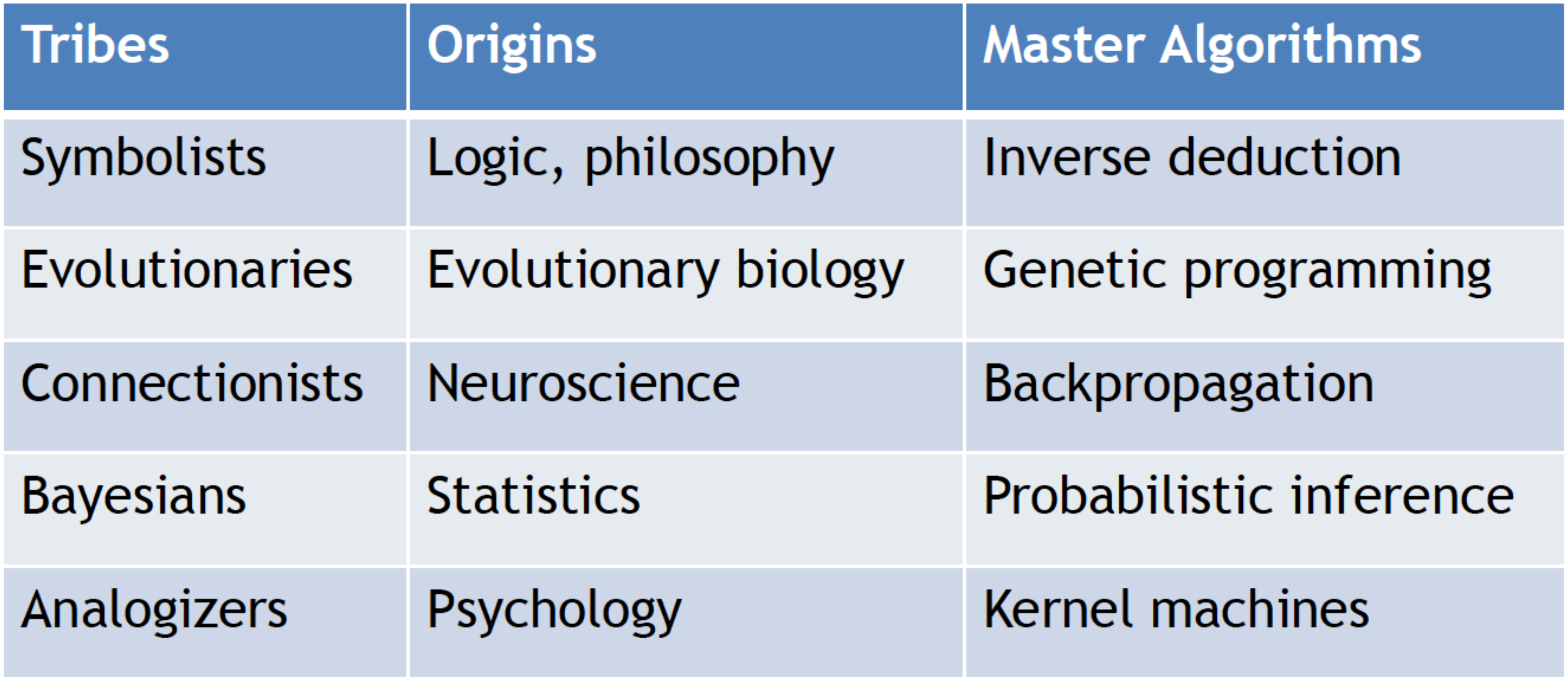
Unsupervised Learning
Lecture 2: Decision Tree and Random Forest

Tho Quan
qttho@hcmut.edu.vn

Agenda

- Inductive learning
- Decision Tree: ID3 and C4.5
- From Decision Tree to Random Forest

Five Tribes of Machine Learning



Tribes	Origins	Master Algorithms
Symbolists	Logic, philosophy	Inverse deduction
Evolutionaries	Evolutionary biology	Genetic programming
Connectionists	Neuroscience	Backpropagation
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

The five tribes of machine learning, Pedro Domingos

Inference Mechanisms

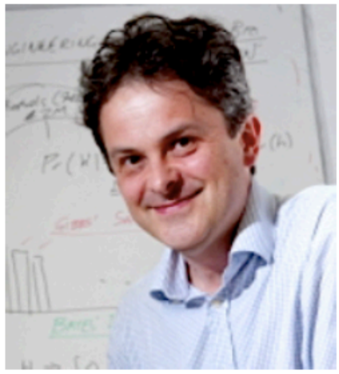
- Deduction: cause + rule → effect
- Abduction: effect + rule → cause
- Induction: cause + effect → rule

- Inference must be made on closed-world assumption
  - All propositions must be TRUE of FALSE
  - Unknown proposition → FALSE

- IF temperature high AND NOT (water level low) THEN pressure high
- IF tranducer output low THEN water level low

