CSCI 5408 Data Analytics: DM and DW Tech
(Mar 22, Week 11)

- Ass5 Due: Mar 28
  - Read Ass5-Tutorial slides
  - Help Hours: Fri, 1:00-2:30 PM, CS 233
- Final Exam: Apr 20, 3:30-5:30 PM
- Write answers for review questions
- Reading: Lecture 18; Text 3rd: 8.1-8.3, or 2nd: 6.1, 6.2-4, 6.6, 6.16

Part II Outline

Overview: (Week 8)

1. Introduction: Overview on DM&DW ***Ass4:*** *ETL/DW/OLAP*

2. Data preprocessing

DW & OLAP: (Week 9)

3. Data warehousing and OLAP

Basic DM Tasks & Algorithms:

4. Association pattern mining (Week 10-11) ***Ass5:*** *Association DM*

5. Classification/prediction (Week 11-12) ***Ass6:*** *Classification DM*

6. Clustering analysis (Week 13)

7. Characterization/Generalization (Week 13)

5. Classification DM

(Text 3rd: 8.1-8.3 / 2nd: 6.1, 6.2-4, 6.6, 6.16)

- Classification problem overview
- General issues of classification DM
- Mining classification model by decision tree induction
- Bayesian classification
- Text classification
- Other classification methods
- Summary

E.g.1, Recap the classification problem: Risk Analysis Modeling:

***Target concept** with answers*

NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

Data from credit history of loan applications

E.g.2, Learning a model for predicting PlayTennis.

***Target concept** with answers*

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	sunny	hot	high	weak	No
2	sunny	hot	high	strong	No
3	overcast	hot	high	weak	Yes
4	rain	mild	high	weak	Yes
5	rain	cool	normal	weak	Yes
6	rain	cool	normal	strong	No
7	overcast	cool	normal	strong	Yes
8	sunny	mild	high	weak	No
9	sunny	cool	normal	weak	Yes
10	rain	mild	normal	weak	Yes
11	sunny	mild	normal	strong	Yes
12	overcast	mild	high	strong	Yes
13	overcast	hot	normal	false	Yes
14	rain	mild	high	strong	No

E.g.3, Learning a model to predict the relevance of search results on homedepot.com

- **Kaggle competition:** Predict the relevance of search results on homedepot.com
  - https://www.kaggle.com/c/home-depot-product-search-relevance
  - **Started:** Monday 18 January 2016
    Ends: Monday 25 April 2016
- **Data fields** (https://www.kaggle.com/c/home-depot-product-search-relevance/data)
  - **id** - a unique Id field which represents a (search_term, product_uid) pair
  - **product_uid** - an id for the products
  - **product_title** - the product title
  - **product_description** - the text description of the product (may contain HTML content)
  - **search_term** - the search query
  - **relevance** - the average of the relevance ratings for a given id (from 1 to 3)
  - **name** - an attribute name
  - **value** - the attribute's value

Target concept

Define Classification Mining

- General description/definition:
  - Given a **training data set** $D = \{t_1, t_2, \ldots, t_n\}$ which has n rows (records, tupels) and m columns (features, attributes); for a given **target concept** (i.e. a selected feature) with **classes** $C = \{c_1, \ldots, c_p\}$, discover a **model** (knowledge/pattern/rules) for **predicting** a target value c_i of a new instance.
- Classification DM: Find a mapping function ***f:*** $D \Rightarrow C$, i.e. the classification knowledge or model, where each t_i in D is assigned to a class c_j of C.

Classification DM: find the hidden model

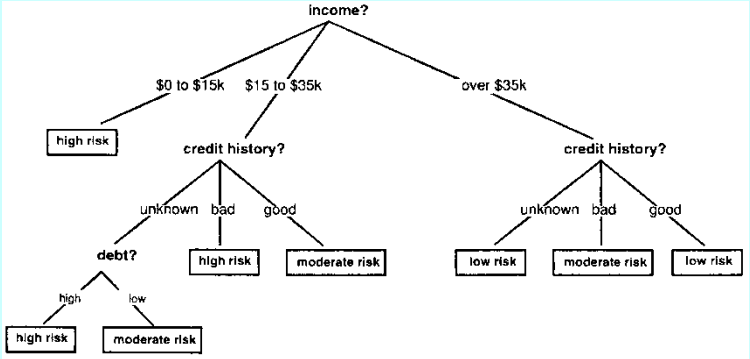
E.g., Find a model for "Risk" target concept.

Training set: D

NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1.	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

Data from credit history of loan applications

Classification Model: ***f***



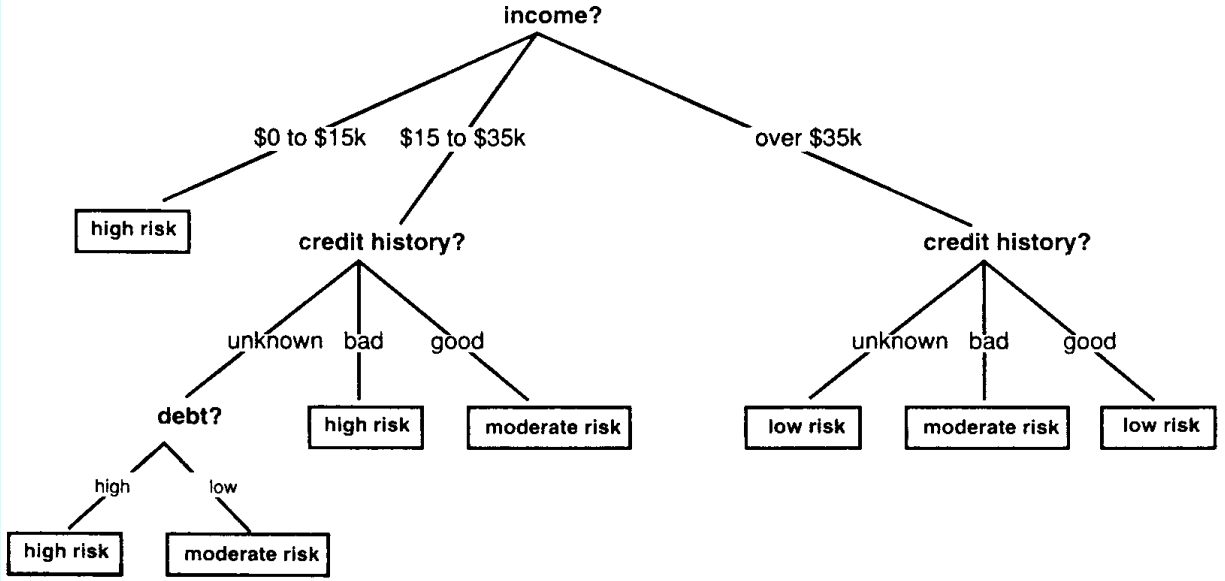
A simplified decision tree for credit risk assessment.

