Course: Analytics, Machine Learning, and the Digital Economy

Introduction to neural networks

Lecturer Radjabova Dilnora

Supervised Learning

- Learning a discrete function: **Classification**
  - Boolean classification:
    - Each example is classified as true(positive) or false(negative).

Supervised Learning

- Learning a continuous function: **Regression**

Classification—A Two-Step Process

- Model construction: describing a set of predetermined classes
  - Each tuple/sample is assumed to belong to a predefined class, as determined by the class label
  - The set of tuples used for model construction is training set
  - The model is represented as **classification rules**, **decision trees**, or **mathematical formulae**

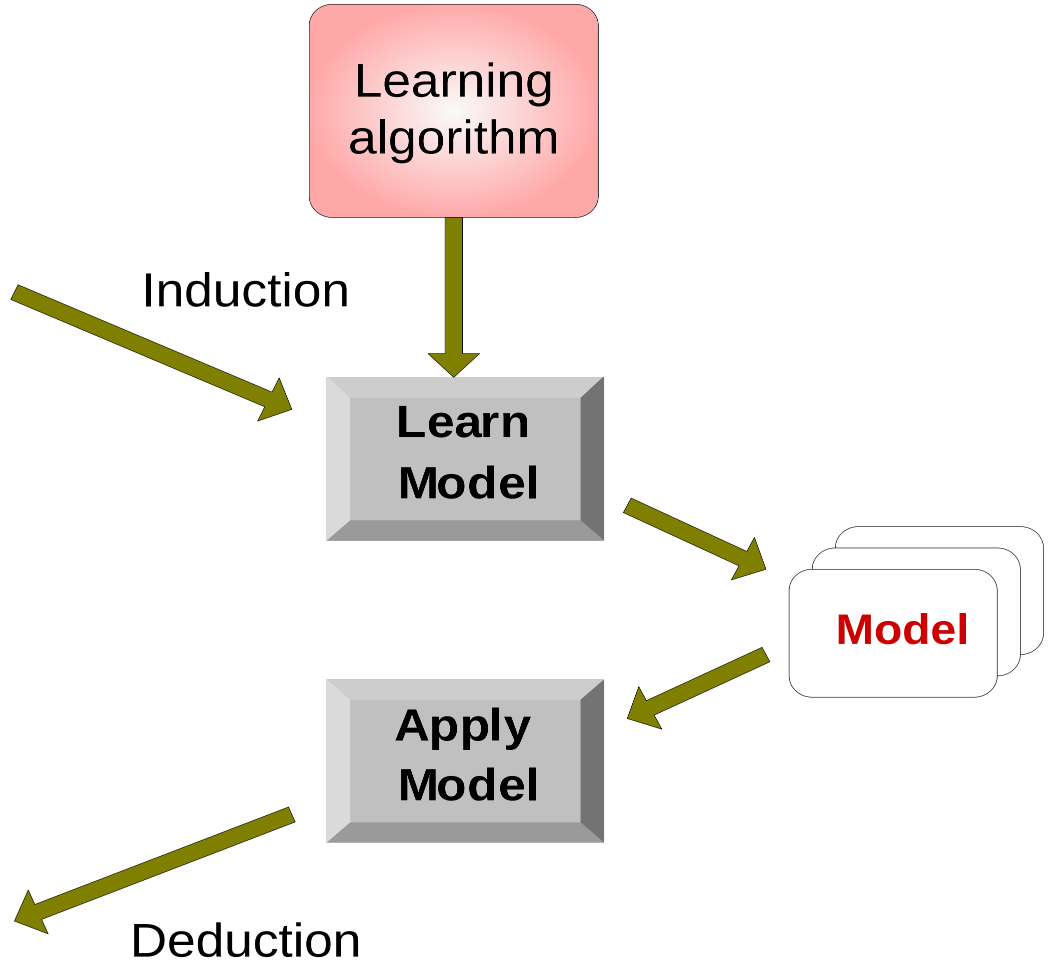
Classification—A Two-Step Process

- Model usage: for classifying future or unknown objects
  - Estimate accuracy of the model
    - The known label of test sample is compared with the classified result from the model
    - **Test set is independent of training set**, otherwise over-fitting will occur
  - If the accuracy is acceptable, use the model to classify data tuples whose class labels are not known