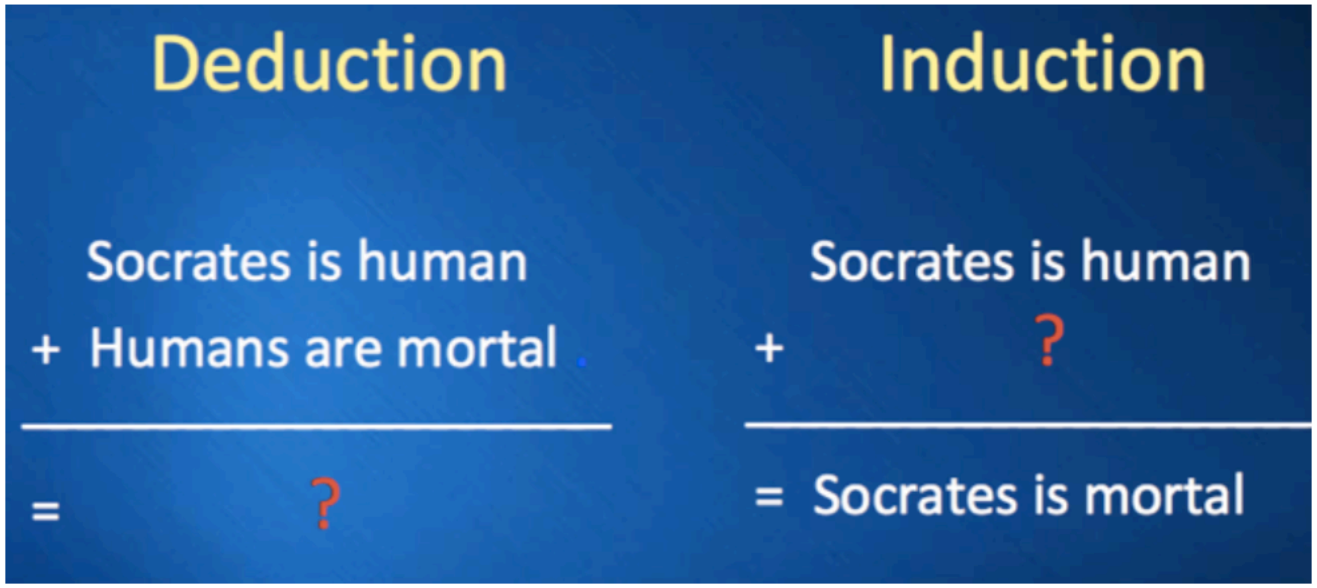
Deductiion and Induction

Tom Mitchell



Steve Muggleton

Ross Quinlan



Deduction

Socrates is human

+ Humans are mortal

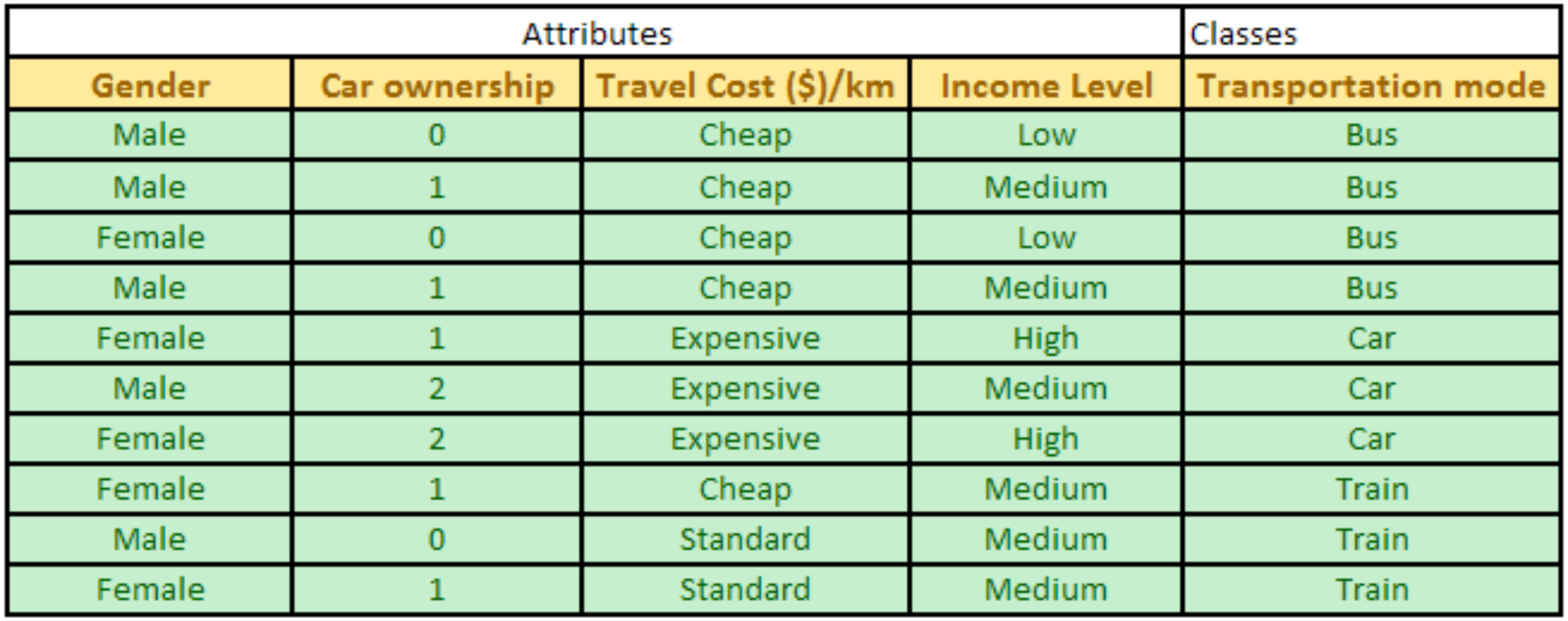
= ?

Induction

Socrates is human

+ ?

= Socrates is mortal



Attributes				Classes
Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Female	1	Cheap	Medium	Train
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

Rule 1 : If Travel cost/km is expensive then mode = car

Rule 2 : If Travel cost/km is standard then mode = train

Rule 3 : If Travel cost/km is cheap and gender is male then mode = bus

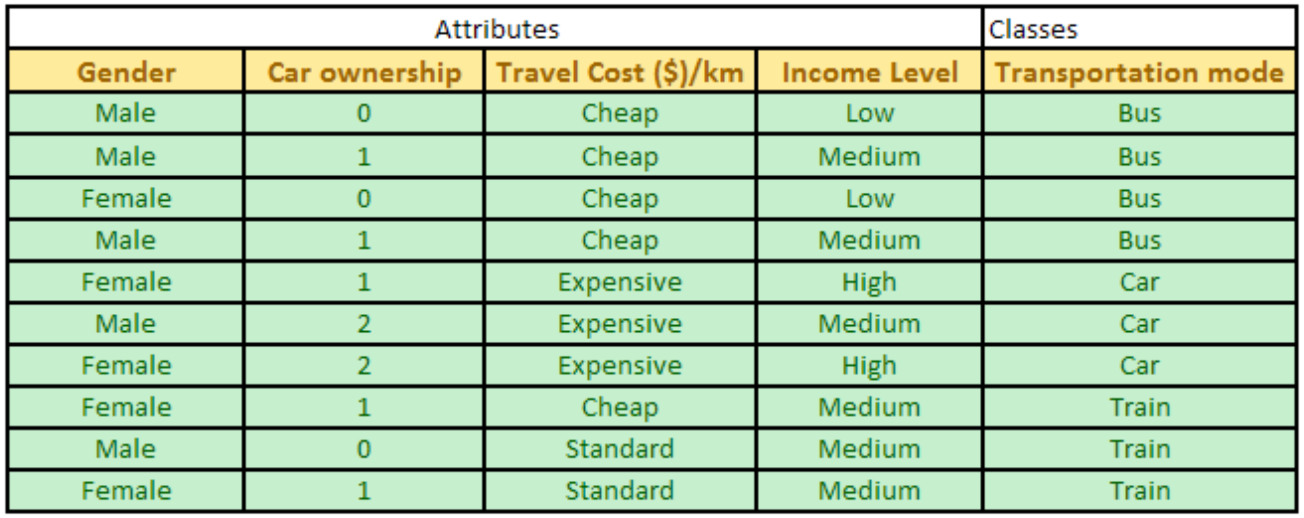
Rule 4 : If Travel cost/km is cheap and gender is female and she owns no car then mode = bus

Rule 5 : If Travel cost/km is cheap and gender is female and she owns 1 car then mode = train

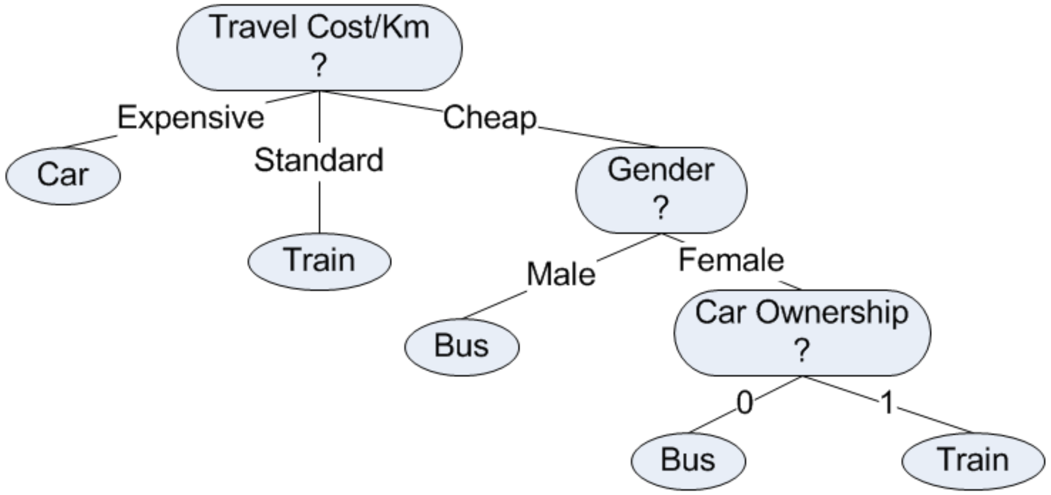
Gender :Male
Car Ownership : 1
Travel Cost/Km : Standard
Income Level : High
Transportation Mode ?

Decision Tree

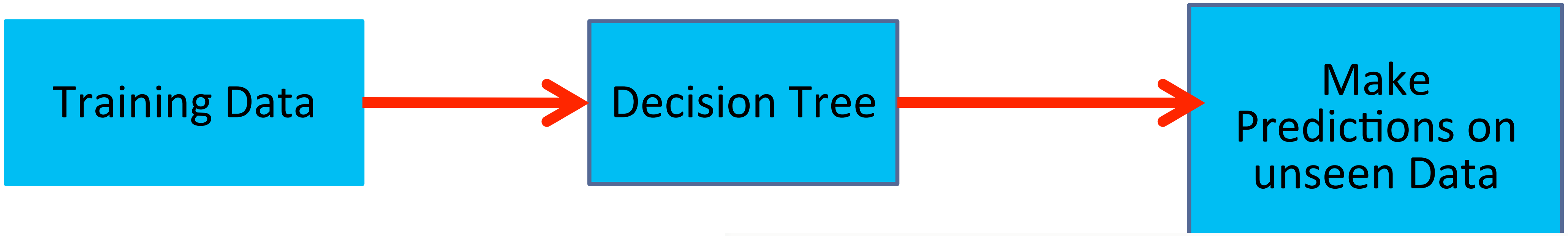
- Decision Tree is a hierarchical tree structure that used to classify classes based on a series of questions (or rules) about the attributes of the class
- Decision tree representation:
  - Each internal node tests an attribute
  - Each branch corresponds to attribute value
  - Each leaf node assigns a classification



