Lecture 19: Decision Tree & Random Forest

COMP90049
Knowledge Technology

Sarah Erfani and Vinh Nguyen, CIS

Semester 2, 2017

Classification Example: Detecting Tax Fraud

Training data:

categorical categorical continuous class

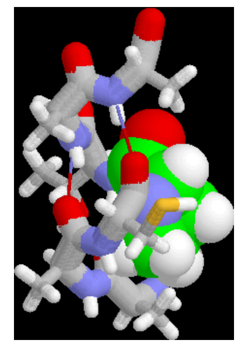
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Test data:

Tid	Refund	Marital status	Taxable Income	Cheat
11	Yes	Married	125K	?

More Examples of Classification Task

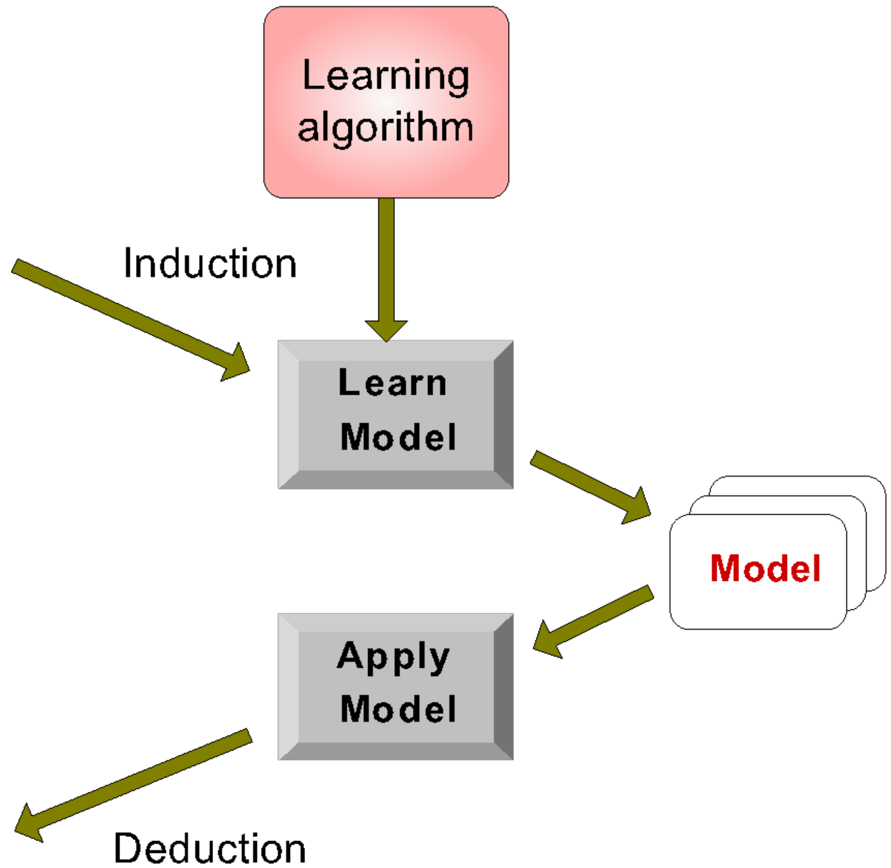
- Classifying credit card transactions as legitimate or fraudulent
- Predicting tumor cells as benign or malignant
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc
- Classifying Facebook posts into Families, Travel, Science & Tech …



Classification framework

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 Trees

- A flow-chart-like tree structure
- Internal node denotes a test on an attribute
- Branch represents an outcome of the test
- Leaf nodes represent class labels or class distribution

Decision Trees

- A flow-chart-like tree structure
- Internal node denotes a test on an attribute
- Branch represents an outcome of the test
- Leaf nodes represent class labels or class distribution

Advantages:

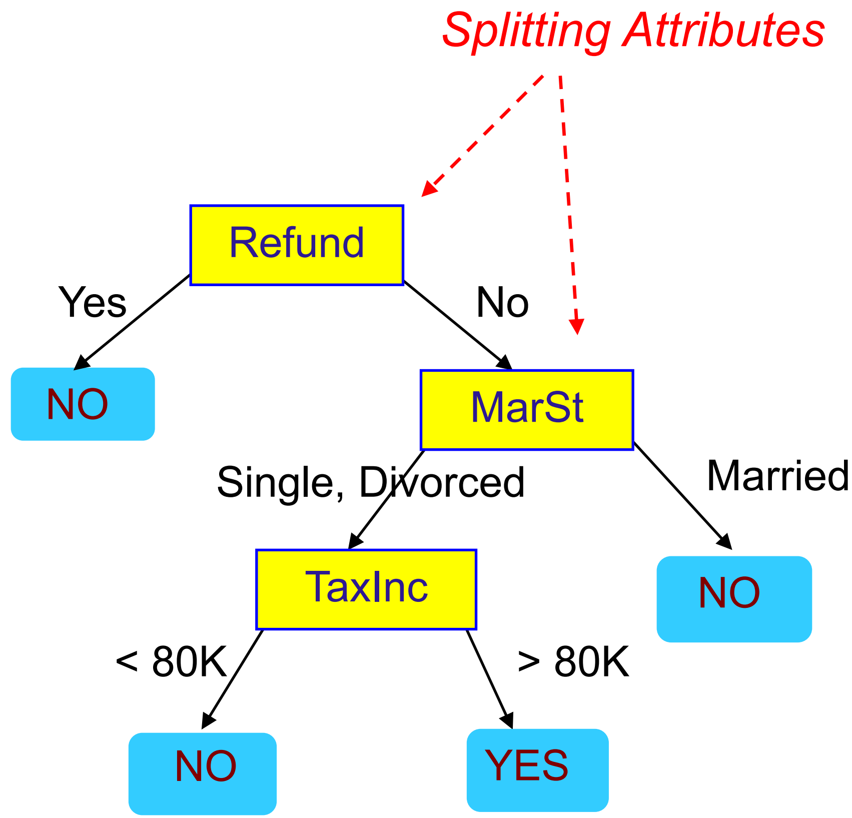
- Basic classification model
- Fast
- Scalable
- Interpretable

Disadvantage:

- Not highest accuracy (Random Forest to the rescue)

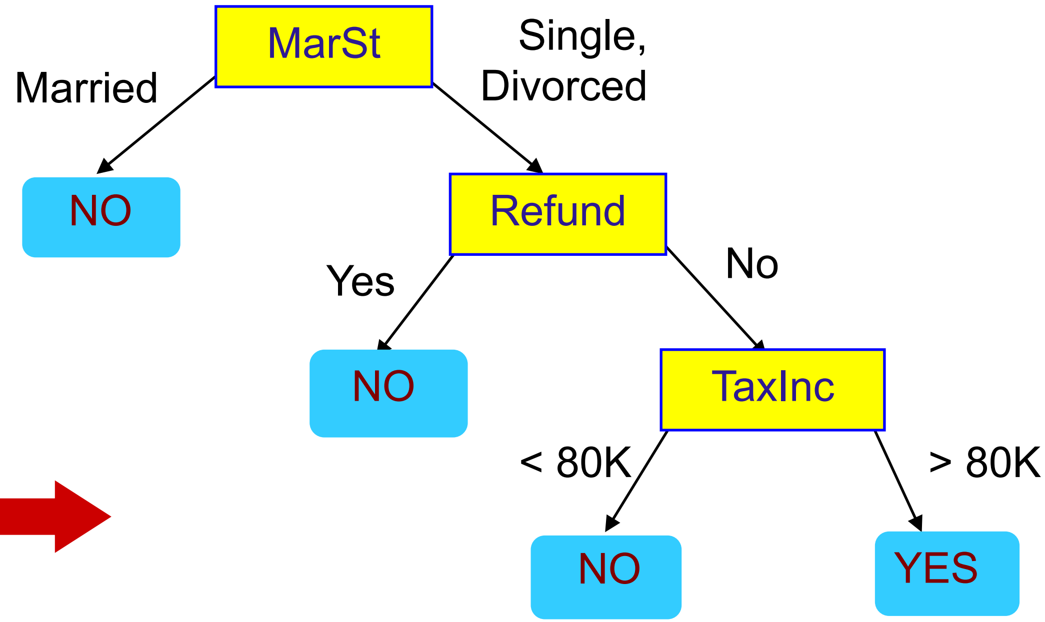
Example of a Decision Tree: Tax Fraud Detection

Tid	Refund	Marital Status	Taxable Income	Cheat
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**



Another Example of Decision Tree for Tax Fraud Detection

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
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**

