CLASSIFICATION

ROAD MAP

- **Basic concepts and Decision Trees**
- Inferring rudimentary rules
- Covering rules
- Experiments with Weka

EXAMPLE

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - has_job
  - own_house
  - credit rating
  - etc.

// label

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

- 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.

CLASSIFICATION

- Definition:
- Given a collection of records (training set) historical Data
- Each record is by characterized by a tuple (*x*,*y*), where *x* is the attribute set and *y* is the class label
  - *x*: attribute, predictor, independent variable, input
  - *y*: class, response, dependent variable, output

$\vec{x} = (x_1, x_2, x_3, x_4)$

Features / variables / attributes

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

Label
= class
= category

Each row = a datapoint

- 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.

training set

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**

D_{Train}

→ model / function —predict→

test set

Age	Has_Job	Own_house	Credit-Rating
young	false	false	good

D_{test}

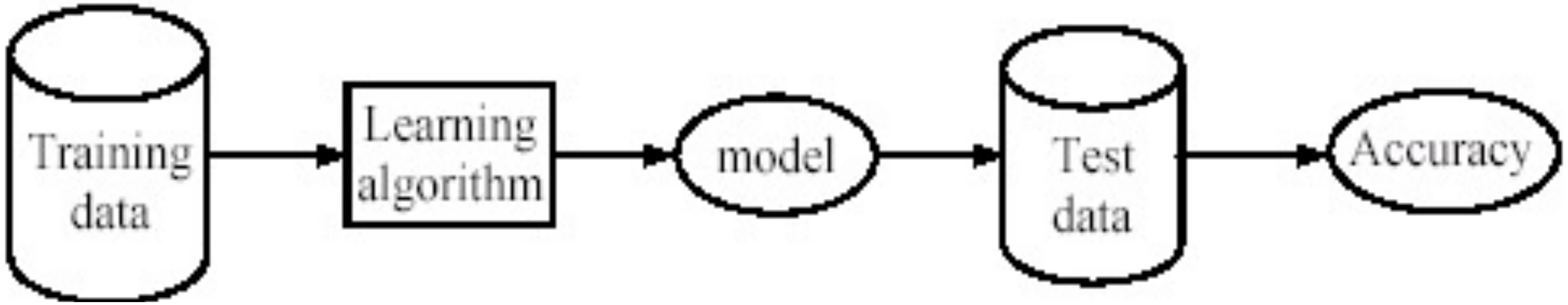
In real life, it may not follow i.i.d assumption of the supervised learning (classification problem).

Accuracy of correctly classify a datapoint = 8/10 = 80%

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

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



T = 100 < 99 A, 1 NA

⇒ 98% A

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% $\Leftarrow$ NOT Good Model

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

FUNDAMENTAL ASSUMPTION OF LEARNING

Classification (supervised learning)

i.i.d = identical independent

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

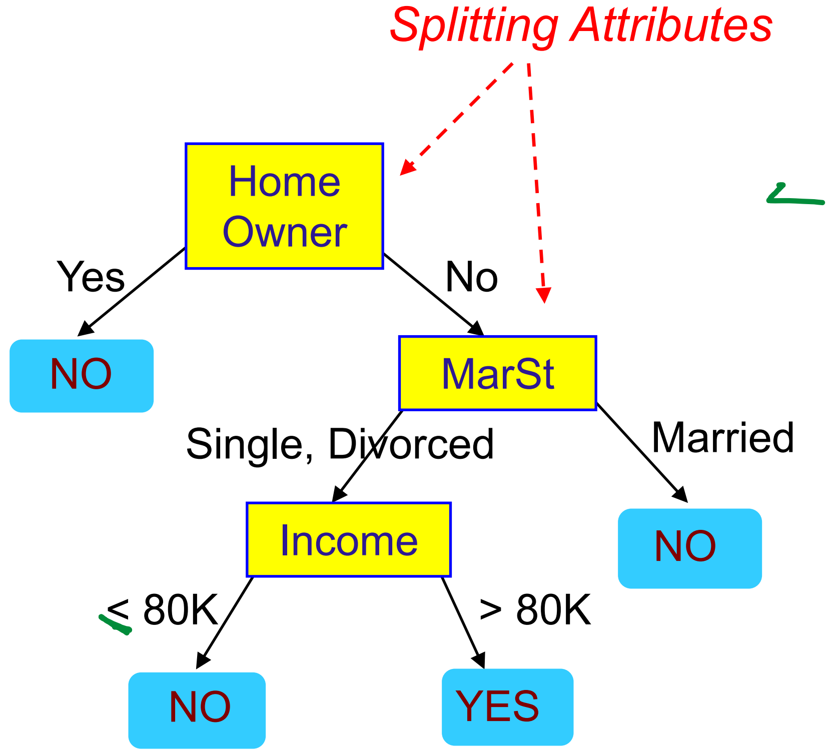
- **Basic concepts and Decision Trees**
- Inferring rudimentary rules
- Covering rules
- Experiments with Weka