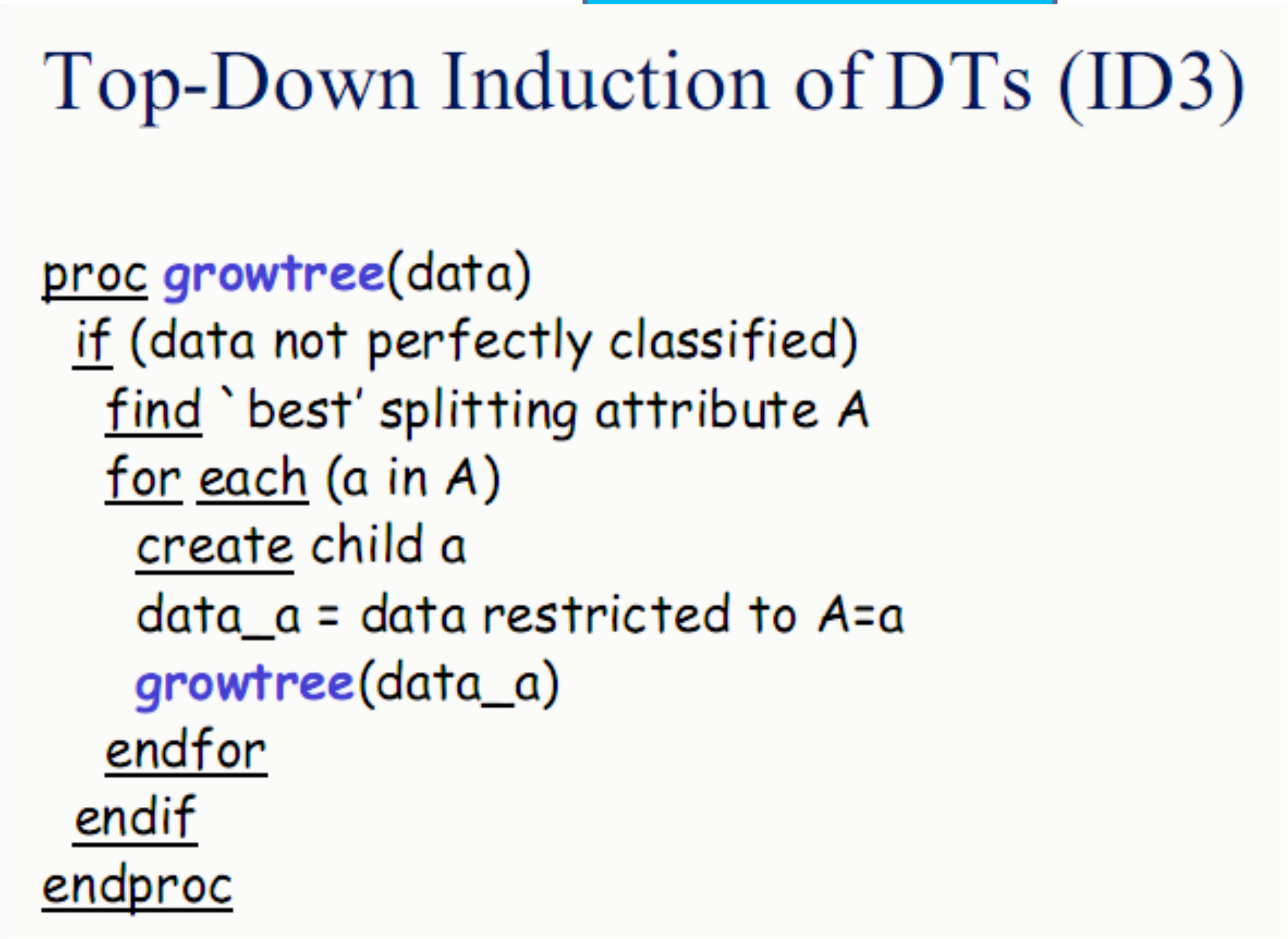
Attributes				Classes
Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Female	1	Cheap	Medium	Train
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train



Generate a Decision Tree



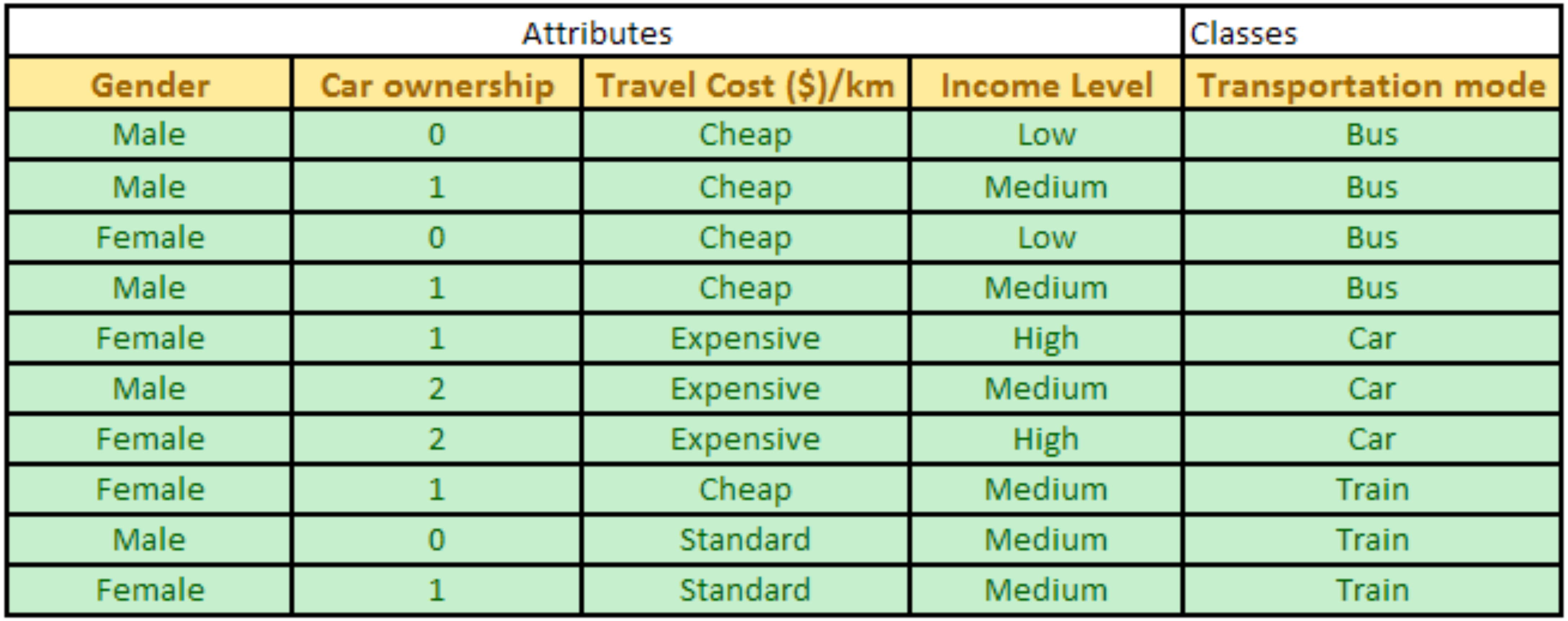
- Choose best attribute
- Split data set
- Recurse until each data item classified correctly



Top-Down Induction of DTs (ID3)

```
proc growtree(data)
  if (data not perfectly classified)
    find `best' splitting attribute A
    for each (a in A)
      create child a
      data_a = data restricted to A=a
      growtree(data_a)
    endfor
  endif
endproc
```

Generate a Decision Tree



Attributes				Classes
Gender	**Car ownership**	**Travel Cost ($)/km**	**Income Level**	**Transportation mode**
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Female	1	Cheap	Medium	Train
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