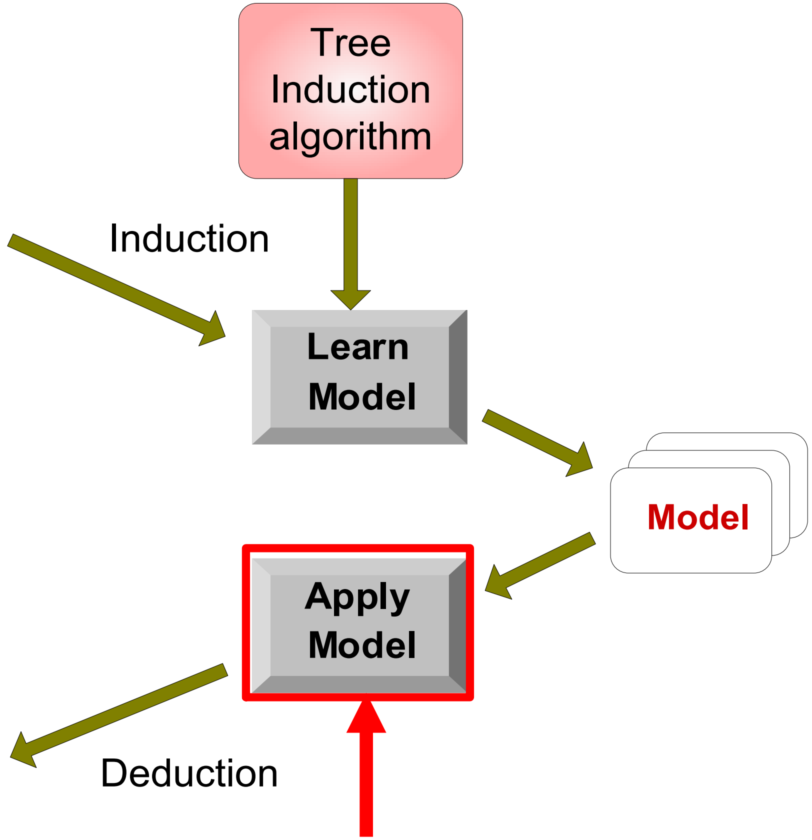


There could be more than one tree that fits the same data!