EXAMPLE OF A DECISION TREE

categorical categorical continuous class

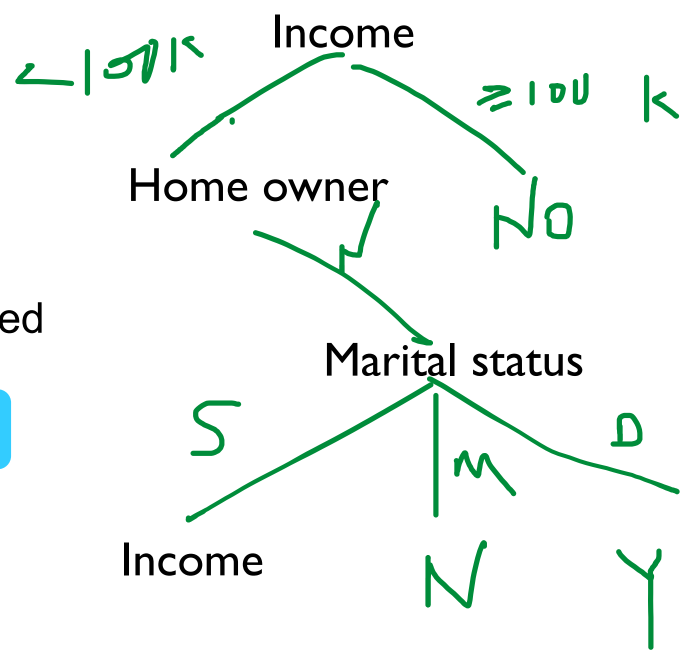
ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: Decision Tree