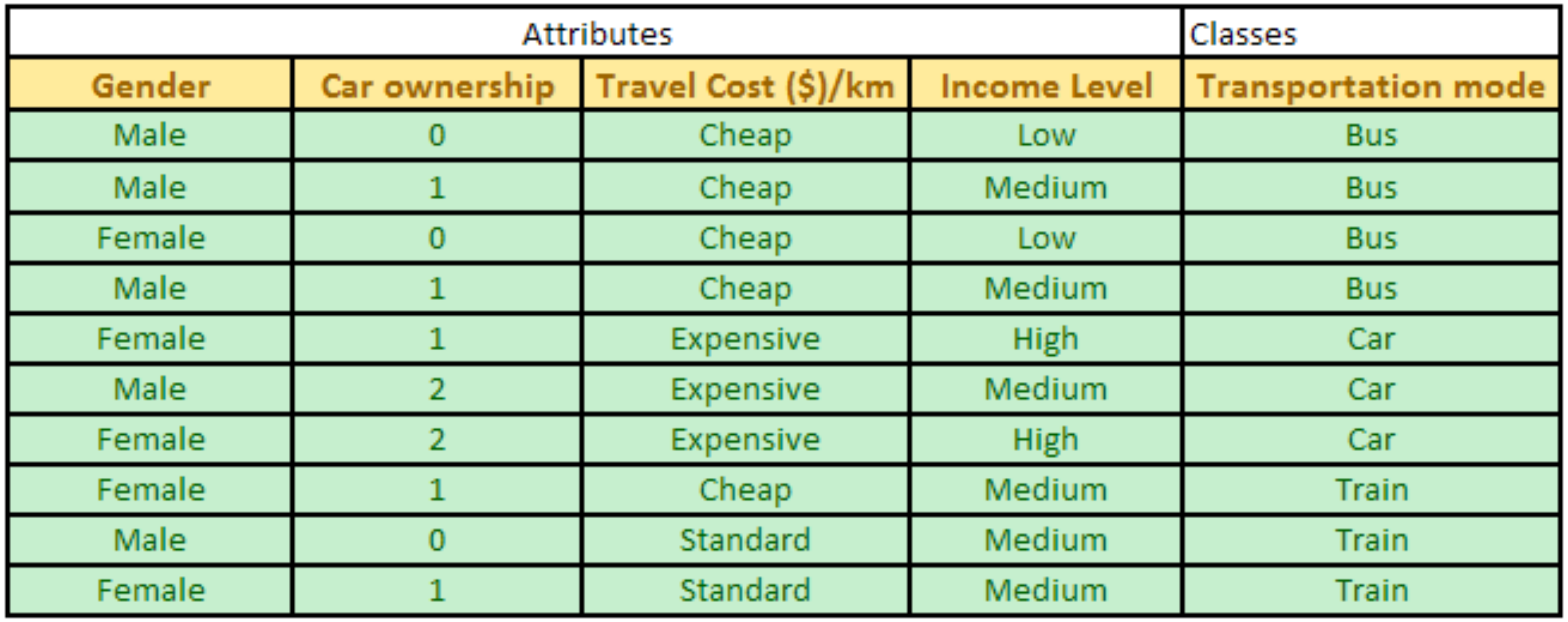
Generate a Decision Tree

- Measure Impurity :

$$Entropy = \sum_{j} -p_j log_2 p_j \quad Gini\ Index = 1 - \sum_{j} p_j^2$$

- Information Gain :

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

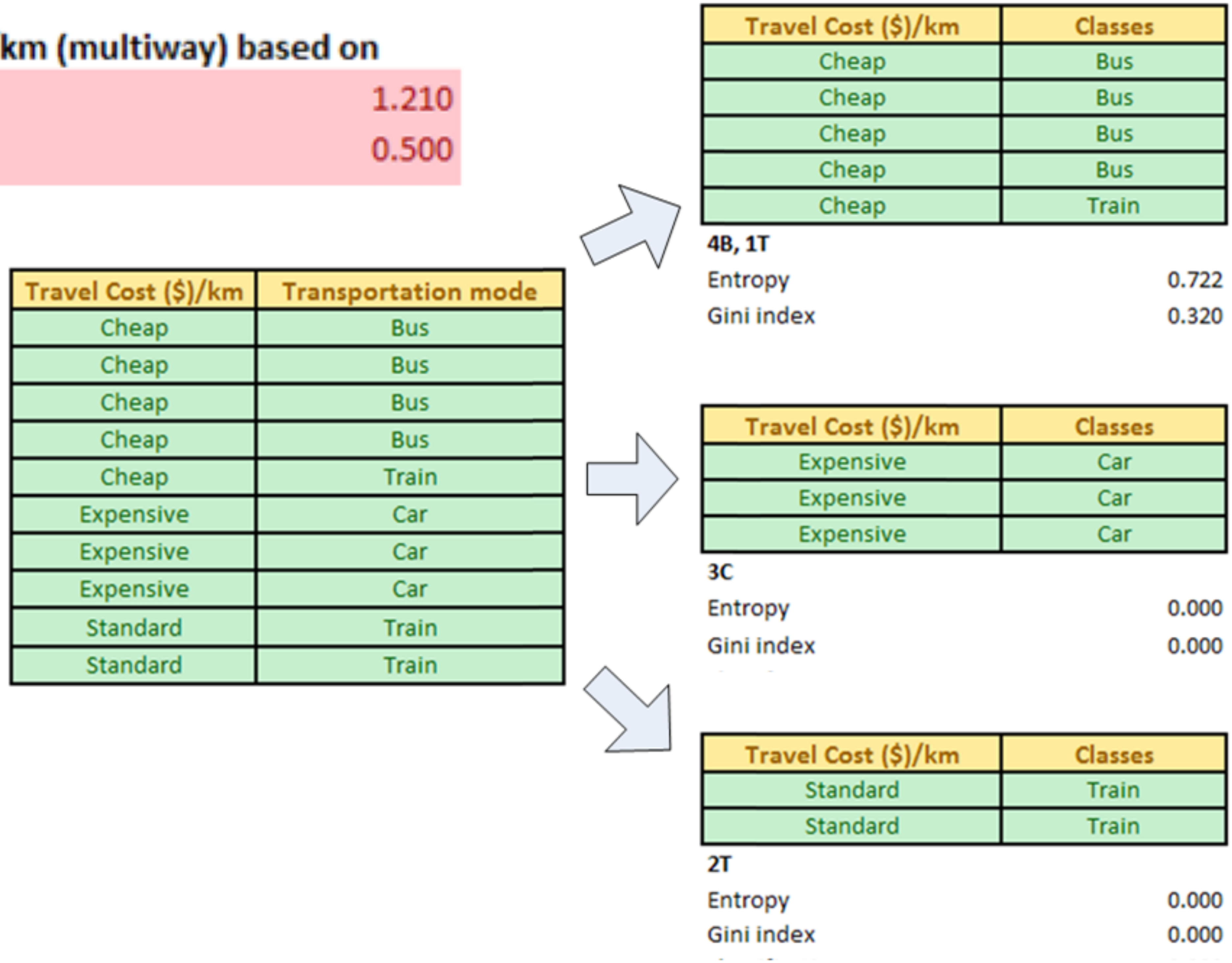


Attributes				Classes
Gender	Car ownership	Travel Cost ($)/km	Income Level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car
Female	1	Cheap	Medium	Train
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train

- Pro(Bus) = 4/10
- Pro(Car) = 3/10
- Pro(Train) = 3/10
- Entropy = – 0.4 log (0.4) – 0.3 log (0.3) – 0.3 log (0.3) = 1.571
- Gini Index = 1 – (0.4^2 + 0.3^2 + 0.3^2) = 0.660

Gain of Travel Cost/km (multiway) based on

Entropy	1.210
Gini index	0.500



Travel Cost ($)/km	Transportation mode
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Train
Expensive	Car
Expensive	Car
Expensive	Car
Standard	Train
Standard	Train

Travel Cost ($)/km	Classes
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Bus
Cheap	Train