Class Predication: Apply *f* (coded as a classifier)

E.g., Predicate the risk value for a new instance:

<CREDIT_HIS=good, DEBT=low, COLLATERAL=unknown, INCOME=$30k>

RISK = ?

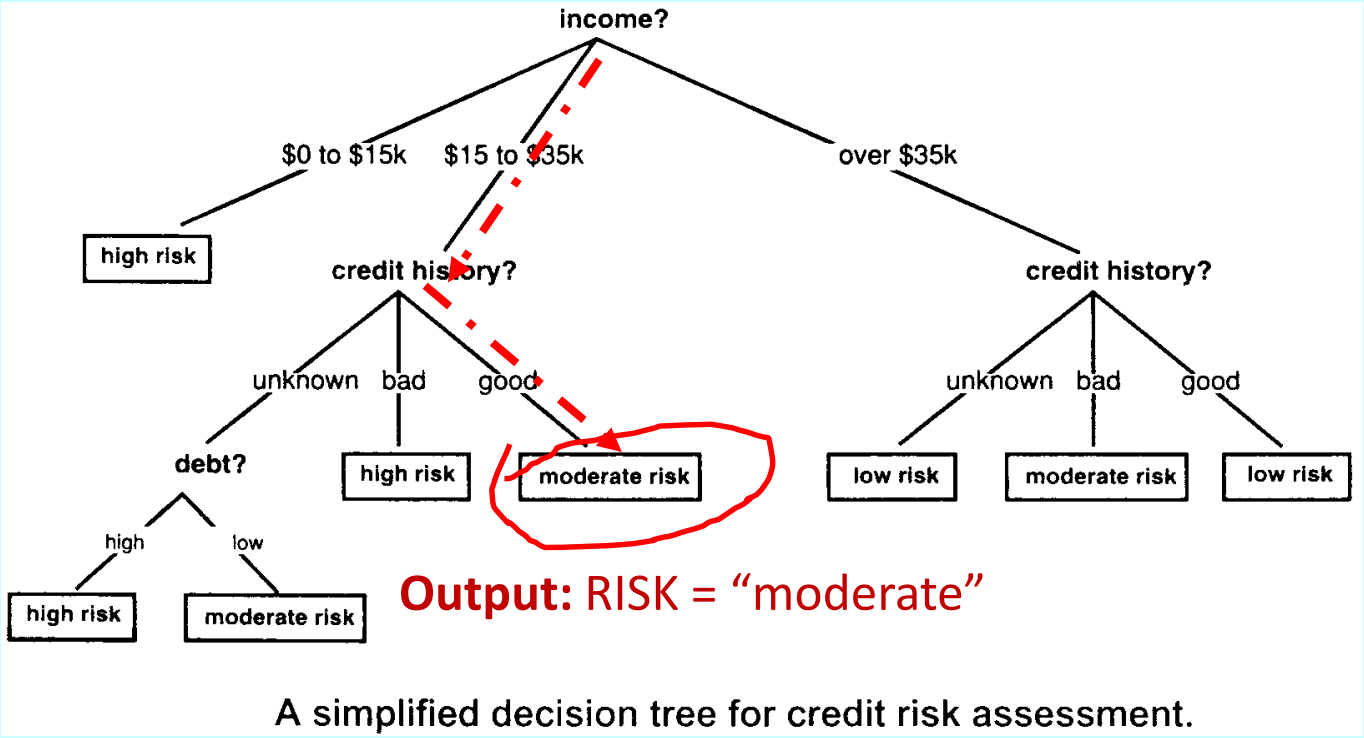


A simplified decision tree for credit risk assessment.

Apply the classifier (coded model) to predict

Input: <CREDIT_HIS=good, DEBT=low, COLLATERAL=unknown, INCOME=$30k>
Output: RISK = ?

Decision rule: If INCOME=$30k (i.e. 15-35k) and CREDIT_HIS=good, then Risk = moderate

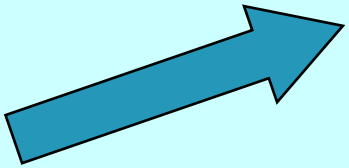


A simplified decision tree for credit risk assessment.

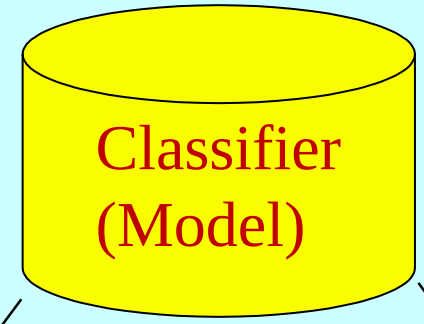
Learning strategy of model finding

- **Discover the classification model *f*, via inductive learning on training data**
  - It is a divide (partition D) and conquer (find C) based approach for finding a decision model ***f***.
  - The model ***f*** is the classes predictor, and also the classes descriptor (or the definition of the target concept, if the ***f*** is transparent).
- Prediction is to assign a new unseen object to a class of the target concept
  - Apply the formed model ***f*** to predicate a class for a new data case.

