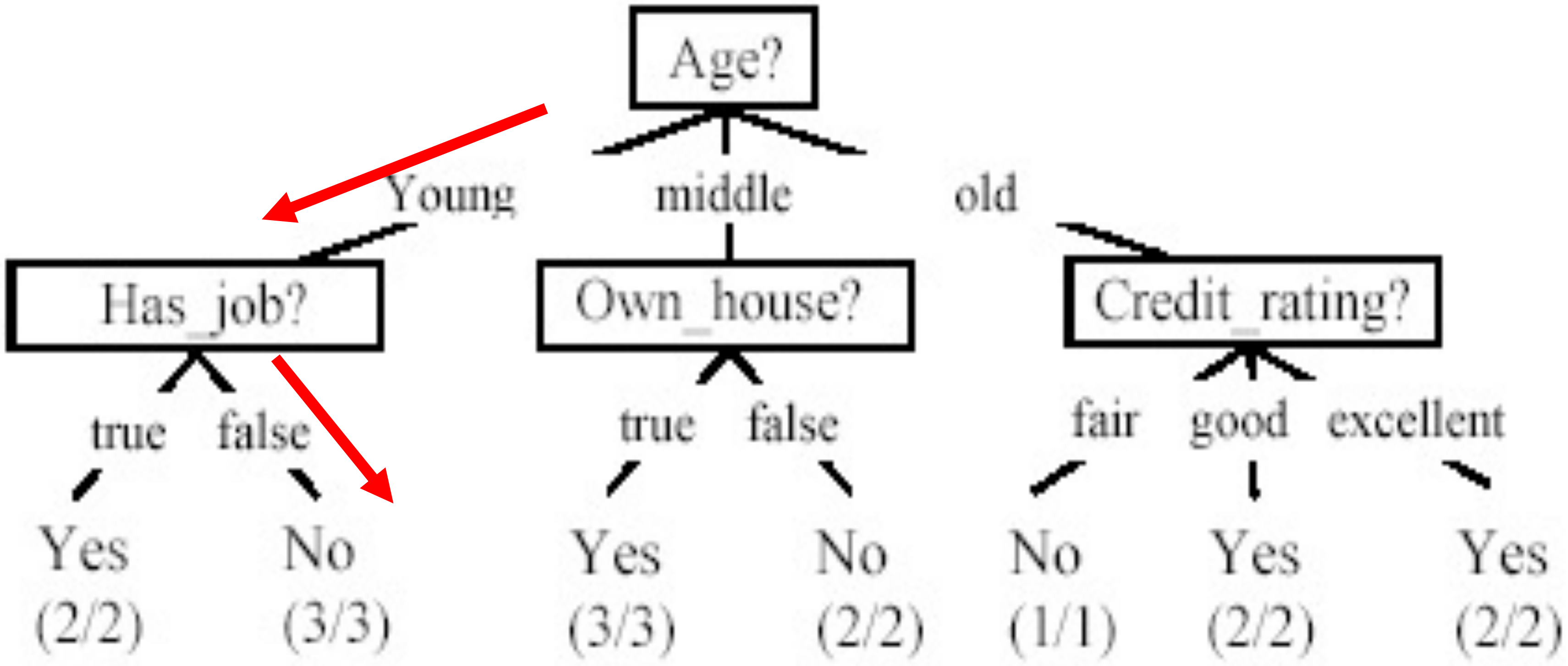
- Decision nodes and leaf nodes (classes)



USE THE DECISION TREE

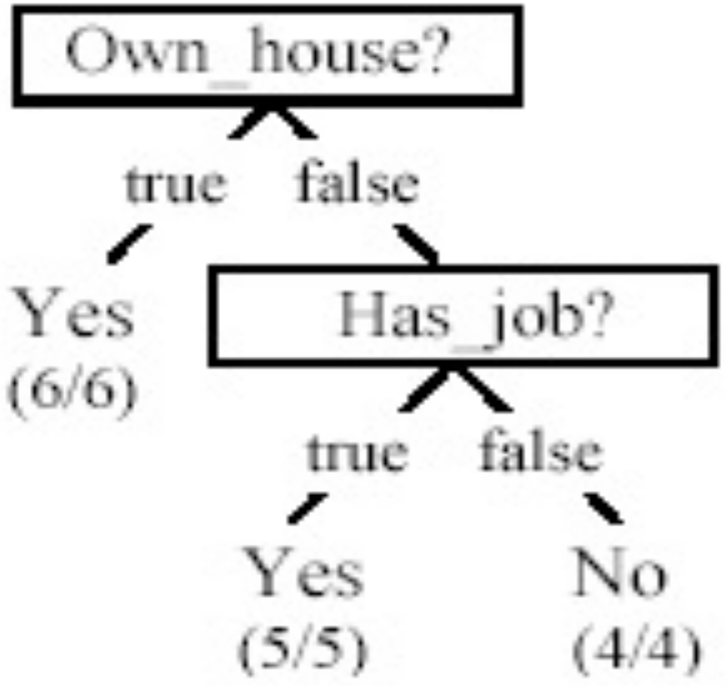
Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