4B, 1T

Entropy	0.722
Gini index	0.320

Travel Cost ($)/km	Classes
Expensive	Car
Expensive	Car
Expensive	Car

3C

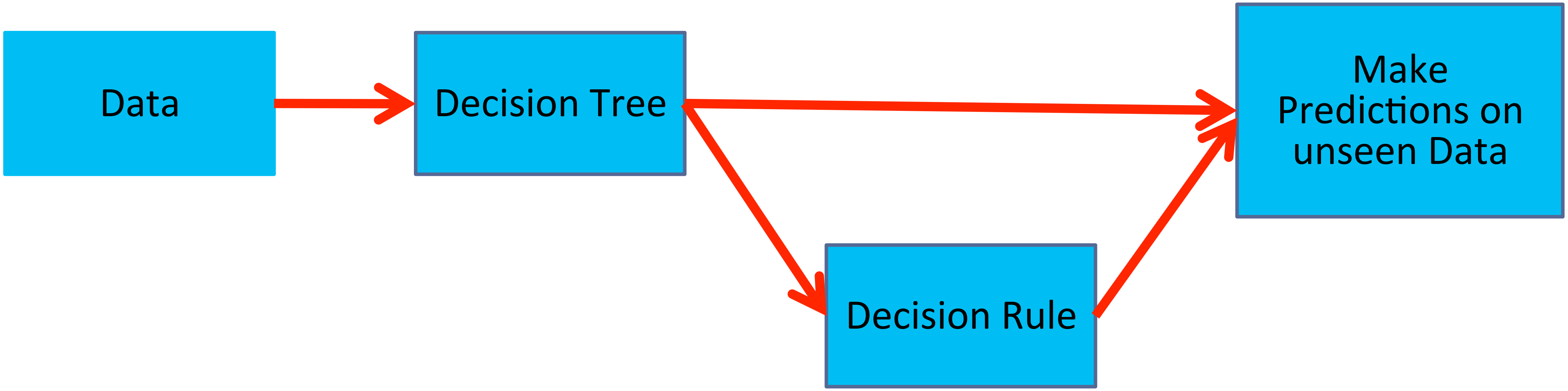
Entropy	0.000
Gini index	0.000

Travel Cost ($)/km	Classes
Standard	Train
Standard	Train

2T

Entropy	0.000
Gini index	0.000

How to Use a Decision Tree



Data

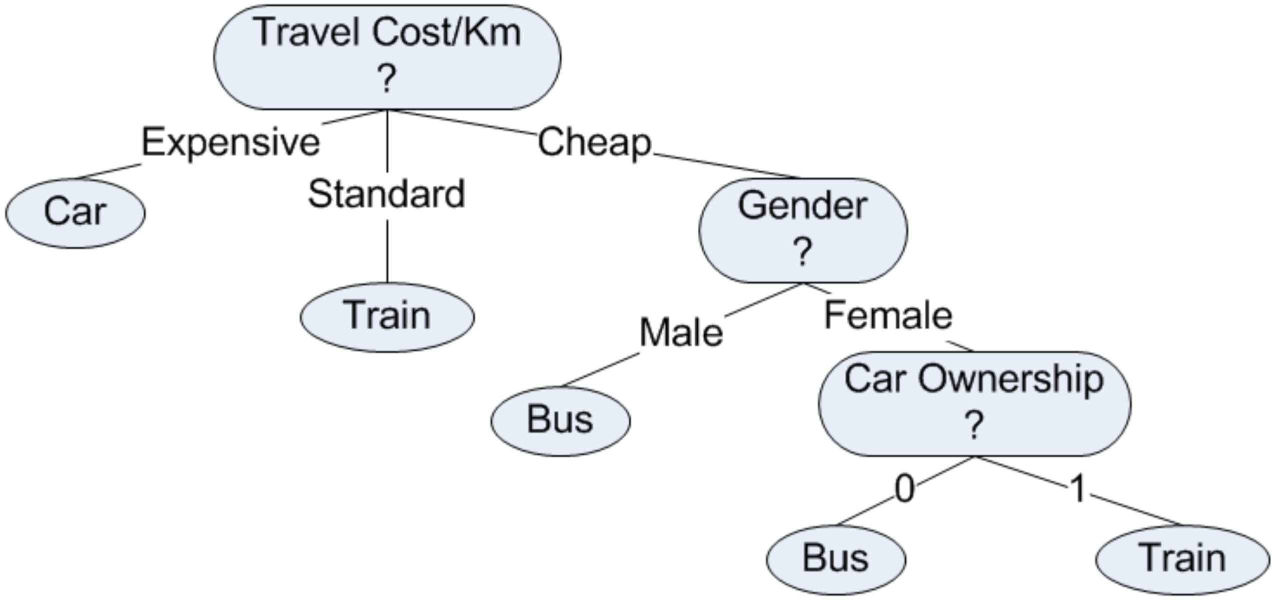
Decision Tree

Decision Rule

Make Predictions on unseen Data

How to Use a Decision Tree

- Gender :Male
- Car Ownership : 1
- Travel Cost/Km : Standard
- Income Level : High
- Transportation Mode ?



- **Rule 1** : If Travel cost/km is expensive then mode = car
- **Rule 2** : If Travel cost/km is standard then mode = train
- **Rule 3** : If Travel cost/km is cheap and gender is male then mode = bus
- **Rule 4** : If Travel cost/km is cheap and gender is female and she owns no car then mode = bus
- **Rule 5** : If Travel cost/km is cheap and gender is female and she owns 1 car then mode = train

From ID3 to C4.5

- Ross Quinlan started with
  - ID3 (Quinlan, 1979)
  - C4.5 (Quinlan, 1993)
- Some assumptions in the basic algorithm
  - All attributes are nominal
  - We do not have unknown values