Illustrating 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

Issues: Data Preparation

- Data cleaning
  - Preprocess data in order to reduce noise and handle missing values

Issues: Data Preparation

- Relevance analysis (feature selection)
  - Remove the irrelevant or redundant attributes

Issues: Data Preparation

- Data transformation
  - Generalize data to (higher concepts, discretization)
  - Normalize attribute values

Classification Techniques

- Decision Tree based Methods
- Rule-based Methods
- Naïve Bayes and Bayesian Belief Networks
- Neural Networks
- Support Vector Machines
- and more...

Learning decision trees

Example Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. **Alternate**: is there an alternative restaurant nearby?
2. **Bar**: is there a comfortable bar area to wait in?

Learning decision trees

Example Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. **Fri/Sat**: is today Friday or Saturday?
2. **Hungry**: are we hungry?

Learning decision trees

Example Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. **Patrons**: number of people in the restaurant (None, Some, Full)
2. **Price**: price range ($, $$, $$$)
3. **Raining**: is it raining outside?

Learning decision trees

Example Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. **Reservation**: have we made a reservation?
2. **Type**: kind of restaurant (French, Italian, Thai, Burger)
3. **WaitEstimate**: estimated waiting time (0-10, 10-30, 30-60, >60)

Feature(Attribute)-based representations

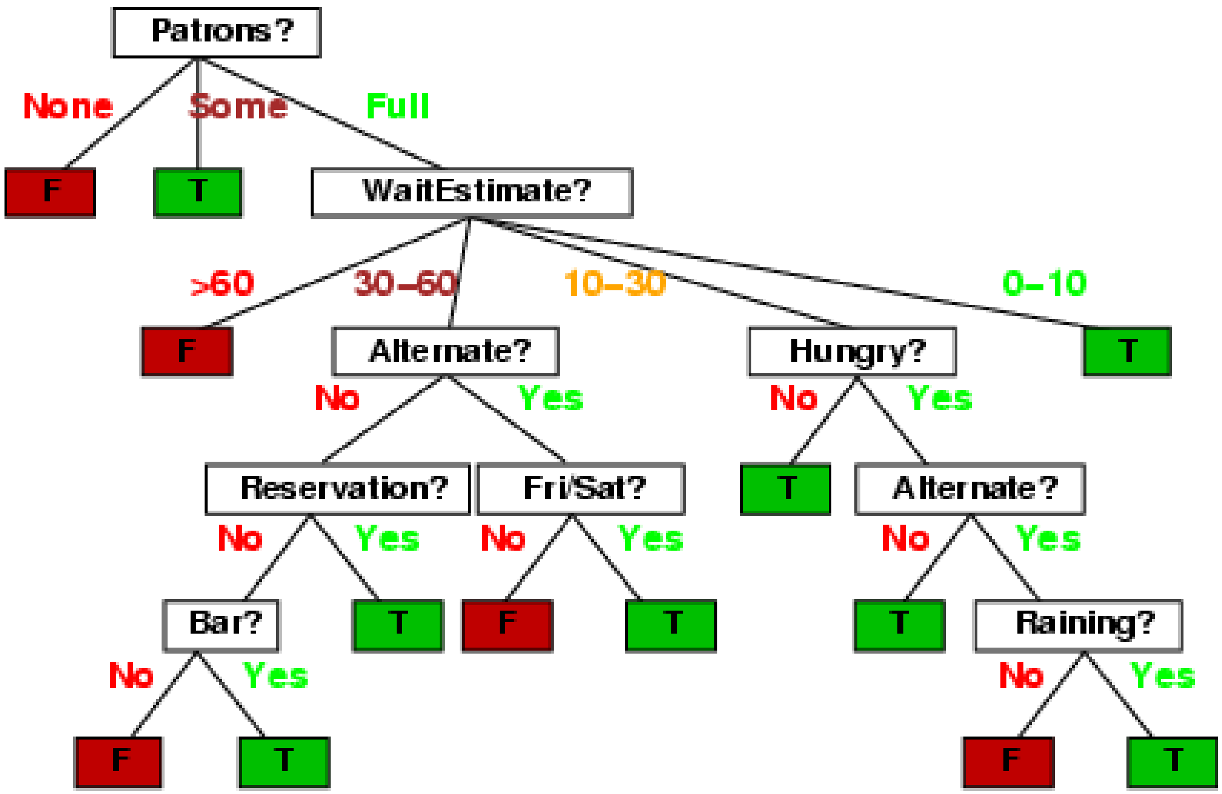
- Examples described by feature(attribute) values
  - (Boolean, discrete, continuous)
- E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