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

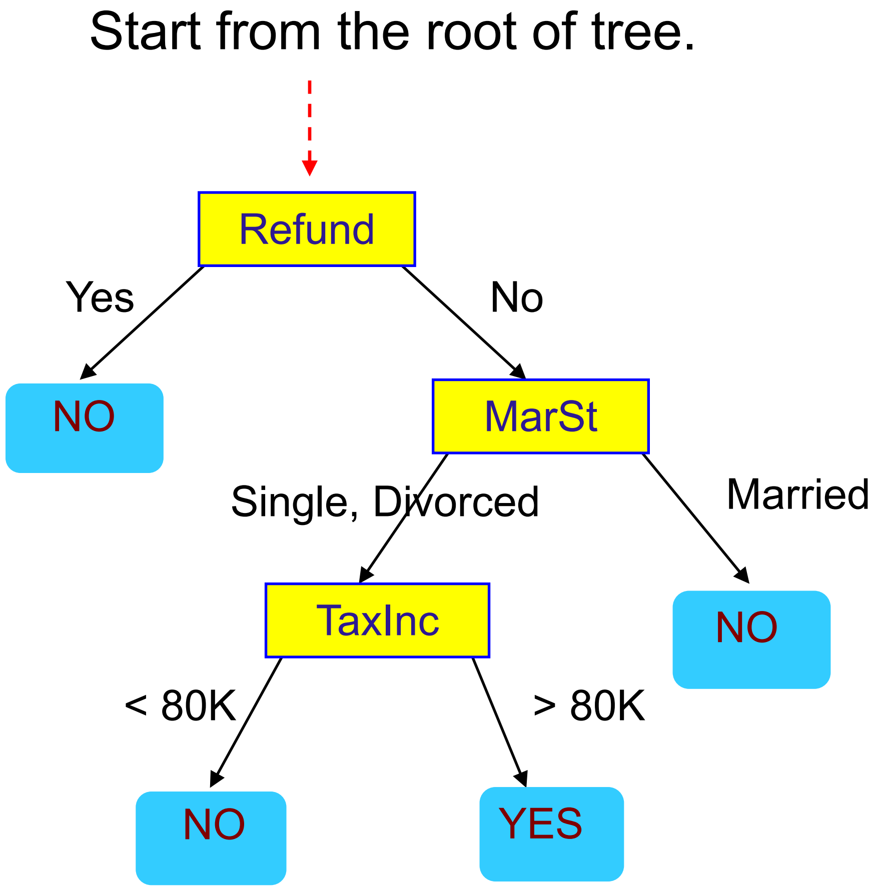


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

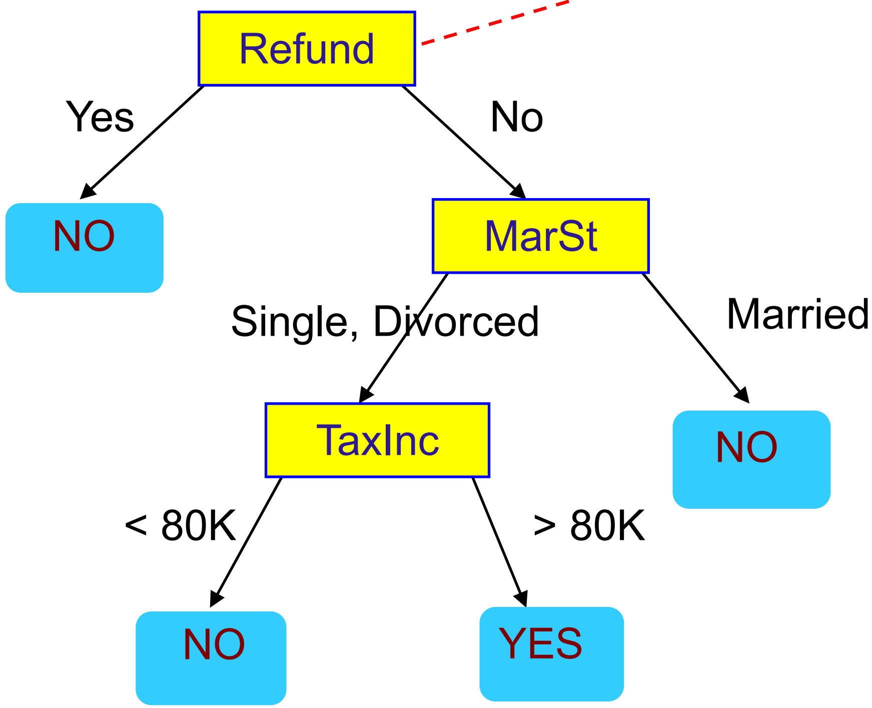
Decision Tree

Apply Model to Test Data



Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

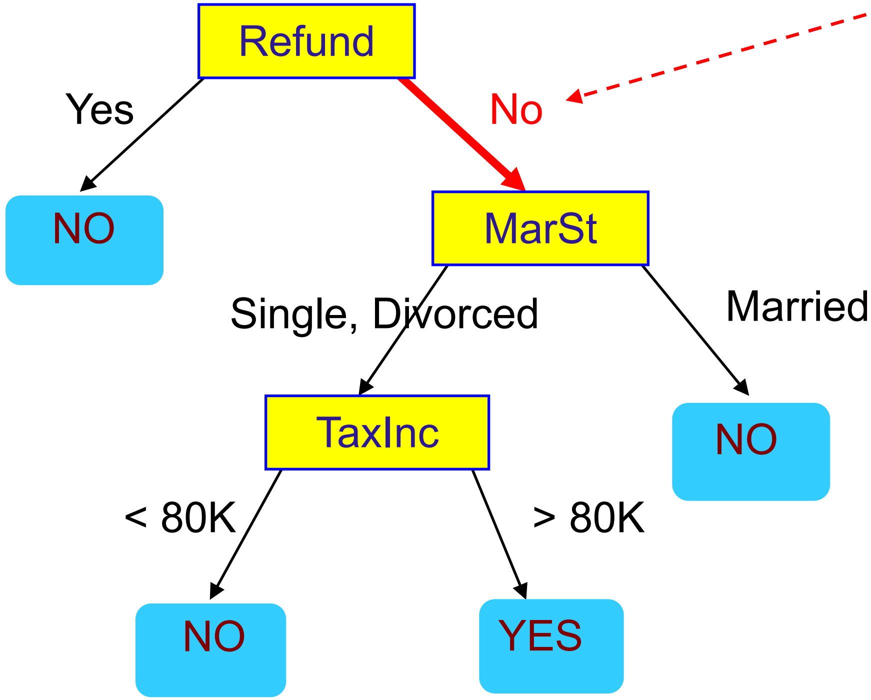
Apply Model to Test Data



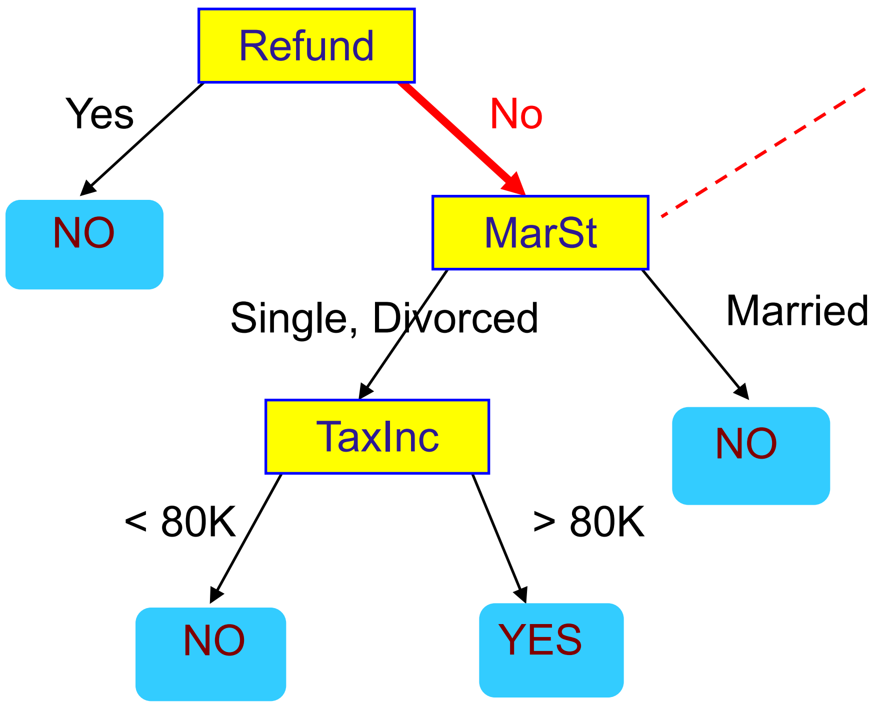
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