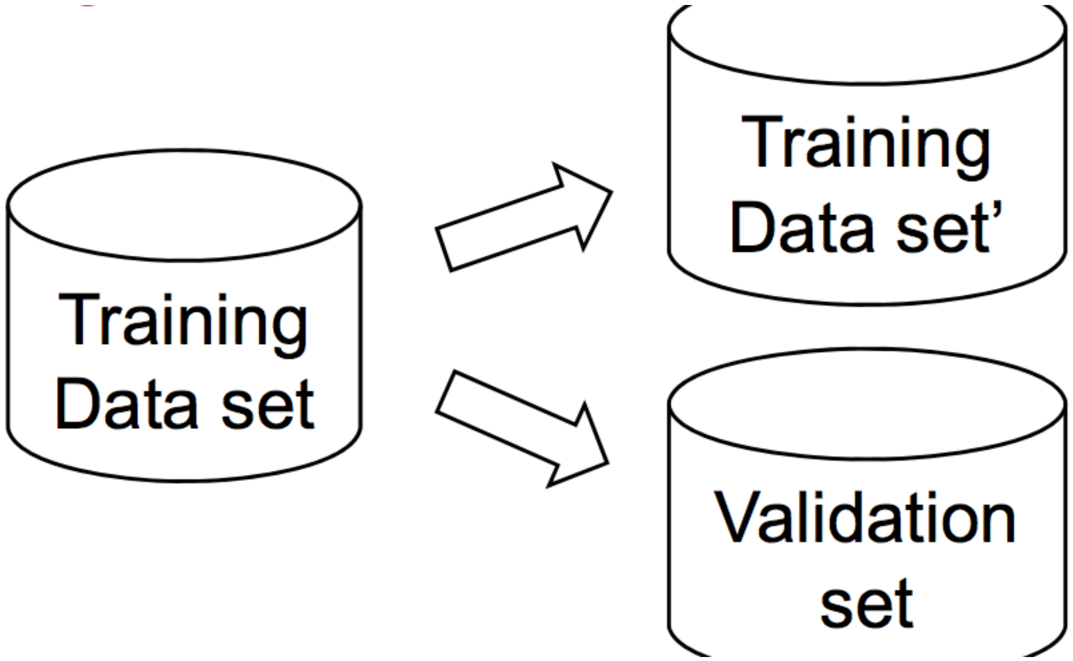
C4.5 algorithm

- Avoid overfitting
- Deal with continuous attributes
- Deal with missing data

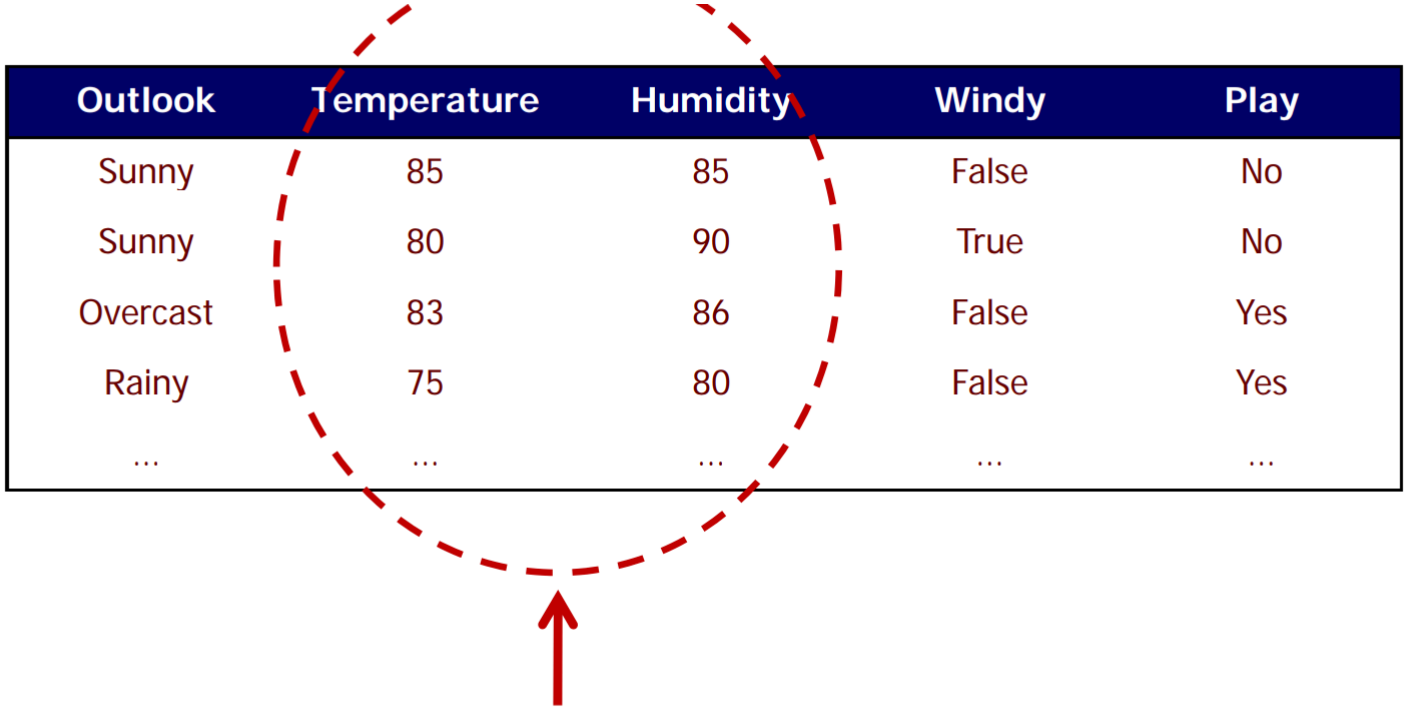
Pruning

1. Pre-prune: Stop growing a branch when information becomes unreliable

2. Post-prune: Take a fully-grown decision tree and discard unreliable parts



Dealing with continuous attributes



Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	False	No
Sunny	80	90	True	No
Overcast	83	86	False	Yes
Rainy	75	80	False	Yes
…	…	…	…	…

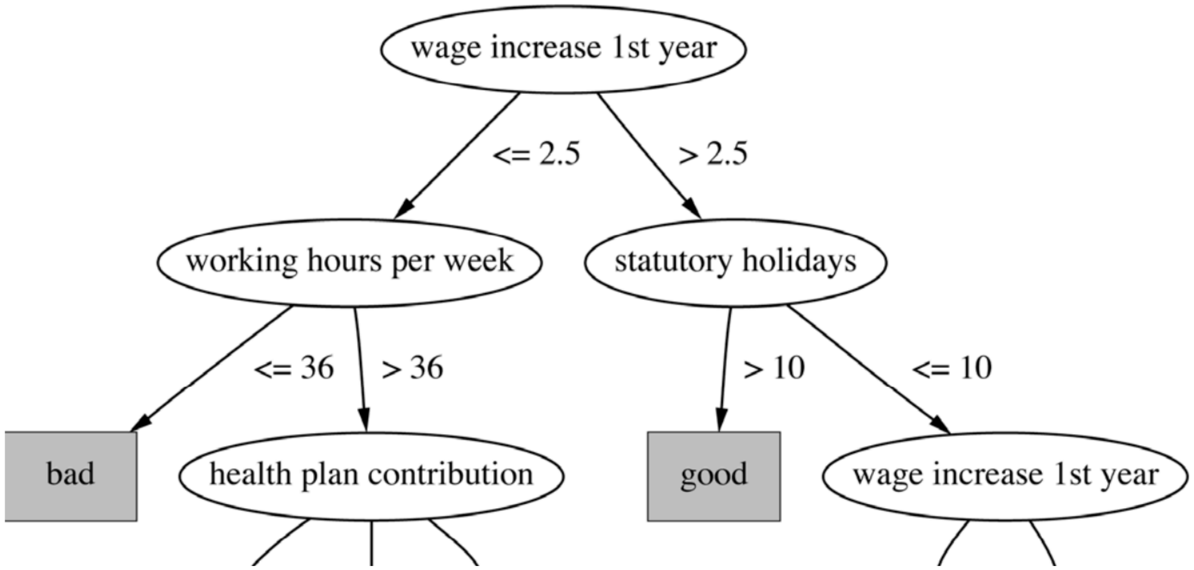
Continuous att

Split on temperature attribute:

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Yes	No	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes	Yes	No

- E.g.: temperature < 71.5: yes/4, no/2
  temperature ≥ 71.5: yes/5, no/3

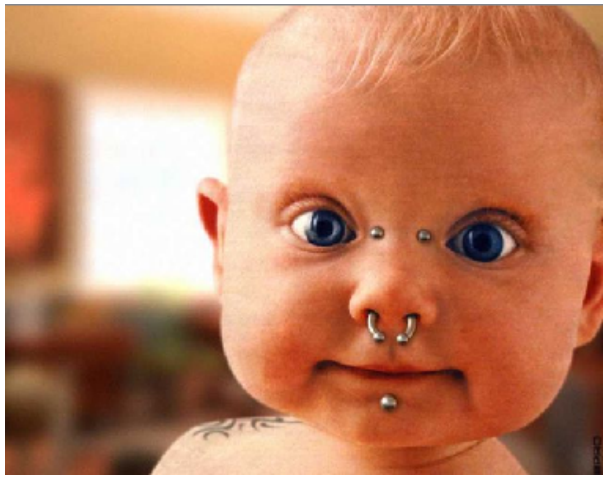
- Info([4,2],[5,3]) = 6/14 info([4,2]) + 8/14 info([5,3]) = 0.939 bits



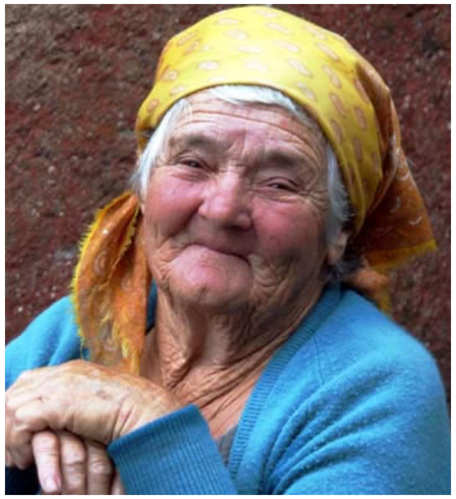
Intuition about Margin

Infant

?