E.g., Classification pattern discovery: **model construction**



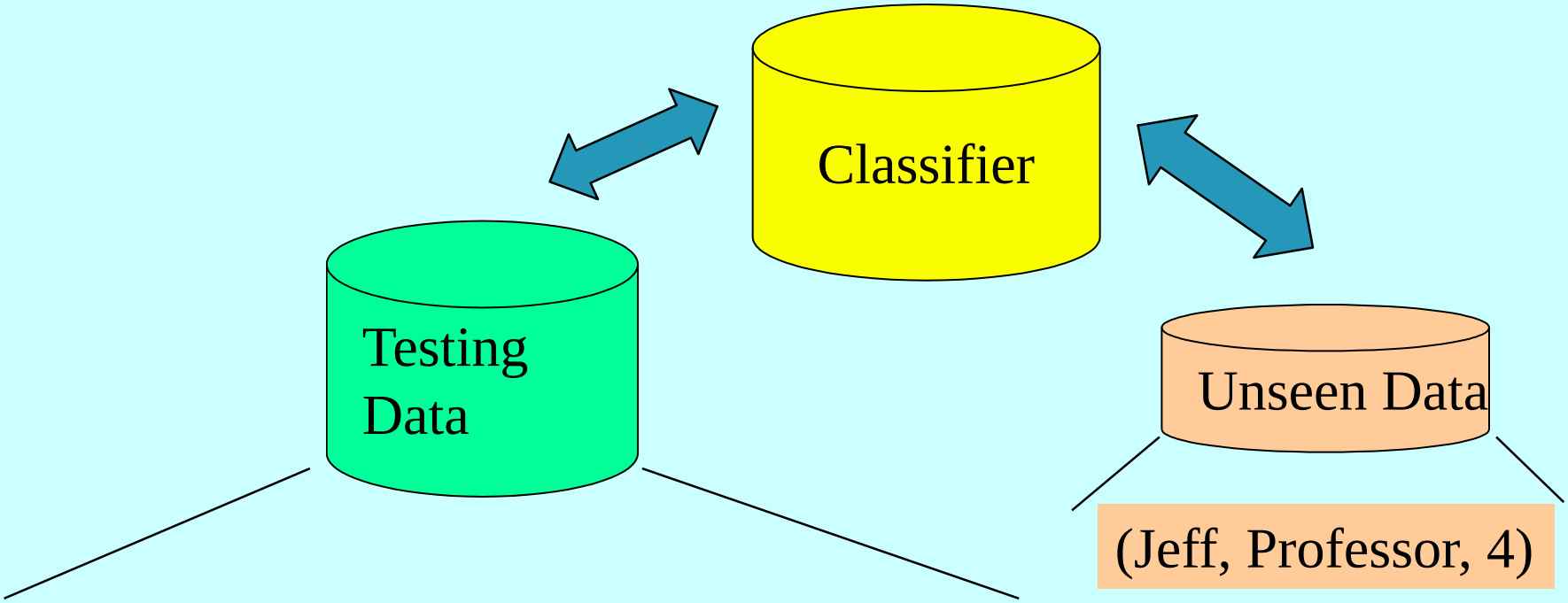
Classification Algorithms



NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Class prediction: class labeling



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Classification application development: a three-phase process

- **Model construction:** find classification rules from the training data set

  1. Training:

  – Each tuple is assumed to belong to a predefined class, as determined by the class label attribute

  – The set of tuples used for model construction is training set

  – The model is represented as classification rules, decision trees, or distributed network weights, mathematical formula, etc.

- **Model usage:** for classifying unseen objects

  2. Testing: Estimate accuracy of the model

  – The known label of test sample is compared with the classified result from the model

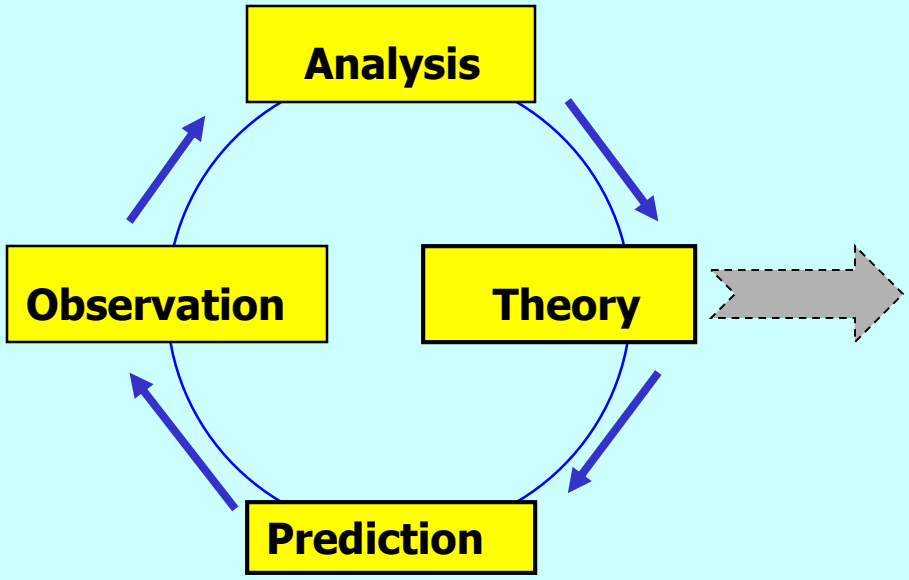
  – Accuracy rate is the percentage of test set samples that are correctly classified by the model

  – Test set is independent of training set

  3. Predicting: If the accuracy is acceptable, use the model to classify unseen data

How does classification match the empirical cycle?

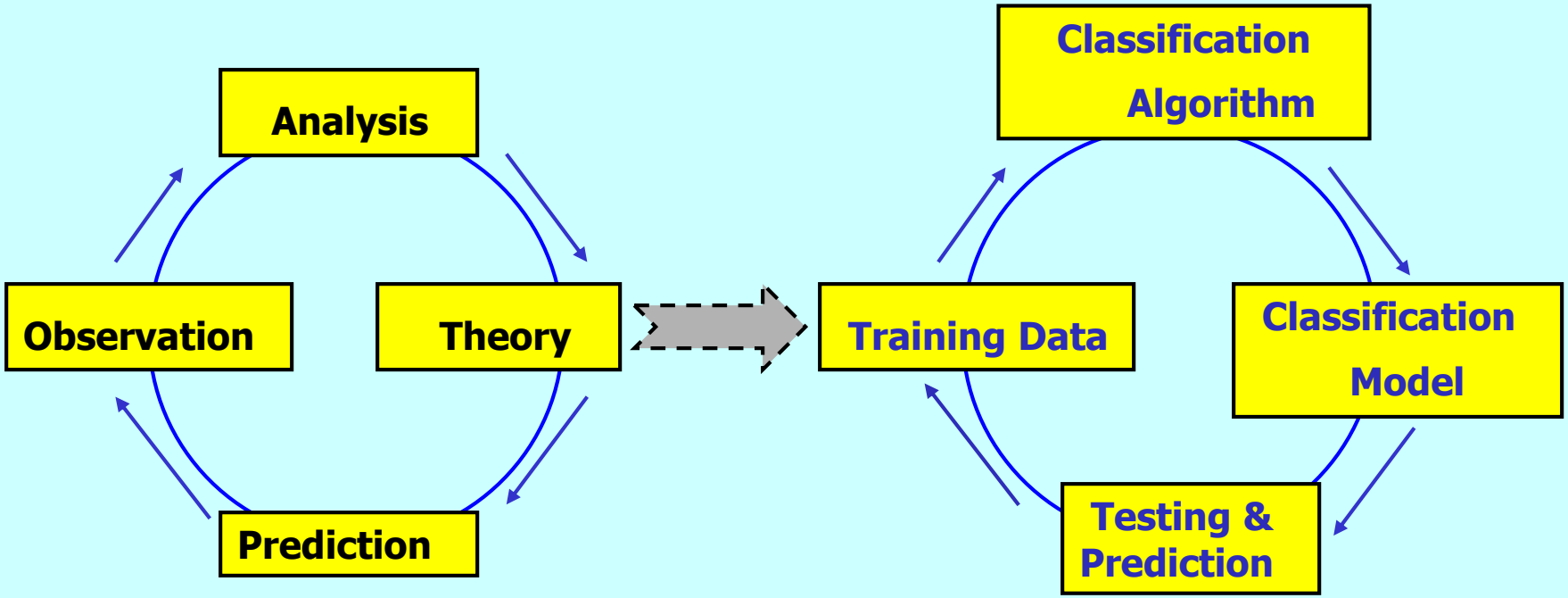
Samples of past experience with known answers are examined and generalized to predict for future cases.