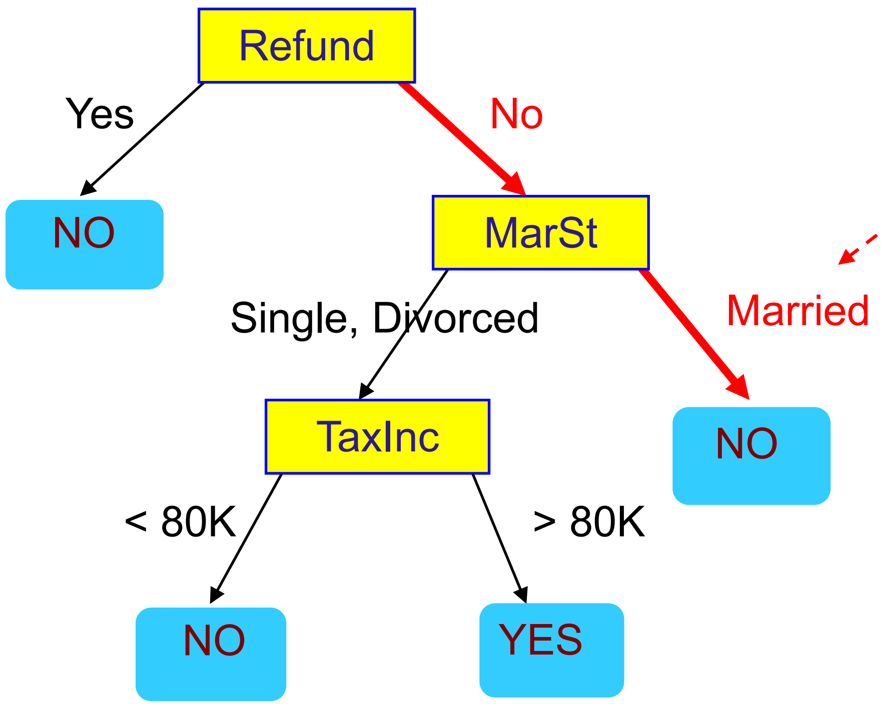


Apply Model to Test Data



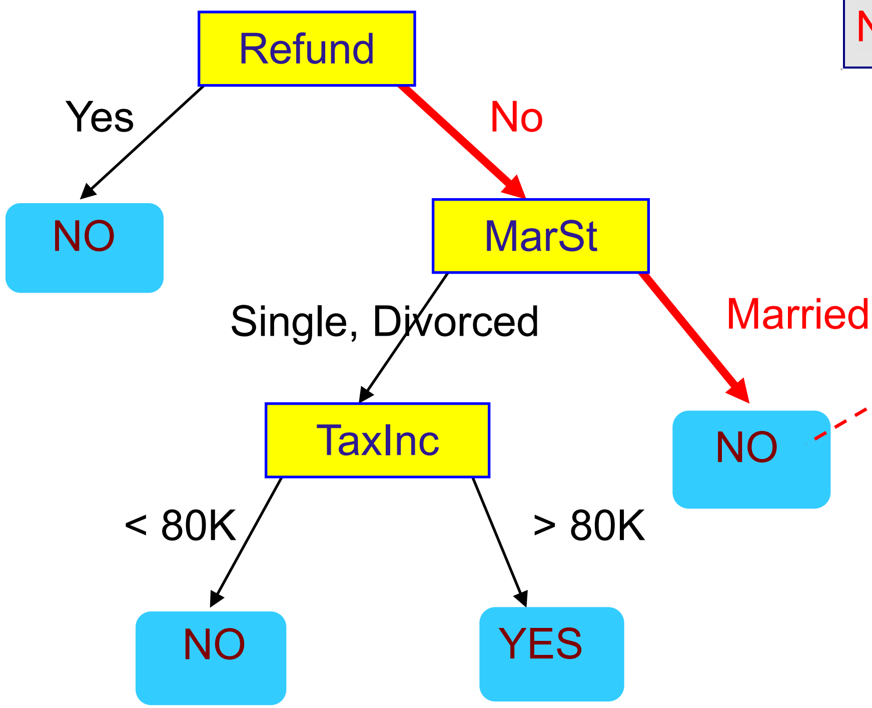
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