No



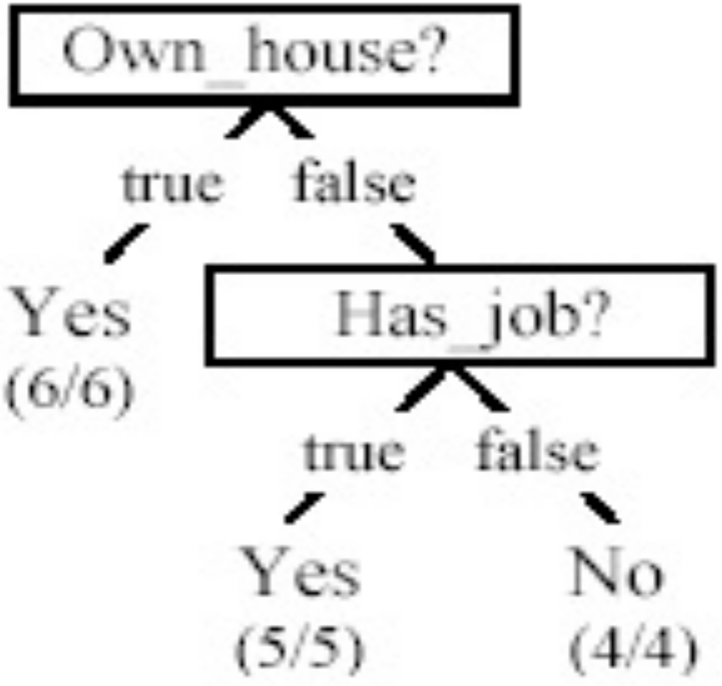
IS THE DECISION TREE UNIQUE?

- No. There could be many trees.
- We want smaller (easy to understand) and accurate tree (good performance).



FROM A DECISION TREE TO A SET OF RULES

- A decision tree can be converted to a set of rules
- Each path from the root to a leaf is a rule.



Own_house = true → Class =Yes [sup=6/15, conf=6/6]
Own_house = false, Has_job = true → Class = Yes [sup=5/15, conf=5/5]
Own_house = false, Has_job = false → Class = No [sup=4/15, conf=4/4]

ALGORITHM FOR DECISION TREE LEARNING

- Basic algorithm (greedy **divide-and-conquer**)
  - given categorical attributes/features
  - tree is constructed in a top-down recursive manner
  - at start, all the training examples are at the root
  - examples are partitioned recursively based on selected attributes
  - attributes are selected based on information gain

ALGORITHM FOR DECISION TREE LEARNING

- When to stop partitioning
  - All examples for a given node belong to the same class
  - There are no remaining attributes for further partitioning
  - There are no examples left