Classification model

How does classification match the empirical cycle?

Samples of past experience with known answers are examined and generalized to predict for future cases.



- **Observation:** The process starts with a number of observations - **training examples.**
- **Analysis:** **Discover classification regularities** for the target in the database, e.g., induction by **sorting the examples into classes** by continuously selecting the best attribute and partitioning the training data into subsets.
- **Theory:** The discovered **classification pattern** is formalized that explains the classes of the data well, such as forming **decision rules**.
- **Prediction:** Use the discovered rules to predict a target value for a new case.

How to evaluating classification methods?

- **Predictive accuracy**
- **Speed and scalability**
  - Time to construct and use the model
  - Ability of handling large data set
- **Robustness**
  - Handling noise and missing values
- **Interpretability:**
  - understanding and insight provided by the model
- **Goodness of rules**
  - decision tree size
  - compactness of classification rules

Data preparation for classification

- **Data cleaning**
  - Preprocess data in order to reduce noise and handle missing values
- **Relevance analysis (feature selection)**
  - Remove the irrelevant or redundant attributes
- **Data transformation**
  - Generalize or normalize data
  - E.g., Use discretization techniques to convert a given continuous attribute into categorical attribute, e.g., age, income, etc.
    - Equal-width (distance) partitioning method
    - Information (entropy) based method, etc.

Basic Leaning Approach - Inductive Reasoning

- **What is "Induction"?**
  - It is a reasoning approach in that a concept can be learned/supported by gathering evidences from individual observations. However, it can not prove the concept.
    - E.g. Induce the concept of "swans are white" from the observations of facts.
    - This indicates the process of **reason from observation**: to make a statement based on the observation of facts.
    - It is not a sound logic reasoning (i.e. deductive reasoning), but a plausible reasoning technique with uncertainty involved.

Decision Tree Induction

- **It is a supervised inductive learning process:**
  - Partitioning training data based on **divide-and-conquer strategy.**
  - Continue dividing D into subsets, based a search method, until each subset has only one label, i.e. all examples in the subset share a same class label.
  - E.g., Each tuple of D is placed into a group representing the region within which it falls.

Decision Tree Induction (cont)

- **Decision Tree (DT):**
  - It uses a graph of tree as a tool to model decision finding process and their possible consequences, including chance event outcomes, resource costs, and utility.
  - All DT methods use DT as a hierarchical data structure and differ in how the tree is built (i.e. DT Induction).
  - An internal node of DT associates with a selected attribute and arcs with values for that attribute.

DT Classification

Training Data Set D:

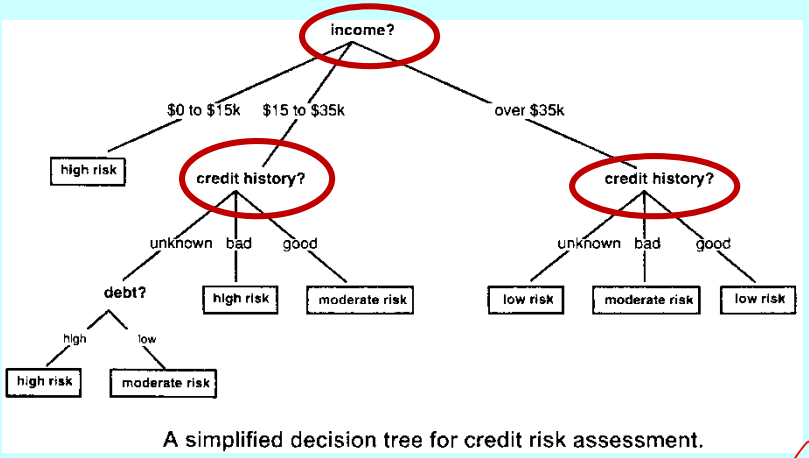
NO.	RISK	CREDIT HISTORY	DEBT	COLLATERAL	INCOME
1	high	bad	high	none	$0 to $15k
2.	high	unknown	high	none	$15 to $35k
3.	moderate	unknown	low	none	$15 to $35k
4.	high	unknown	low	none	$0 to $15k
5.	low	unknown	low	none	over $35k
6.	low	unknown	low	adequate	over $35k
7.	high	bad	low	none	$0 to $15k
8.	moderate	bad	low	adequate	over $35k
9.	low	good	low	none	over $35k
10.	low	good	high	adequate	over $35k
11.	high	good	high	none	$0 to $15k
12.	moderate	good	high	none	$15 to $35k
13.	low	good	high	none	over $35k
14.	high	bad	high	none	$15 to $35k

Data from credit history of loan applications

DT:

Decision node: ⬭

Leave node: ▭



A simplified decision tree for credit risk assessment.

Predication:

Input: <RISK=?, CREDIT HIS=good, DEBT=low, COLLATERAL=unknown, INCOME=$30k>

Output: RISK= **moderate risk**

E.g., Generate a model for Play Tennis.