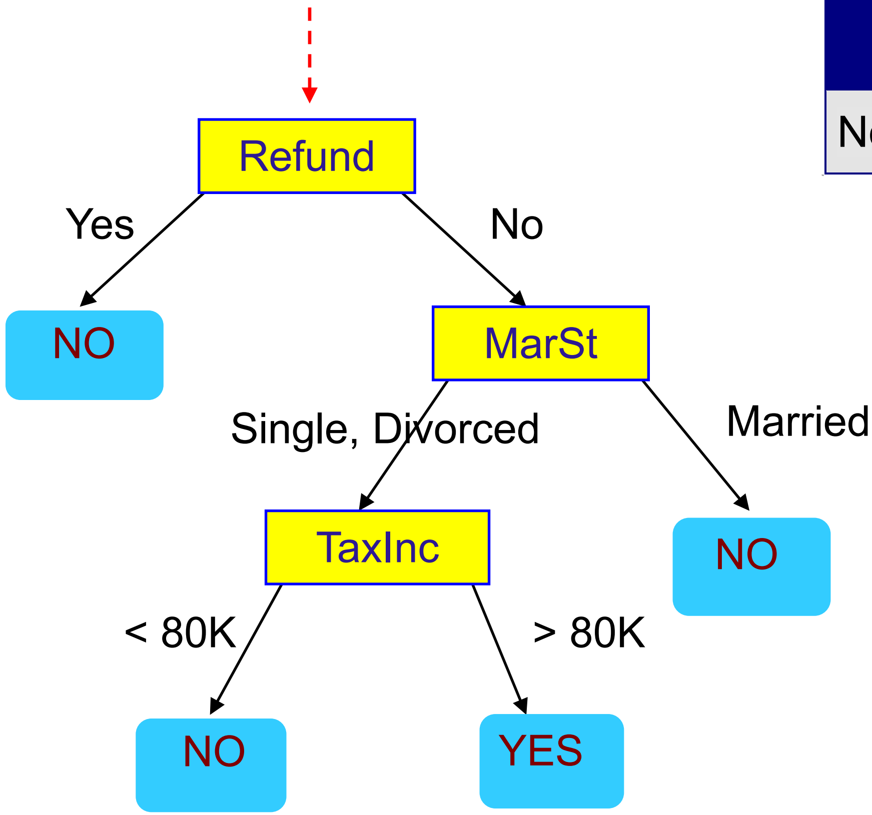


Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to “No”

Apply Model to Test Data

Start from the root of tree.



Refund	Marital Status	Taxable Income	Cheat
No	Single	100K	?