- Classification of examples is positive (T) or negative (F)
-

Decision trees

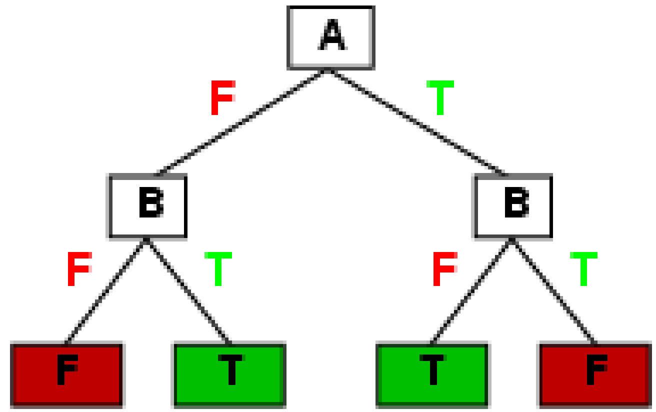
- One possible representation for hypotheses
- E.g., here is the “true” tree for deciding whether to wait:



Expressiveness

- Decision trees can express any function of the input attributes.
- E.g., for Boolean functions, truth table row → path to leaf:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



- Trivially, there is a consistent decision tree for any training set with one path to leaf for each example (unless *f* nondeterministic in *x*) but it probably won't generalize to new examples

- Prefer to find more compact decision trees

Decision tree learning

- Aim: find a small tree consistent with the training examples
- Idea: (recursively) choose "most significant" attribute as root of (sub)tree

function DTL(*examples*, *attributes*, *default*) **returns** a decision tree

  if *examples* is empty **then return** *default*
  else if all *examples* have the same classification **then return** the classification
  else if *attributes* is empty **then return** MODE(*examples*)
  else
    best ← CHOOSE-ATTRIBUTE(*attributes*, *examples*)
    tree ← a new decision tree with root test *best*
    for each value v_i of *best* **do**
      $examples_i \leftarrow$ {elements of *examples* with $best = v_i$}
      $subtree \leftarrow \mathrm{DTL}(examples_i, attributes - best, \textsc{Mode}(examples))$
      add a branch to *tree* with label v_i and subtree *subtree*
    return *tree*

Decision Tree Construction Algorithm

- **Principle**
  - Basic algorithm (adopted by ID3, C4.5 and CART): a **greedy algorithm**
  - Tree is constructed in a *top-down recursive divide-and-conquer* manner

Decision Tree Construction Algorithm

- **Iterations**
  - At start, all the training tuples are at the root
  - Tuples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g, information gain)