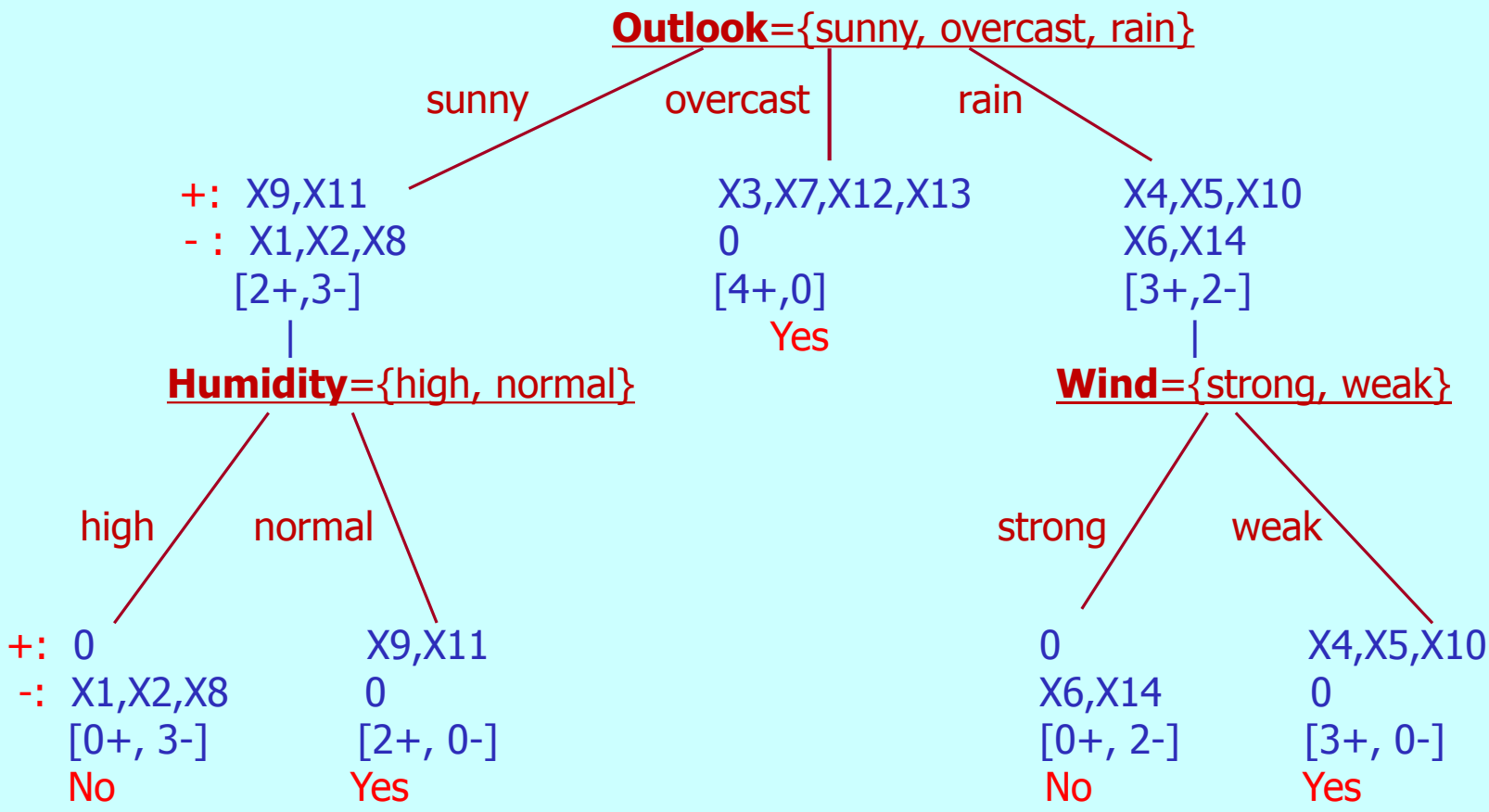
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
1	sunny	hot	high	weak	No
2	sunny	hot	high	strong	No
3	overcast	hot	high	weak	Yes
4	rain	mild	high	weak	Yes
5	rain	cool	normal	weak	Yes
6	rain	cool	normal	strong	No
7	overcast	cool	normal	strong	Yes
8	sunny	mild	high	weak	No
9	sunny	cool	normal	weak	Yes
10	rain	mild	normal	weak	Yes
11	sunny	mild	normal	strong	Yes
12	overcast	mild	high	strong	Yes
13	overcast	hot	normal	false	Yes
14	rain	mild	high	strong	No

Induction by sorting examples

D: (Outlook, Temperature, Humidity, Wind, PlayTennis)

+: X3,X4,X5,X7,X9,X11,X12,X13 Yes for PlayTennis

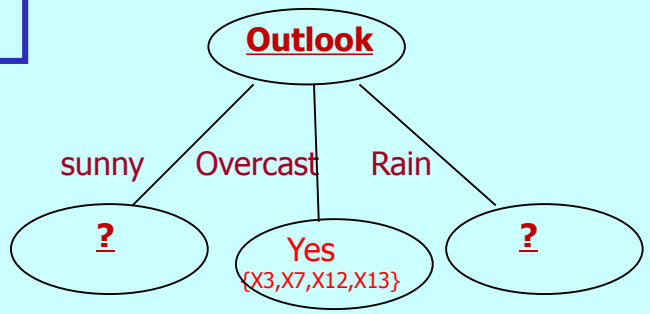
--: X1,X2,X6,X8,X14 No for PlayTennis



DT: Splits Area: Sort examples into subareas/subgroups

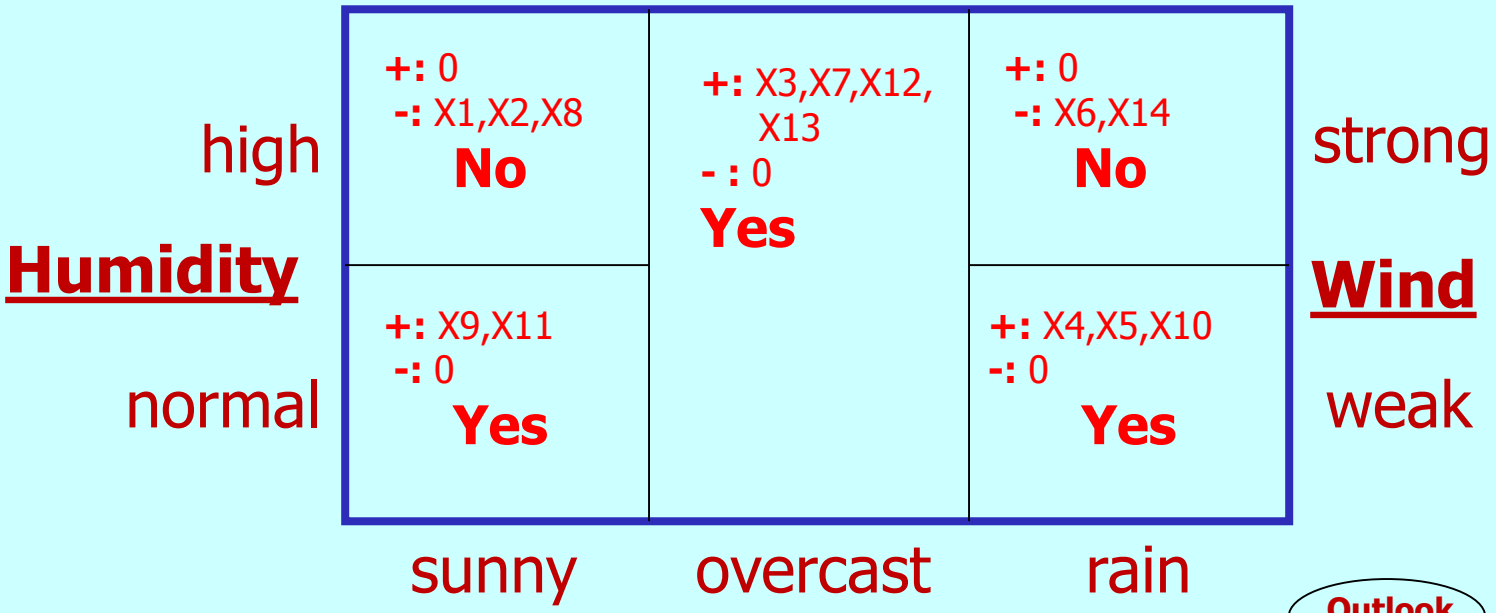
sunny	overcast	rain
+: X9,X11 - : X1,X2,X8	+: X3,X7,X12, X13 - : 0 **Yes**	+: X4,X5,X10 - : X6,X14