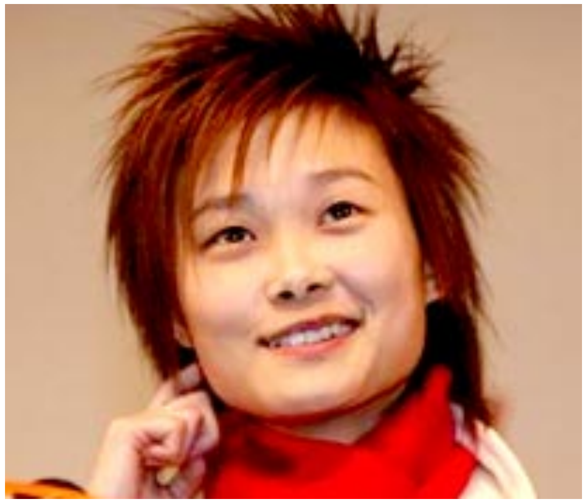


Elderly



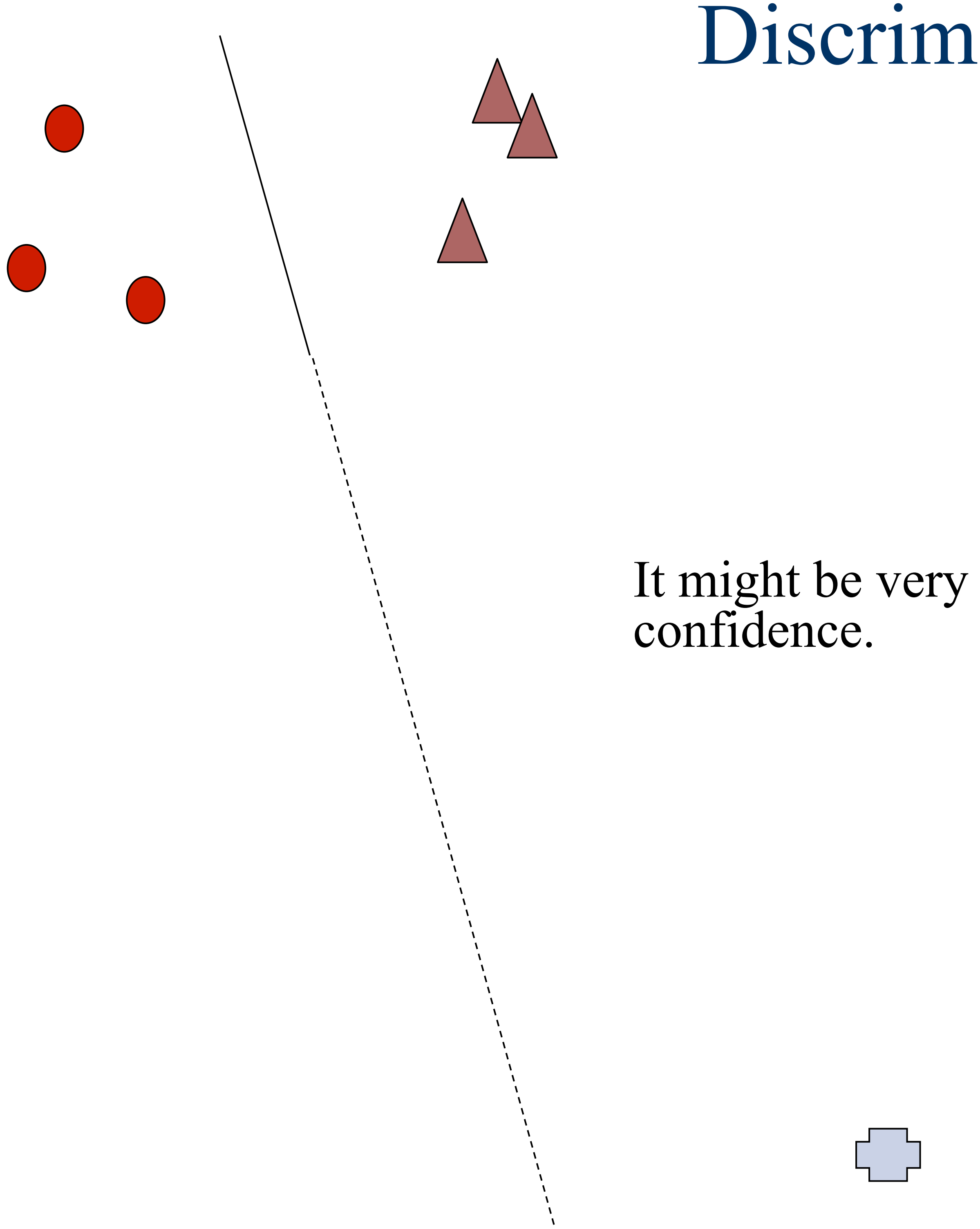
Man

?



Woman

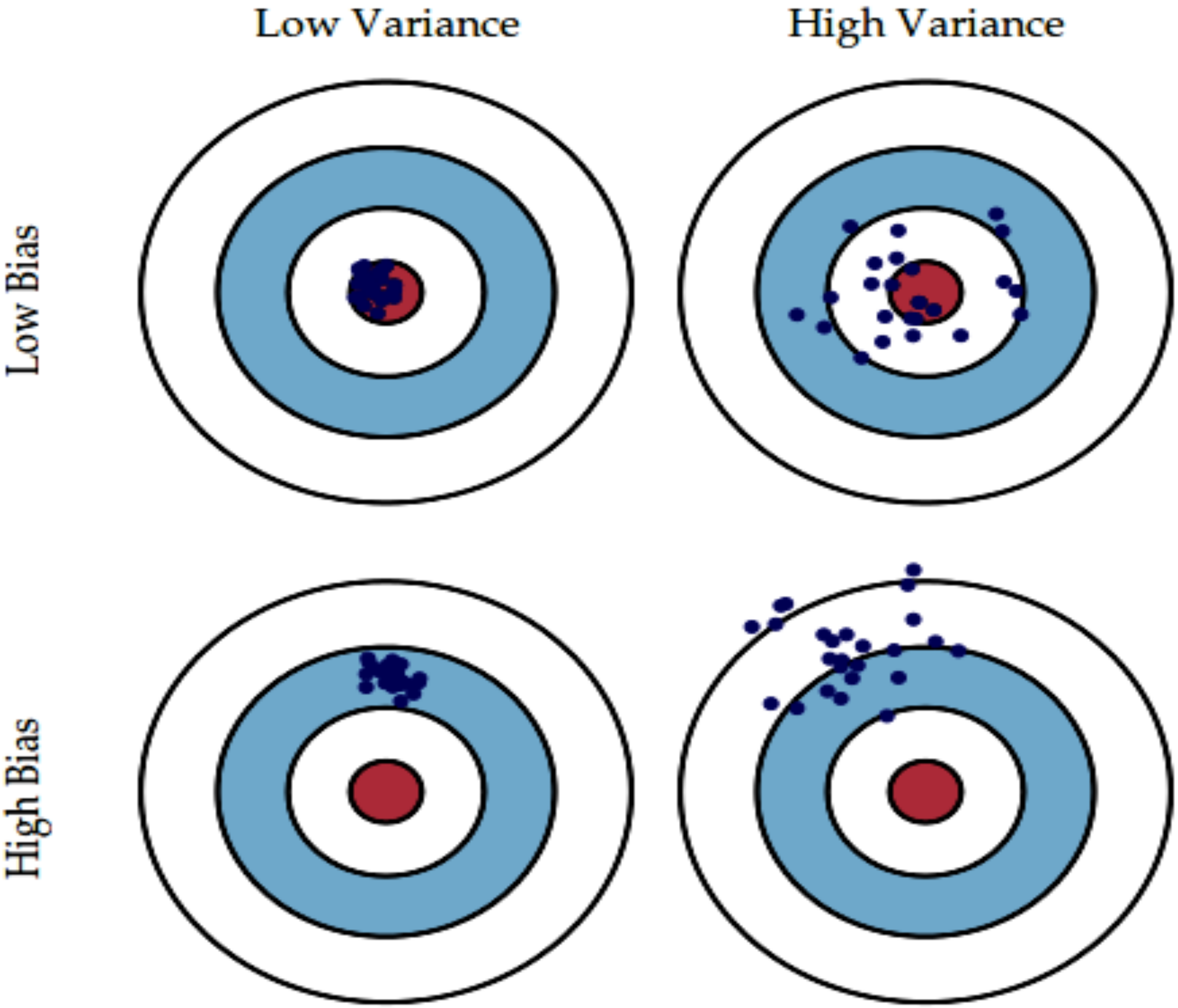
Problem with All Margin-based Discriminative Classifier



It might be very miss-leading to return a high confidence.

From Decision Tree to Random Forest

- Ensemble Learning
  - Average out biases
  - Reduce the variance
  - Unlikely to overfit


Low Variance
High Variance
Low Bias
High Bias

Bias-variance Decomposition

- For any learning scheme,
  - Bias = expected error of the combined classifier on new data
  - Variance = expected error due to the particular training set used
- Total expected error ~ bias + variance