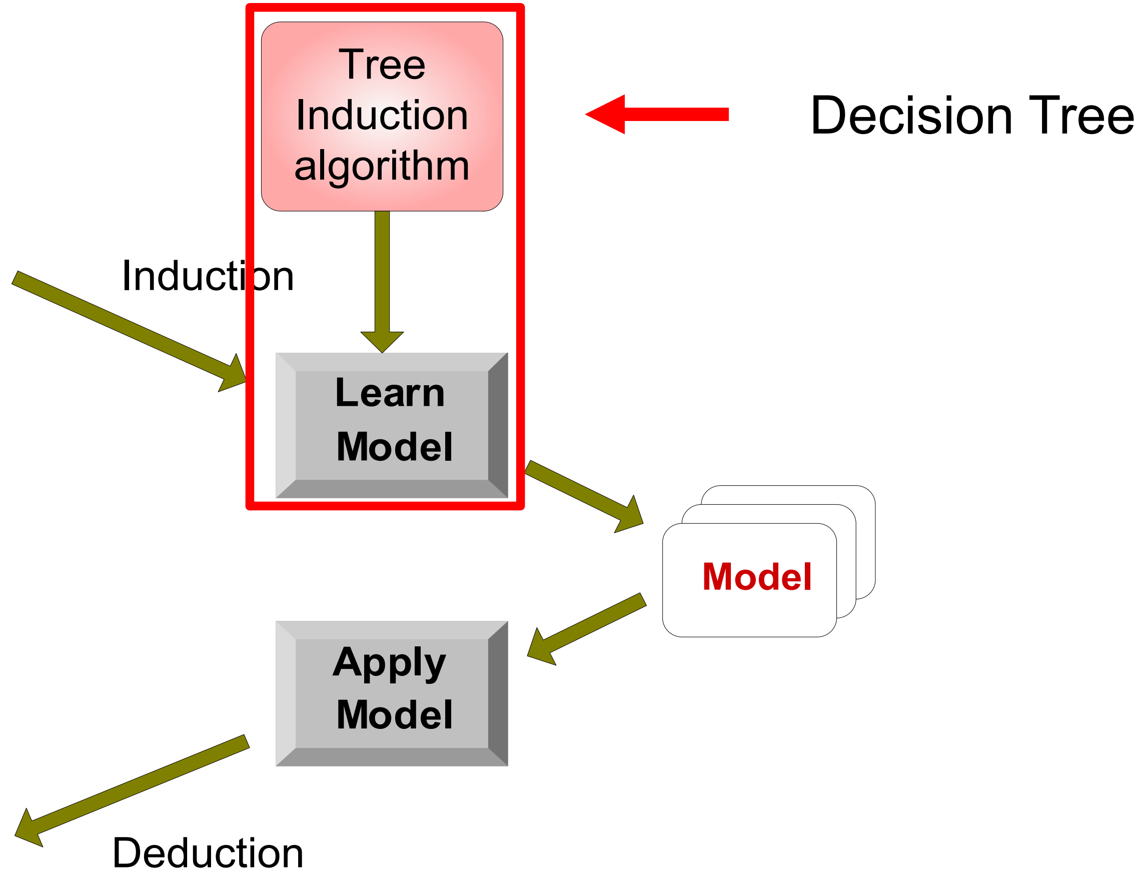
Decision Tree Classification Task

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



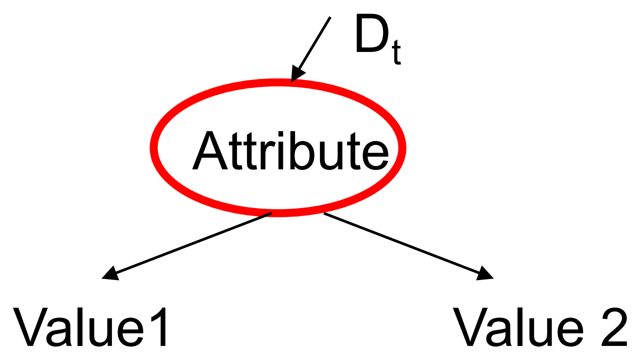
Decision Tree Induction: Hunt's Algorithm

Let D_t be the set of training records that reach a node t

Tid	Refund	Marital Status	Taxable Income	Cheat
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

General Procedure:

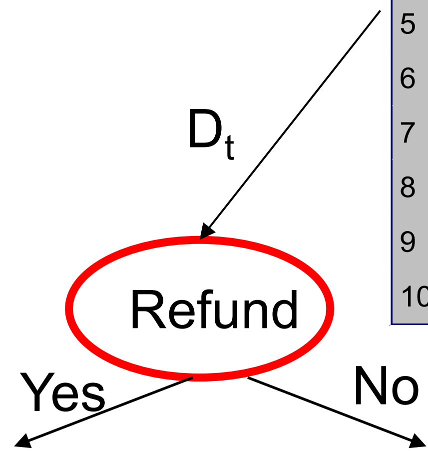
- If D_t contains records that **belong to more than one class**, select an attribute test to **split** the data into smaller subsets. Recursively apply the procedure to each subset.
- If D_t contains records that **belong the same class** y_t, then t is a leaf node labeled as y_t
- If D_t is an **empty set**, then t is a leaf node labeled by the default class, y_d (majority class in the data)



Example of Hunt's Algorithm

- If D_t contains records that belong to more than one class, use an attribute to split the data into smaller subsets.
- Recursively apply the procedure to each subset.

Tid	Refund	Marital Status	Taxable Income	Cheat
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

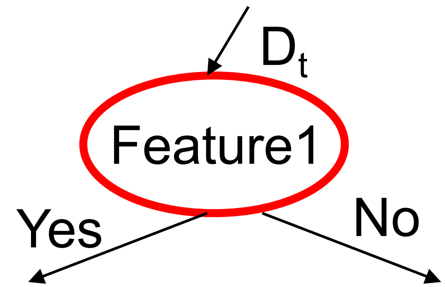


Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No

Tid	Refund	Marital Status	Taxable Income	Cheat
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Stopping Condition: Leaf Node