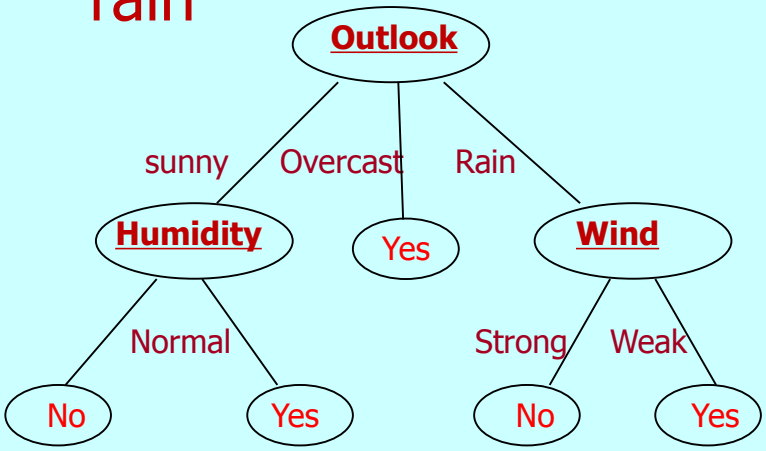
Outlook



D: (Outlook, Temperature, Humidity, Wind, PlayTennis)

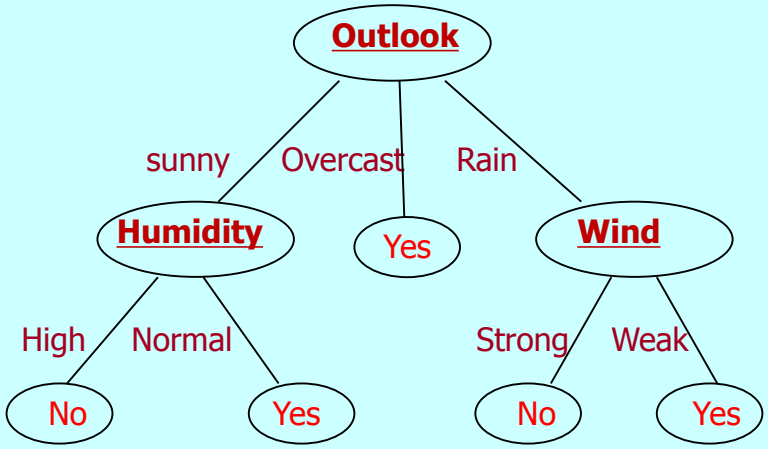
DT: Splits Area

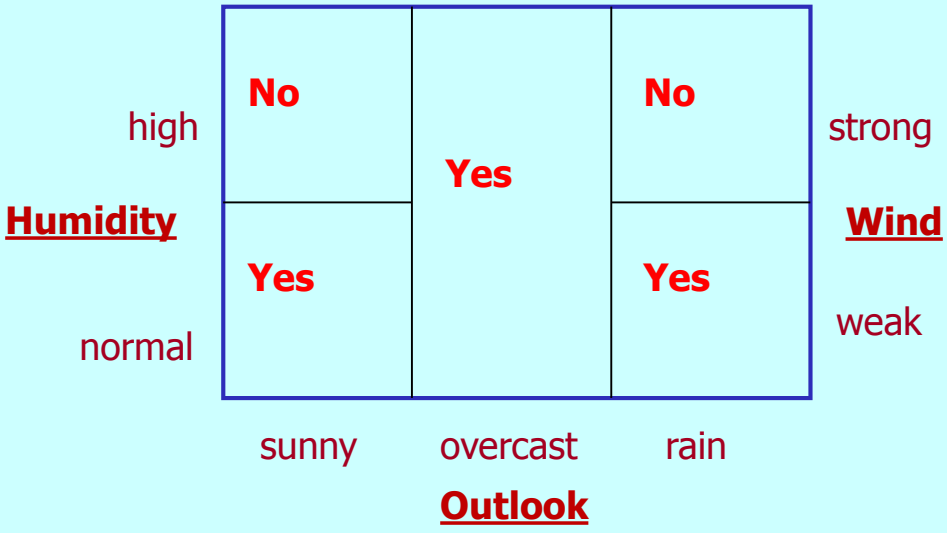




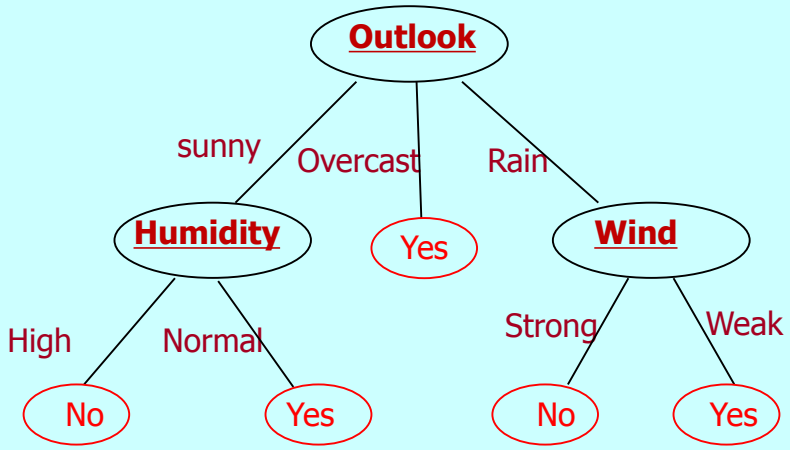
DT:

Constructs three= Splits area





Define DT



Given:

- Training data set D with n tuples: D = $\{t_1, \ldots, t_n\}$
- Schema of D with m attributes: $(A_1, A_2, \ldots, A_m)$
- Classes: $C=\{C_1, \ldots., C_p\}$ of a selected target concept

Decision Tree (a tree associated with D) is defined:

- A root node has no incoming arcs and zero or more outgoing arcs
- Each internal node is a selected attribute, A_i
- Each arc is labeled with an attribute value (predicate condition) of the parent node
- Each leaf node is labeled with a class, C_j

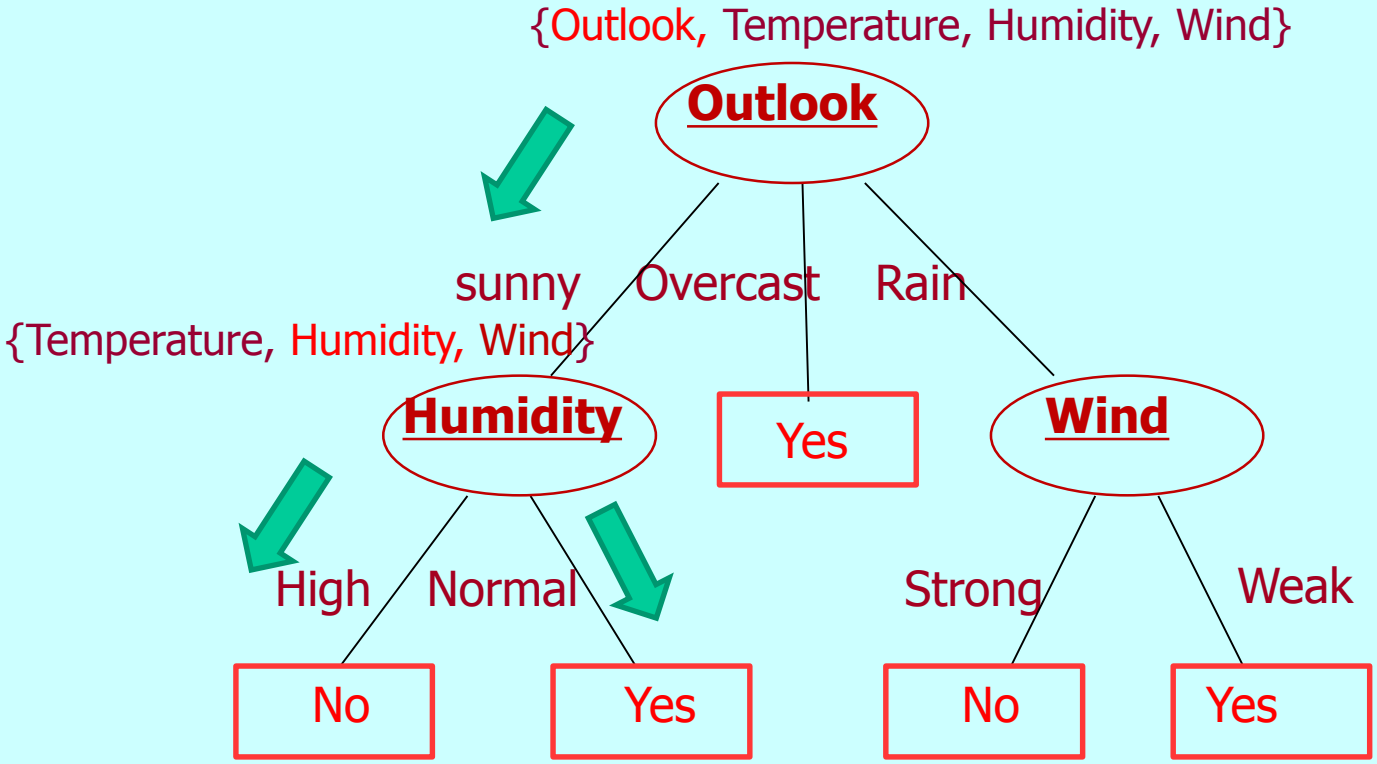
A General DT Induction Algorithm

```
Input:
   D      //Training data
Output:
   T      //Decision Tree
DTBuild Algorithm:
          //Simplistic algorithm to illustrate naive approach to building DT
   T = ∅;
   Determine best splitting criterion;
   T = Create root node node and label with splitting attribute;
   T = Add arc to root node for each split predicate and label;
   for each arc do
      D = Database created by applying splitting predicate to D;
      if stopping point reached for this path then
         T' = Create leaf node and label with appropriate class;
      else
         T' = DTBuild(D);
      T = Add T' to arc;
```

Greedy search

Recap DT Classification Learning Concepts:

- **Supervised learning** (with a training data set)
- **Top-down, Divide & Conquer strategy** (by testing and splitting the training dataset & subsets)
- **Tree construction/induction by Greedy Search**, i.e. Depth-first search + heuristic function



Technique Issues of DT Classification

- Preparing datasets: (training & testing)
  - A training dataset for learning a model
  - A test dataset for evaluating the learned model
- Classification model discovery: (constructing a DT)
  - Stopping criteria for testing at each node
  - How to choose which attribute to split, and how to split (method)
  - Control structure for tree construction (recursive process)
  - Pruning method

DT Induction: Divide & Conquer

```
Input:
   D      //Training data
Output:
   T      //Decision Tree
DTBuild Algorithm:
          //Simplistic algorithm to illustrate naive approach to building DT
   T = ∅;
   Determine best splitting criterion;
   T = Create root node node and label with splitting attribute;
   T = Add arc to root node for each split predicate and label;
   for each arc do
      D = Database created by applying splitting predicate to D;
      if stopping point reached for this path then
         T' = Create leaf node and label with appropriate class;
      else
         T' = DTBuild(D);
      T = Add T' to arc;
```