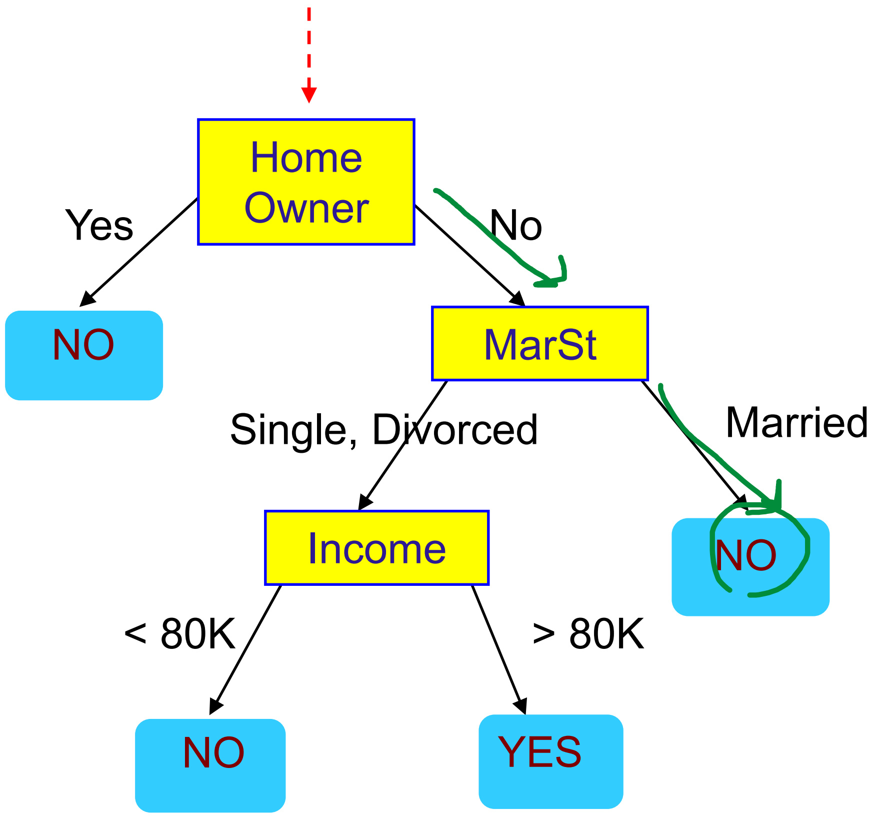
1DT



2nd DT

APPLY MODEL TO TEST DATA

Start from the root of tree.



Test Data

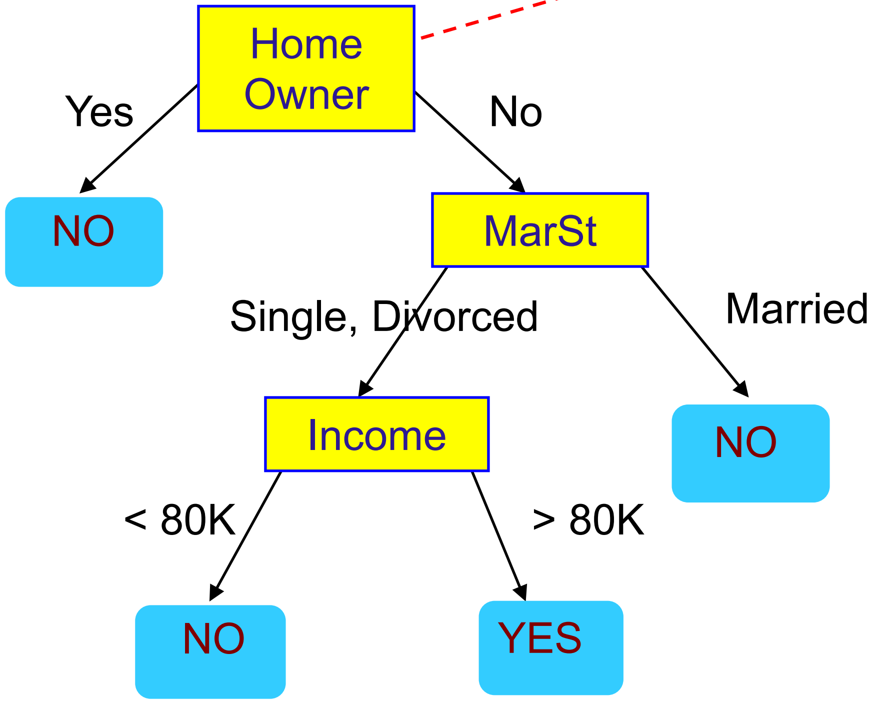
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

NO

APPLY MODEL TO TEST DATA

Test Data

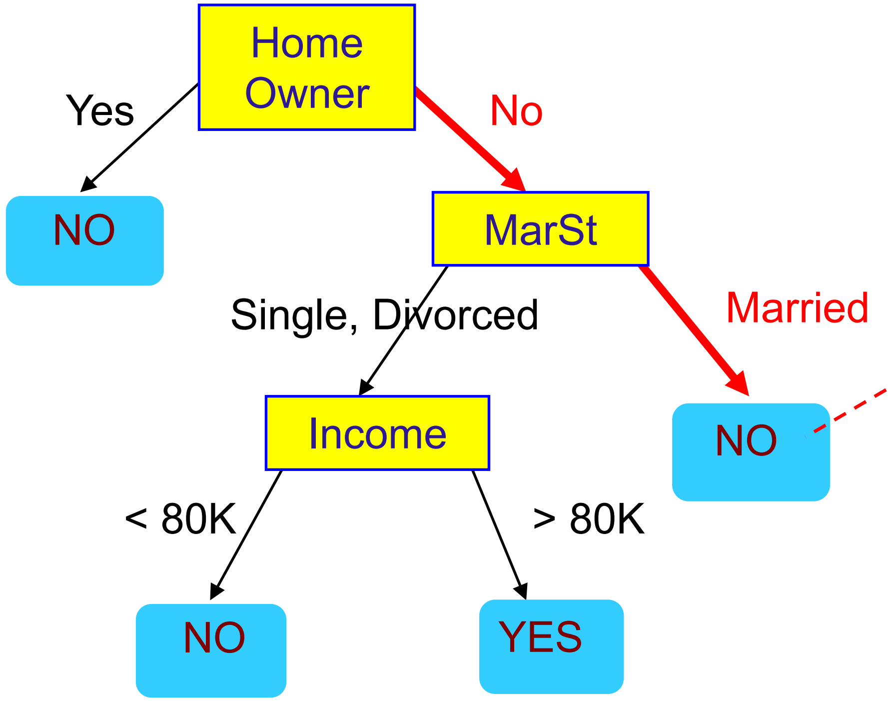
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