CHOOSE AN ATTRIBUTE TO PARTITION DATA

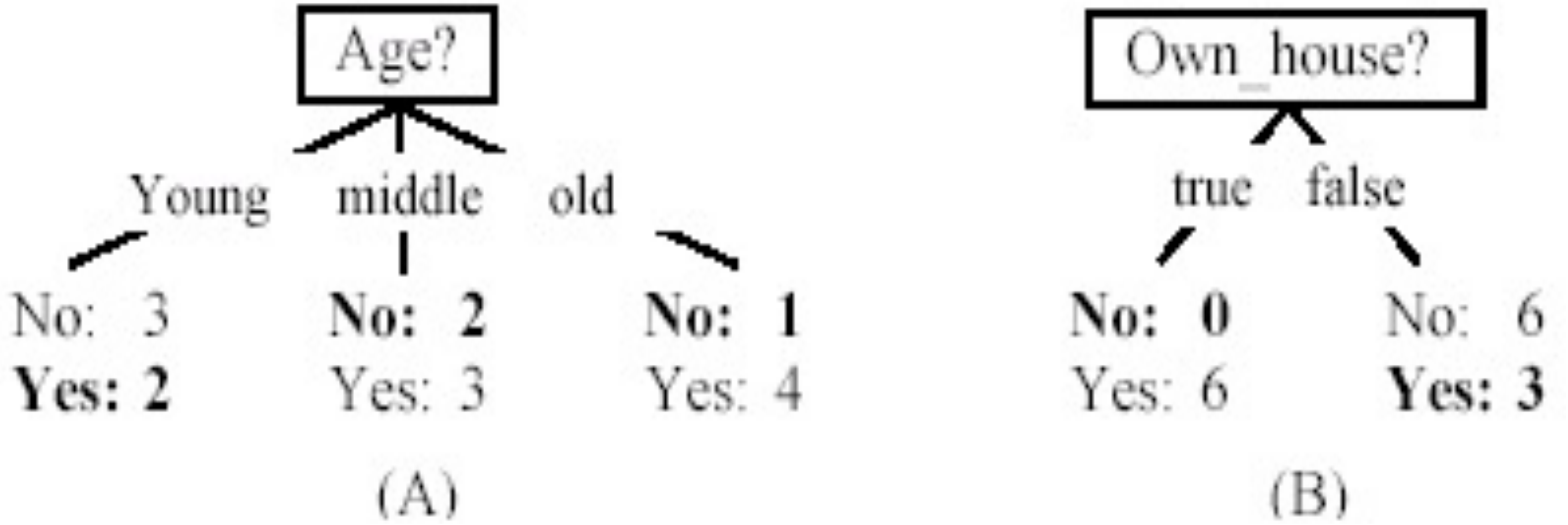
- the *key* to building a decision tree - choose attribute.
- the objective is to reduce impurity in data.
  - A subset of data is pure if all instances belong to the same class.
- The *heuristic* in C4.5 is to choose the attribute with the maximum Information Gain.

THE LOAN DATA (REPRODUCED)

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Approved or not

TWO POSSIBLE ROOTS, WHICH IS BETTER?



- Fig. (B) seems to be better.

- The entropy formula,

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j) \log_2 \Pr(c_j)$$

$$\sum_{j=1}^{|C|} \Pr(c_j) = 1,$$

- $\Pr(c_j)$ is the probability of class c_j in data set D
- We use entropy as a measure of impurity or disorder of data set D. (Or, a measure of information in a tree)

ENTROPY MEASURE: LET US GET A FEELING

1. The data set D has 50% positive examples ($\Pr(positive) = 0.5$) and 50% negative examples ($\Pr(negative) = 0.5$).

$$entropy(D) = -0.5 \times \log_2 0.5 - 0.5 \times \log_2 0.5 = 1$$

2. The data set D has 20% positive examples ($\Pr(positive) = 0.2$) and 80% negative examples ($\Pr(negative) = 0.8$).

$$entropy(D) = -0.2 \times \log_2 0.2 - 0.8 \times \log_2 0.8 = 0.722$$

3. The data set D has 100% positive examples ($\Pr(positive) = 1$) and no negative examples, ($\Pr(negative) = 0$).

$$entropy(D) = -1 \times \log_2 1 - 0 \times \log_2 0 = 0$$

- As the data become purer and purer, the entropy value becomes smaller and smaller. This is useful to us!

ENTROPY MEASURE: LET US GET A FEELING

- Given a set of examples *D*, we first compute its entropy:

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j)\log_2 \Pr(c_j)$$

- If we make attribute A_i, with v values, the root of the current tree, this will partition *D* into v subsets $D_1, D_2 \ldots, D_v$. The expected entropy if A_i is used as the current root:

$$entropy_{A_i}(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times entropy(D_j)$$

ENTROPY MEASURE: LET US GET A FEELING

- Information gained by selecting attribute A_i to branch or to partition the data is

$$gain(D, A_i) = entropy(D) - entropy_{A_i}(D)$$

- We choose the attribute with the highest gain to branch/split the current tree.