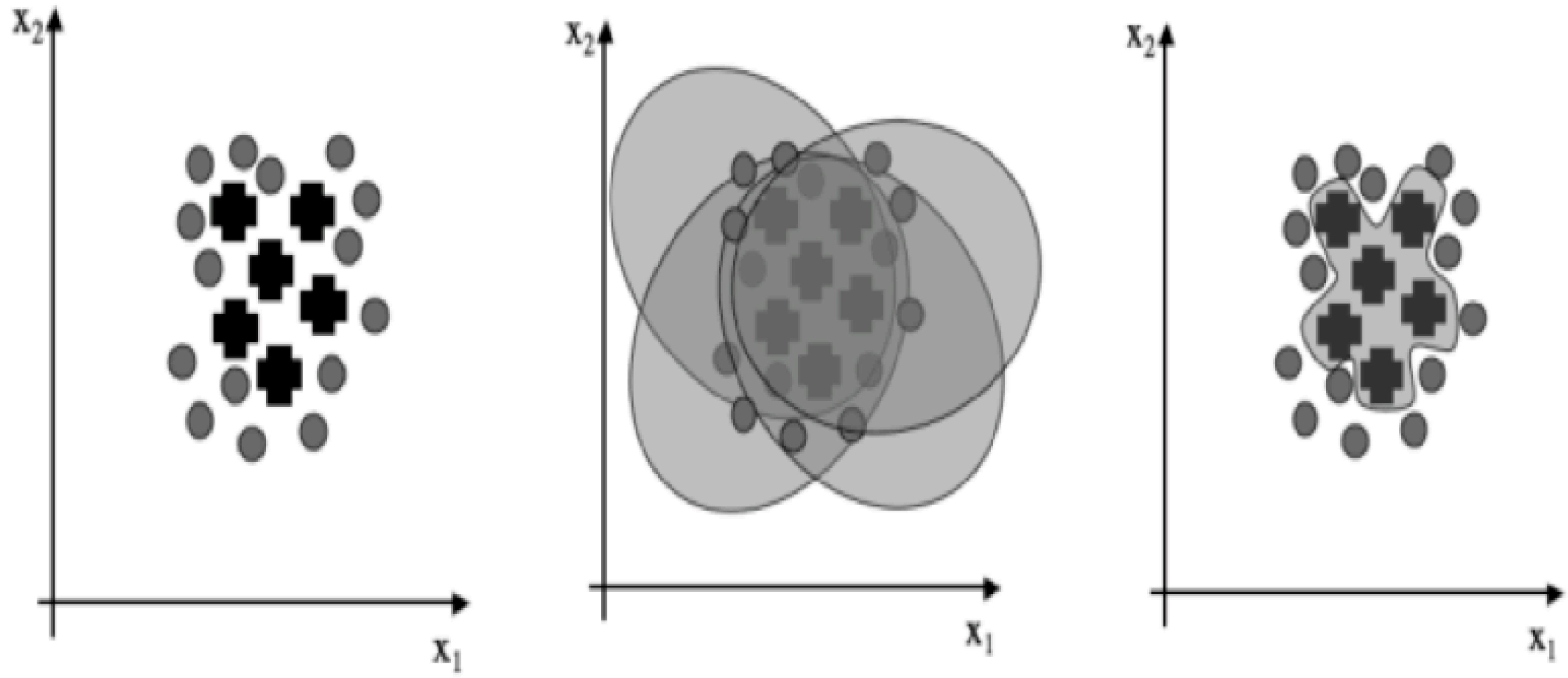
Ensemble Methods

Bagging $\left(\text{Breiman 1994,}\ldots\right)$

Boosting $\left(\text{Freund and Schapire 1995, Friedman et al. 1998,}\ldots\right)$

Random forests $\left(\text{Breiman 2001,}\ldots\right)$

Predict class label for unseen data by aggregating a set of predictions (classifiers learned from the training data).


x_2
x_1
x_2
x_1
x_2
x_1

Bagging

- Bootstrap data sets:
- Original data set: **X = {x 1 , x 2 , ..., x N }**.
- Creation of a new data set **XB** : draw N points at random from **X**, with **replacement**, so that some points in **X** may be replicated in **XB** (where as other points may be absent from **XB**).

Bagging

λTraining **the same mode**l on M multiple bootstrap data sets.

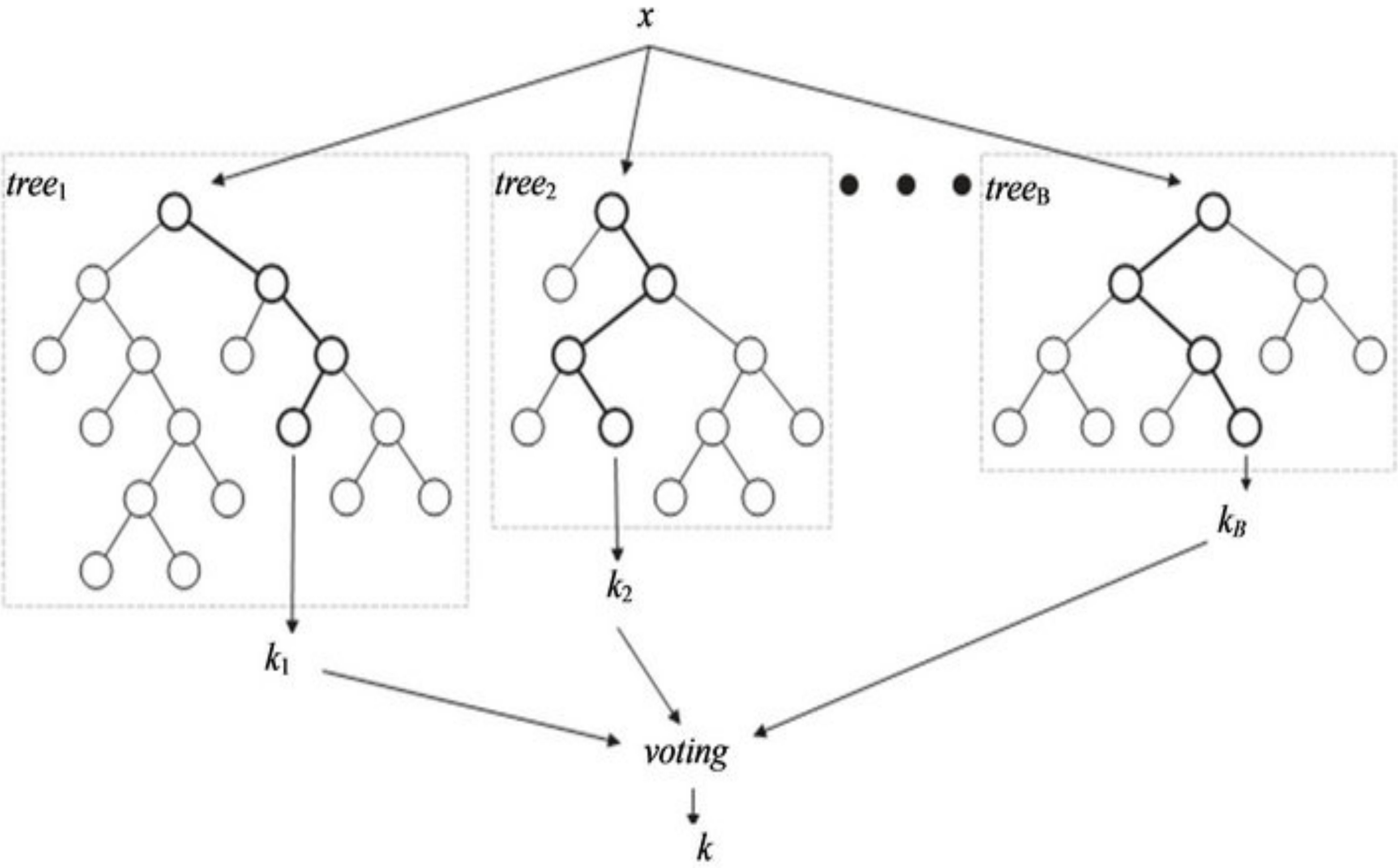
When does Bagging work?

- Learning algorithm is unstable: if small changes to the training set cause large changes in the learned classifier.
- If the learning algorithm is unstable, then Bagging almost always improves performance

Why Bagging works?

- Let $S = \{(y_i, x_i), i = 1...N\}$ be the set of training dataset
- Let $\{S_k\}$ be a sequence of training sets containing a sub-set of S
- Let P be the underlying distribution of S.
- Bagging replaces the prediction of the model with the majority of the predictions given by the classifiers

$$\varphi_A(x, P) = E_S(\varphi(x, S_k))$$


x
tree$_1$
tree$_2$
tree$_B$
k_1
k_2
k_B
voting
k

The Basic Random Forest Training Algorithm

- For each of N trees:
  - create a new bootstrap sample of the training set
  - use this bootstrap sample to train a decision tree
  - at each node of the decision tree, randomly select m features, and compute the information gain (or Gini impurity) only on that set of features, selecting the optimal one
  - repeat until the tree is complete

Probability Behind

- The probability of the ensemble getting the correct answer is a **binomial distribution**:

$$\sum_{k=T/2+1}^{T} \binom{T}{k} p^k (1-p)^{T-k},$$

where **p** is the success rate of each base classifier, and **T** is the number of base classifiers.

Random Forest Power

- The power of ensemble learning: if **p** > 0.5 then the correctness probability approaches 1 as **T** → ∞ .

Random Forest - Pros

λRequire almost no input preparation.

λPerform implicit feature selection

λVery quick to train.

λPretty tough to beat.

λIt’s really hard to build a bad Random Forest!

Random Forest

- Drawbacks?
  - Model size
  - Black boxes

Explainable Random Forest

λGet back to the case study of Flight Delay prediction

λ What made the flight delay?

Explanable Random Forest

- ***Input*:** A selected flight in future (a flight from Changi to Tan Son Nhat for the next 48 hours by Singapore Airlines).
- ***Output*:** Delay prediction (Y/N)

Explanation

Arrival hour: 0.25467993054
Airline: 0.253308988692
Origin: 0.158077791536
Departure time: 0.1364141321
Destination: 0.105243518586
Duration: 0.0660441127126
Type: 0.0219200955523
Arrival DoW: 0.00245074824256
Departure DoW: 0.00186059074095
Operation Type: 9.12956600922e-08