APPLY MODEL TO TEST DATA

Test Data

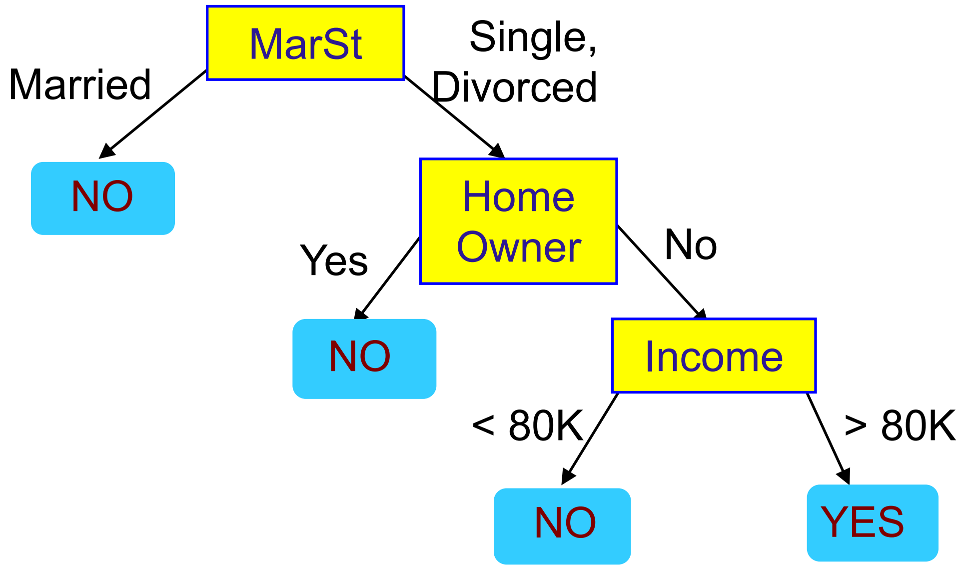
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Assign Defaulted to "No"

ANOTHER EXAMPLE OF DECISION TREE

ID	Home Owner (categorical)	Marital Status (categorical)	Annual Income (continuous)	Defaulted Borrower (class)
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



There could be more than one tree that fits the same data!