Decision Tree Construction Algorithm

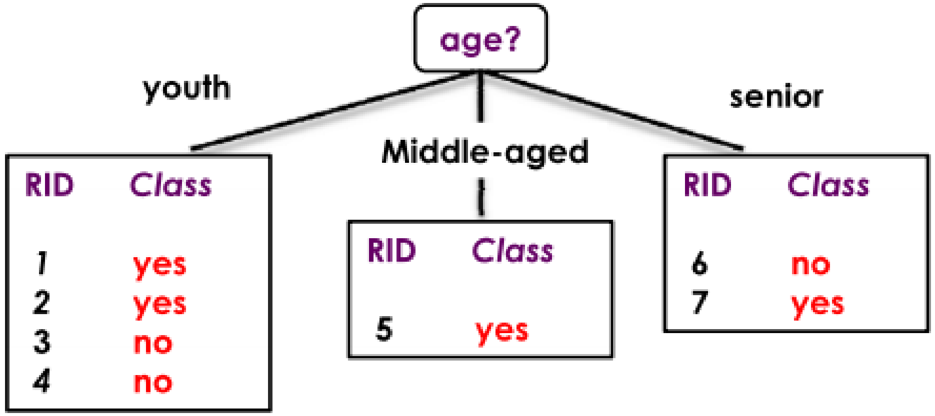
- **Stopping conditions**
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – **majority voting** is employed for classifying the leaf
  - There are no samples left

Decision Tree Induction: Training Dataset

This follows an example of Quinlan's ID3 (Playing Tennis)

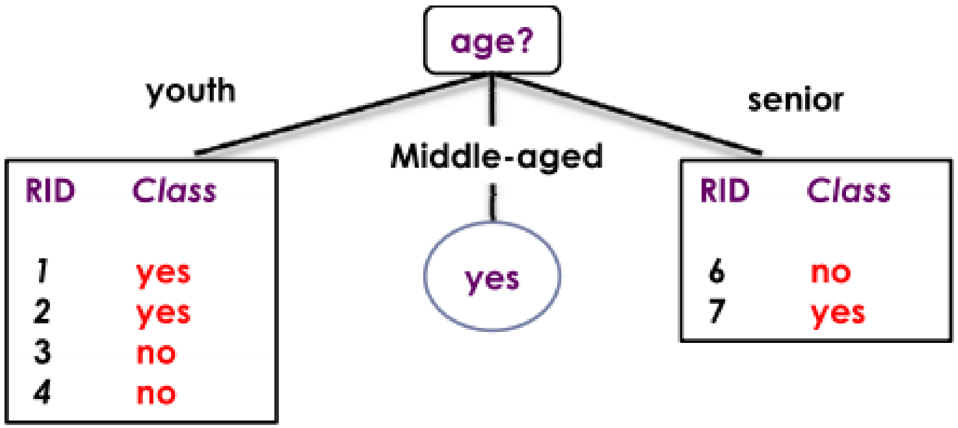
age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Example



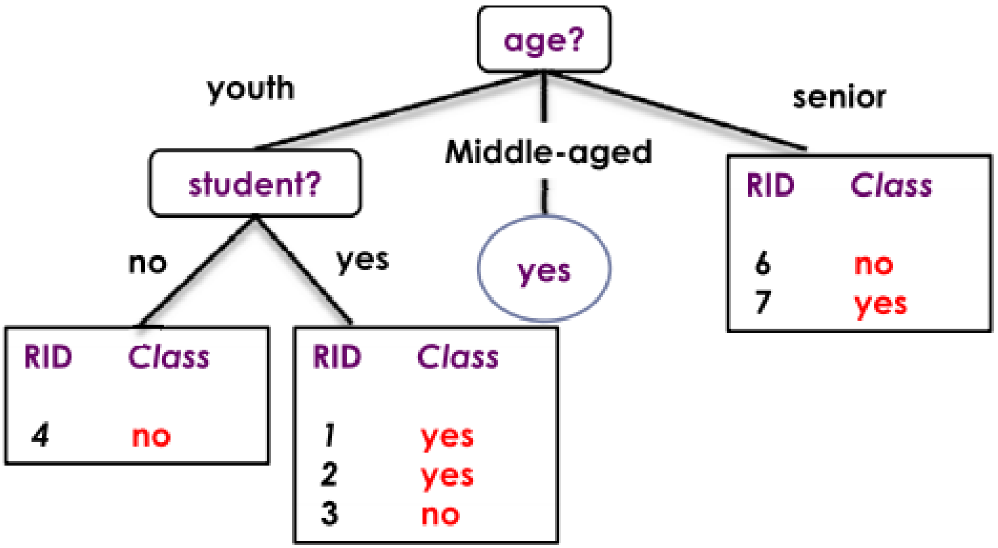
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes

Example



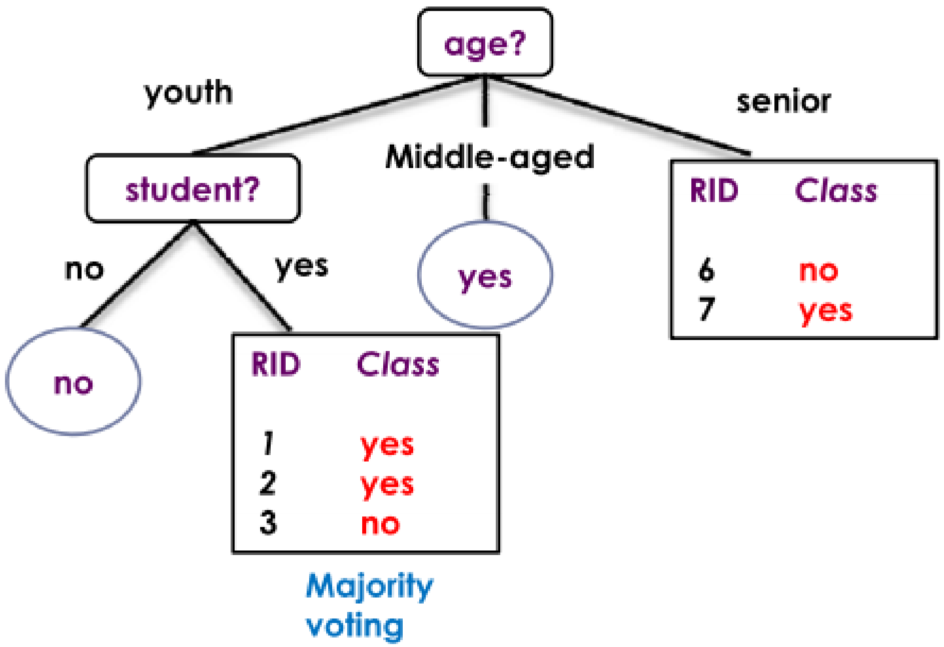
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes

Example



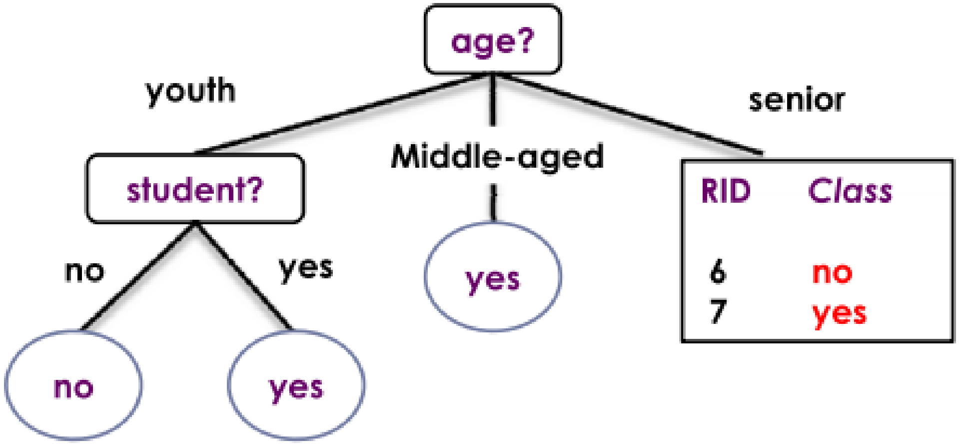
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes

Example



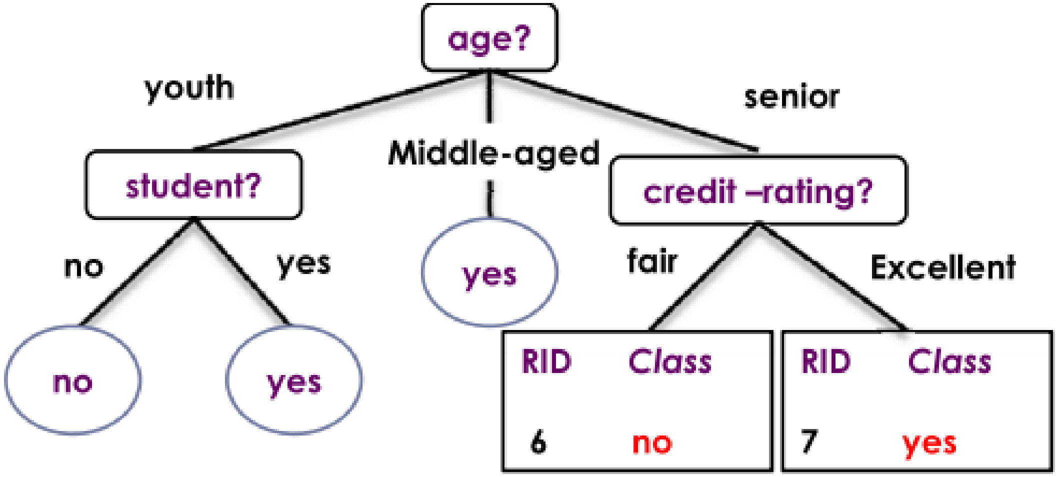
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes

Example



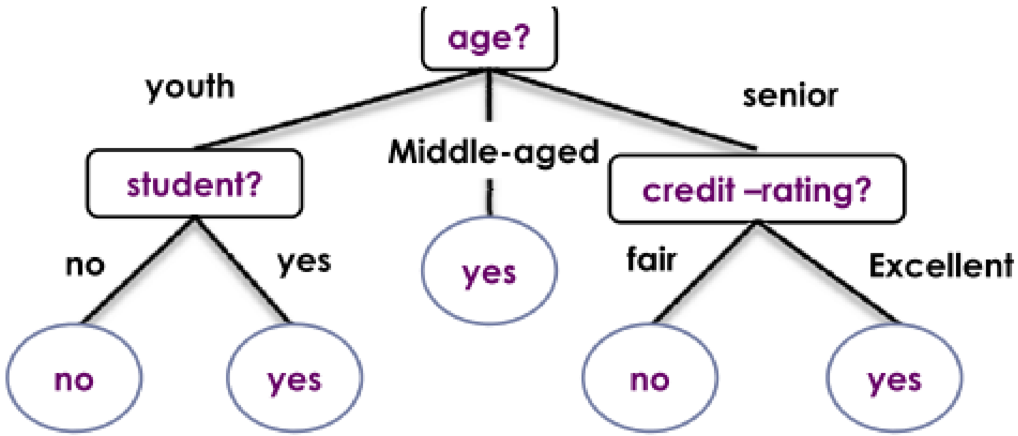
RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes

Example



RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	no
7	senior	yes	excellent	yes

Example



RID	age	student	credit-rating	Class: buys_computer
1	youth	yes	fair	yes
2	youth	yes	fair	yes
3	youth	yes	fair	no
4	youth	no	fair	no
5	middle-aged	no	excellent	yes
6	senior	yes	fair	yes
7	senior	yes	excellent	no

Reference

- 1. Shalev-Shwartz and Ben-David. Understanding Machine Learning: From Theory to Algorithms (Cambridge University Press, 2014)
- 2. Daum´e. A Course in Machine Learning.
- 3. The Art of Statistics: How to Learn from Data by David Shpigelter
- 4. Learning From Data – January 1, 2012 by Yaser S. Abu-Mostafa (Author), Malik Magdon-Ismail (Author), Hsuan-Tien Lin (Author)
- 5. Statistics: The Art and Science of Learning from Data by Alan Agresti
- 6. Learning From Data: An Introduction To Statistical Reasoning by M.Glenber.
- 7. Statistics: Learning from Data (with JMP Printed Access Card) by Rocky Pek
- 8. The Elements of Statistical Learning by Gerim Garold
- 9. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems 2nd Edition
- by Aurélien Géron (Author)