Choose attribute to split

Split

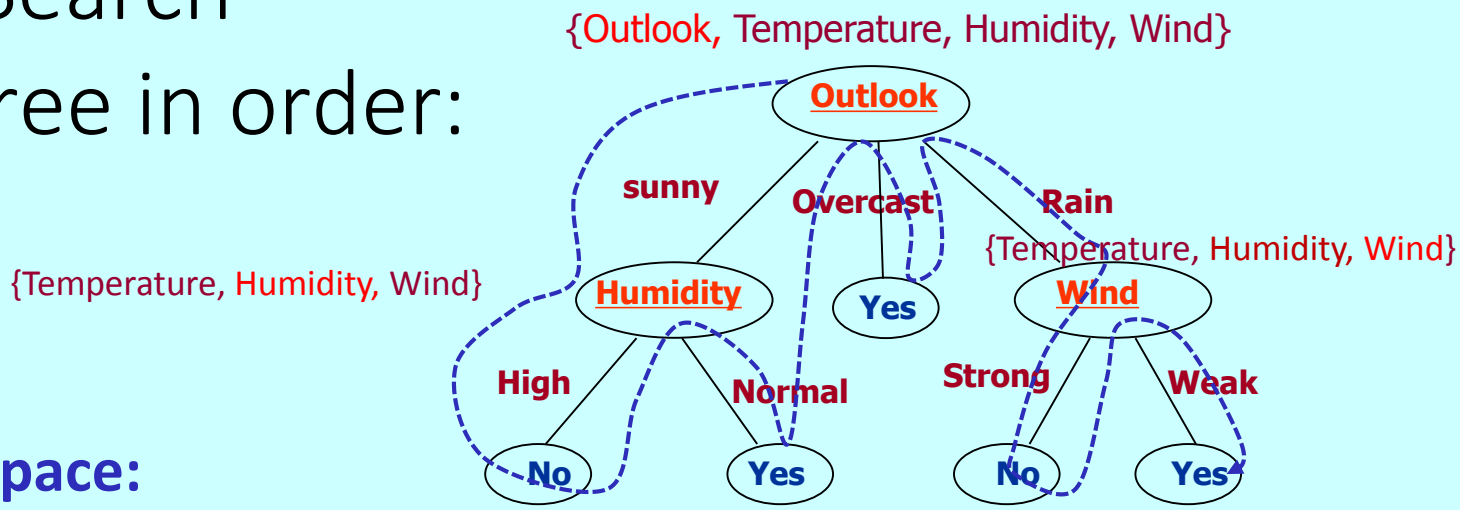
Test

Choose attribute to split

DT Algorithms

- ID3 (Interactive Dichotomiser 3)
  - Quinlan, J.R., "Induction of decision trees", Machine Learning, Vol. 1, No.1, pp 81-106, 1986.
- C4.5 (C5.0)
  - Quinlan, J.R., "C4.5: Programs for Machine Learning", San Francisco: Morgan Kaufman, 1993.
- Others: CART, CHAID, Chi-squared, etc.

Greedy Search-
Build a tree in order:



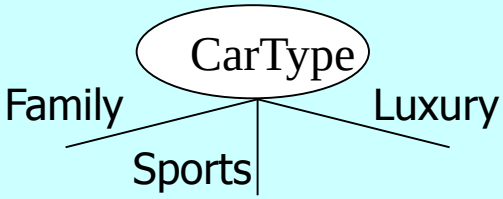
- **Search space:**
  - There are exponentially many decision trees base on same set of attributes.
- **Goal:**
  - Find accurate, optimal decision tree in a reasonable amount of time.
- **Strategy:**
  - Greedy search (depth first search + attribute evaluation).

How to Specify Test Condition?

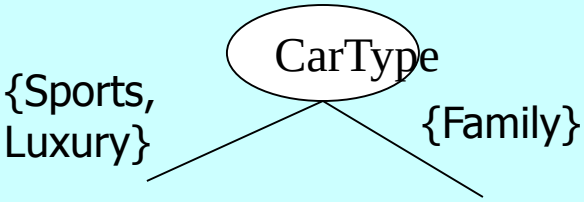
- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

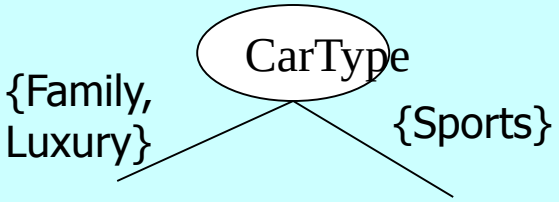
- Multi-way split: Use as many partitions as distinct values



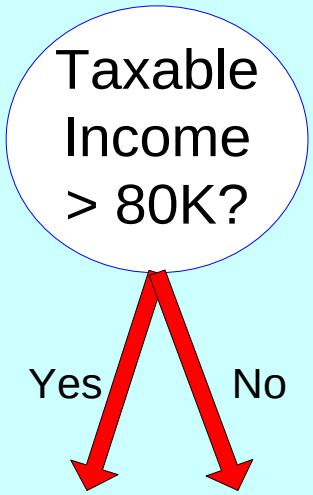
- Binary split: Divides values into two subsets. Need to find optimal partitioning



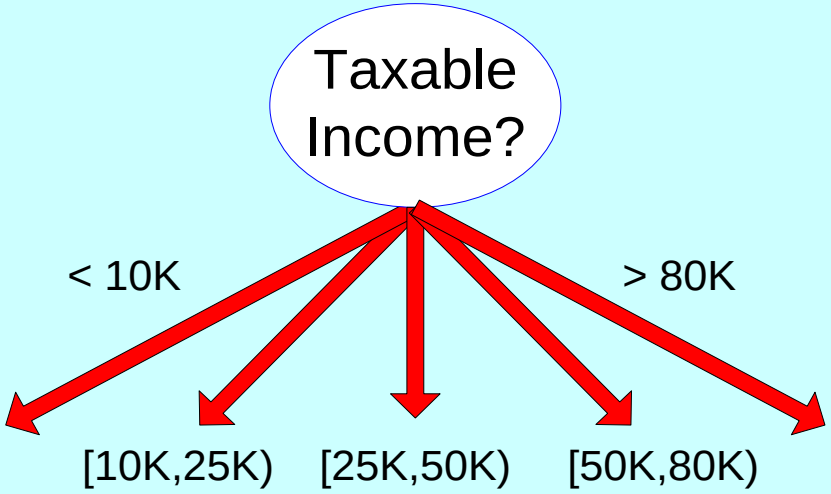
OR



Splitting Based on Continuous Attributes (static)



(i) Binary split



(ii) Multi-way split

DT Induction Summary:

- Based on a Greedy Search strategy
  - Depth-first + Heuristic function
  - Split the records based on an attribute test that optimizes certain criterion
- Issues on partitioning data set
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

Review Questions

1. Why classification mining is a supervised learning process? How about association mining?
2. What are the major phases of conducting a classification mining application?
3. Can you describe a mapping between a classification application process and the empirical cycle?
4. What is the general idea/strategy/approach of DT induction for classification mining?