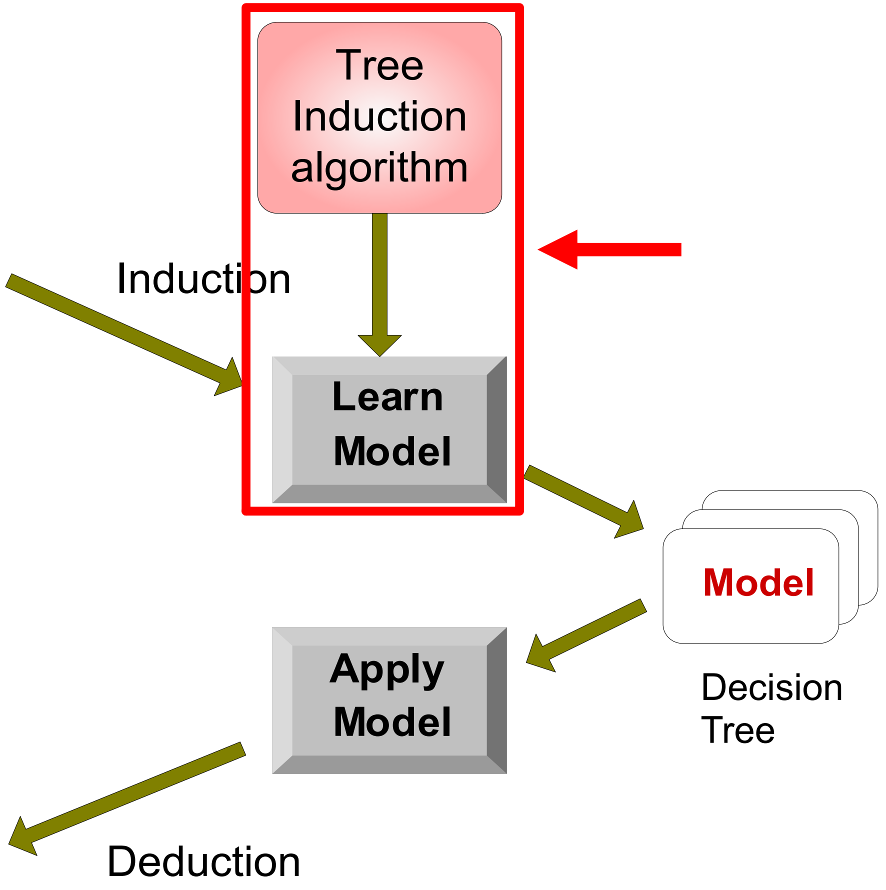
Complicate

Simple

DECISION TREE CLASSIFICATION TASK

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

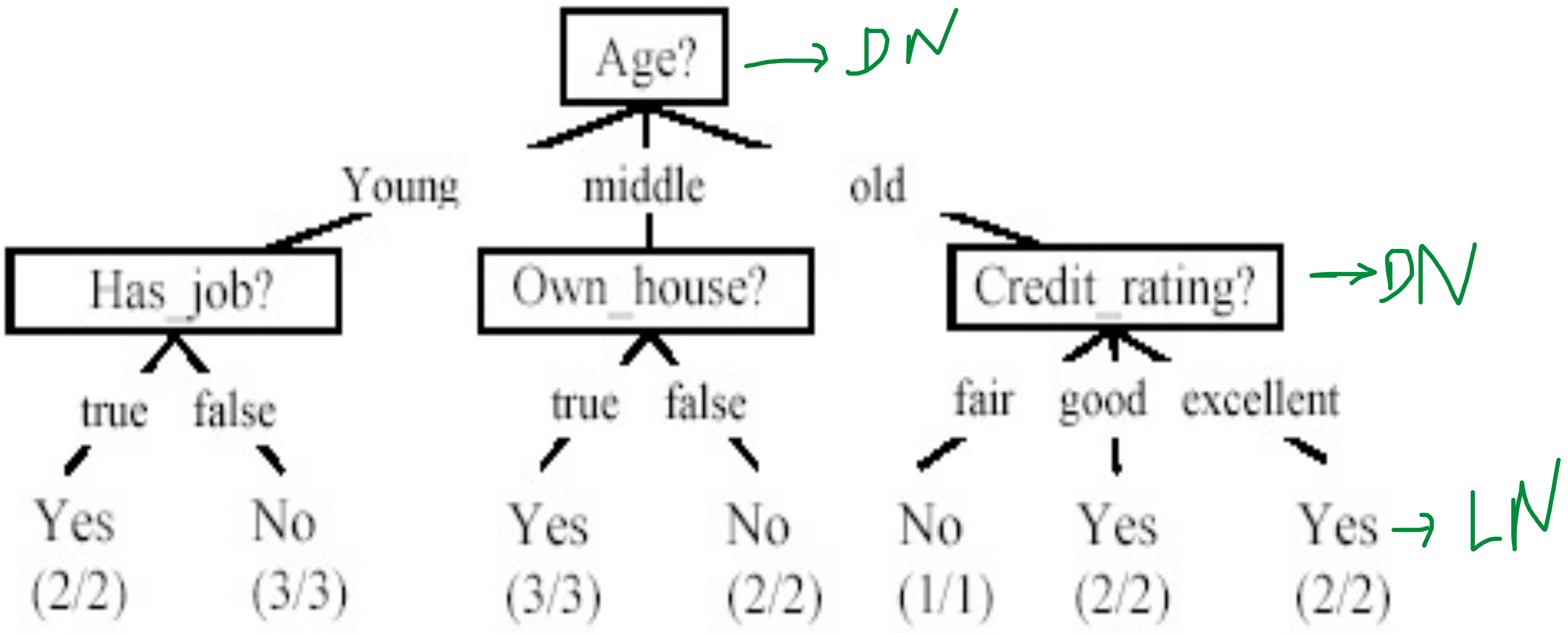
DECISION TREE INDUCTION

- Many Algorithms:
  - Hunt's Algorithm (one of the earliest)
  - CART
  - ID3, C4.5
  - SLIQ,SPRINT

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

A DECISION TREE FROM THE LOAN DATA

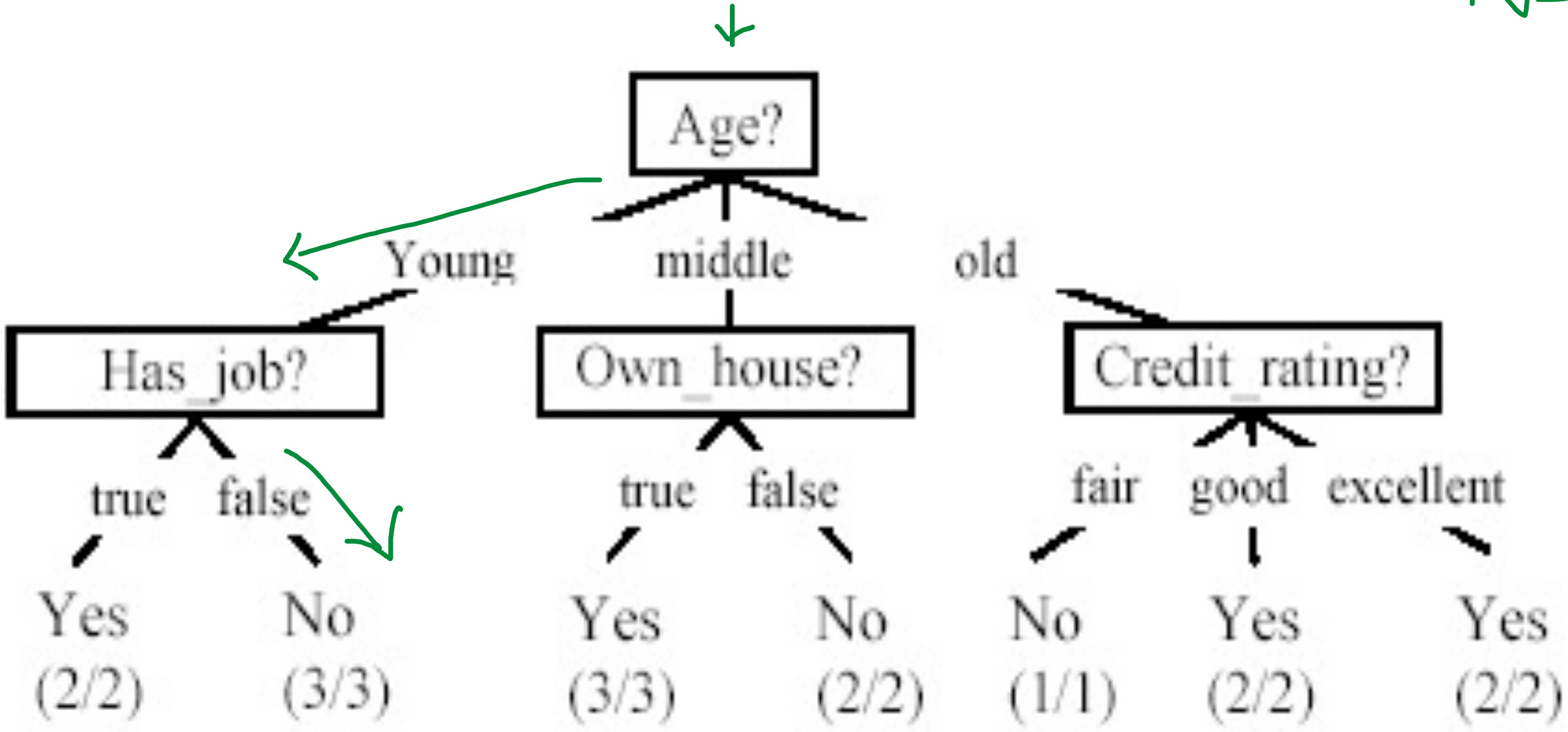
- Decision nodes and leaf nodes (classes)