$$entropy(D) = \frac{6}{15}\times\log_2\frac{6}{15} + \frac{9}{15}\times\log_2\frac{9}{15} = 0.971$$

$$\begin{aligned} entropy_{Own\_house}(D) &= \frac{6}{15}\times entropy(D_1) + \frac{9}{15}\times entropy(D_2) \\ &= \frac{6}{15}\times 0 + \frac{9}{15}\times 0.918 \\ &= 0.551 \end{aligned}$$

$$\begin{aligned} entropy_{Age}(D) &= \frac{5}{15}\times entropy(D_1) + \frac{5}{15}\times entropy(D_2) + \frac{5}{15}\times entropy(D_3) \\ &= \frac{5}{15}\times 0.971 + \frac{5}{15}\times 0.971 + \frac{5}{15}\times 0.722 \\ &= 0.888 \end{aligned}$$

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	excellent	No
3	young	true	false	good	Yes
4	young	true	true	good	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

Age	Yes	No	entropy(Di)
young	2	3	0.971
middle	3	2	0.971
old	4	1	0.722

- Own_house is the best choice for the root.

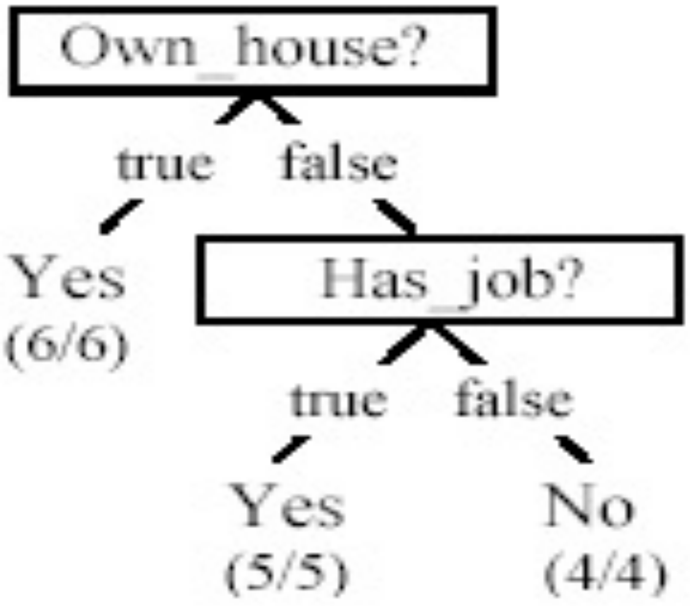
$gain(D, \text{Age}) = 0.971 - 0.888 = 0.083$

$gain(D, \text{Own\_house}) = 0.971 - 0.551 = 0.420$

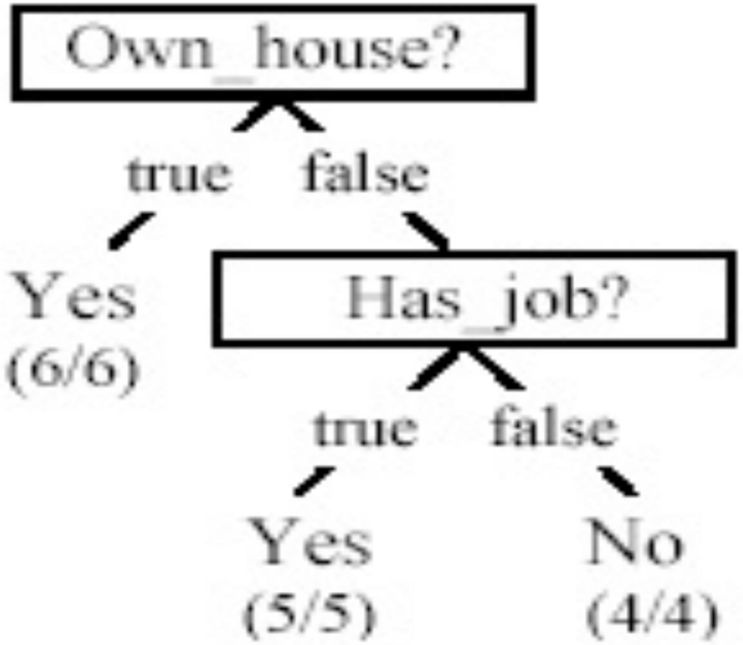
$gain(D, \text{Has\_Job}) = 0.971 - 0.647 = 0.324$

$gain(D, \text{Credit\_Rating}) = 0.971 - 0.608 = 0.363$

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	**No**
2	young	false	false	excellent	**No**
3	young	true	false	good	**Yes**
4	[redacted]				
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	[redacted]				
9					
10					
11					
12					
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**



WE BUILD THE FINAL TREE



- We can use information gain ratio to evaluate the impurity as well (see the handout)