USE THE DECISION TREE

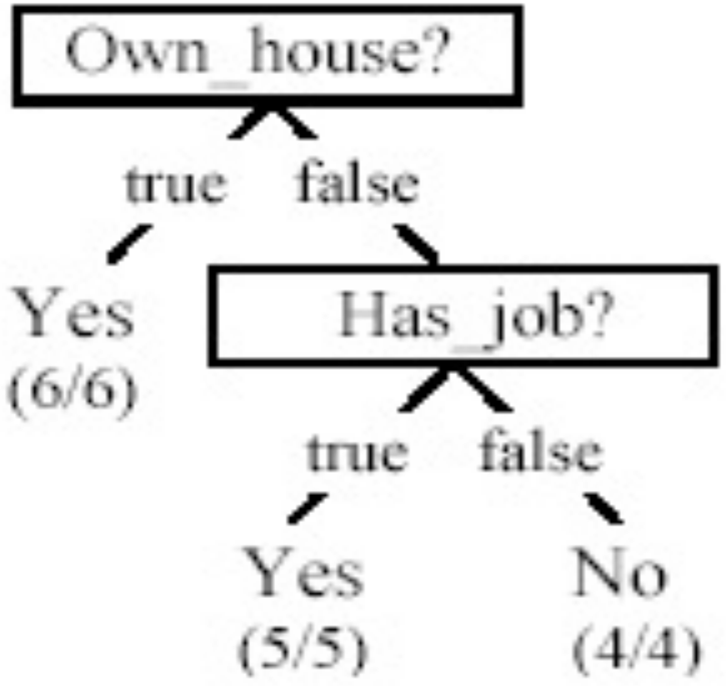
Age	Has_Job	Own_house	Credit-Rating	Class
young	false	false	good	?

NO



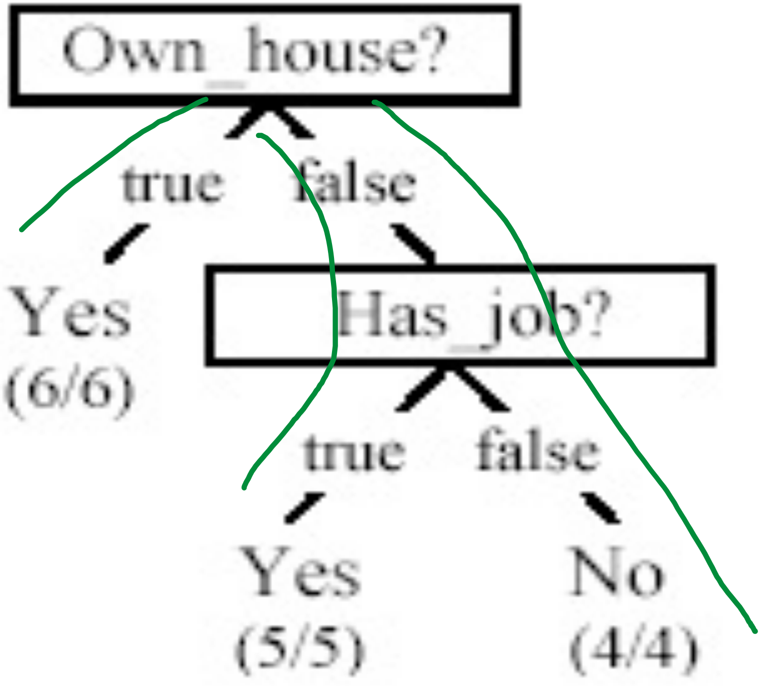
IS THE DECISION TREE UNIQUE?

- 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 (if condition)
- Each path from the root to a leaf is a rule.



3 rules

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