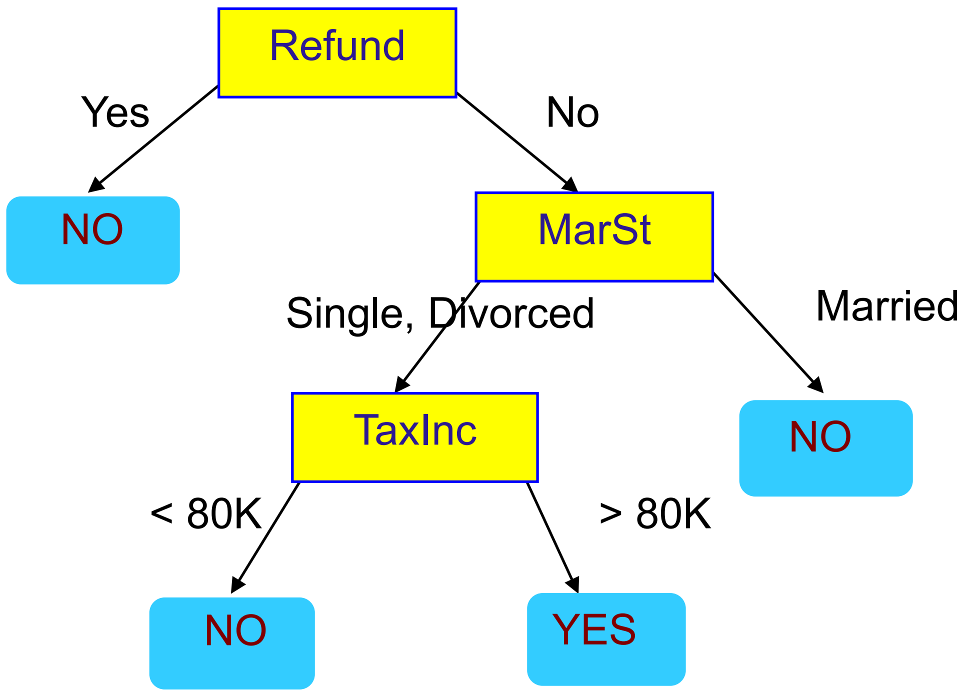
- If D_t contains records that belong the same class y_t, then t is a leaf node labeled as y_t
- If D_t is an empty set, then t is a leaf node labeled by the default class, y_d (majority class in the data)



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No

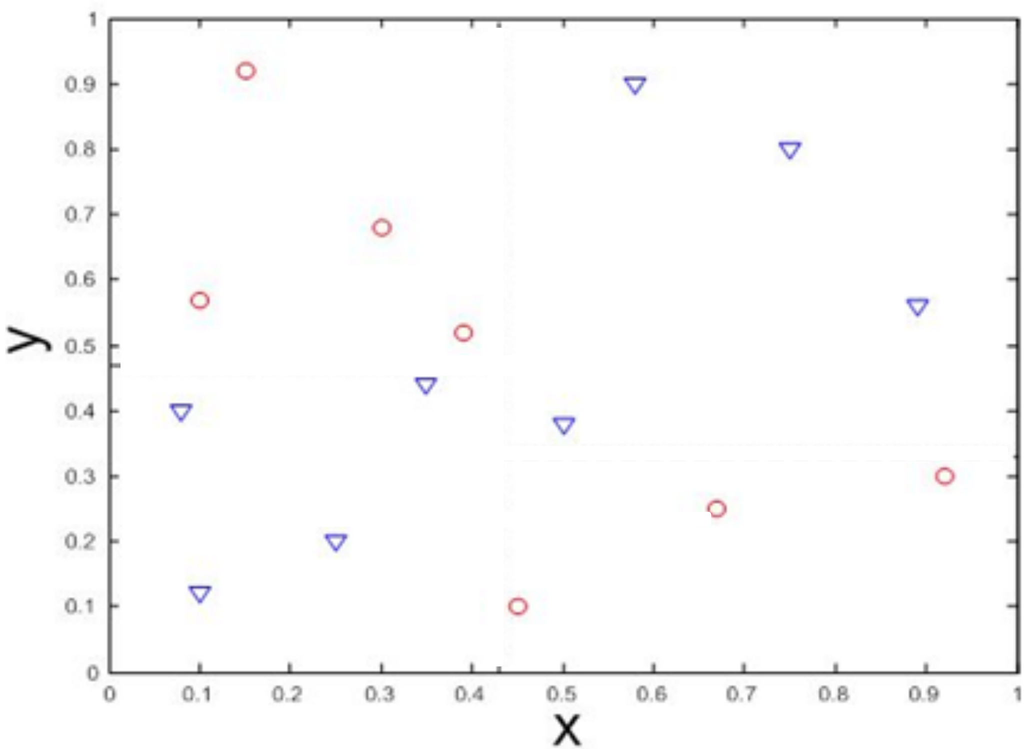
Decision Tree Model

- One possible model



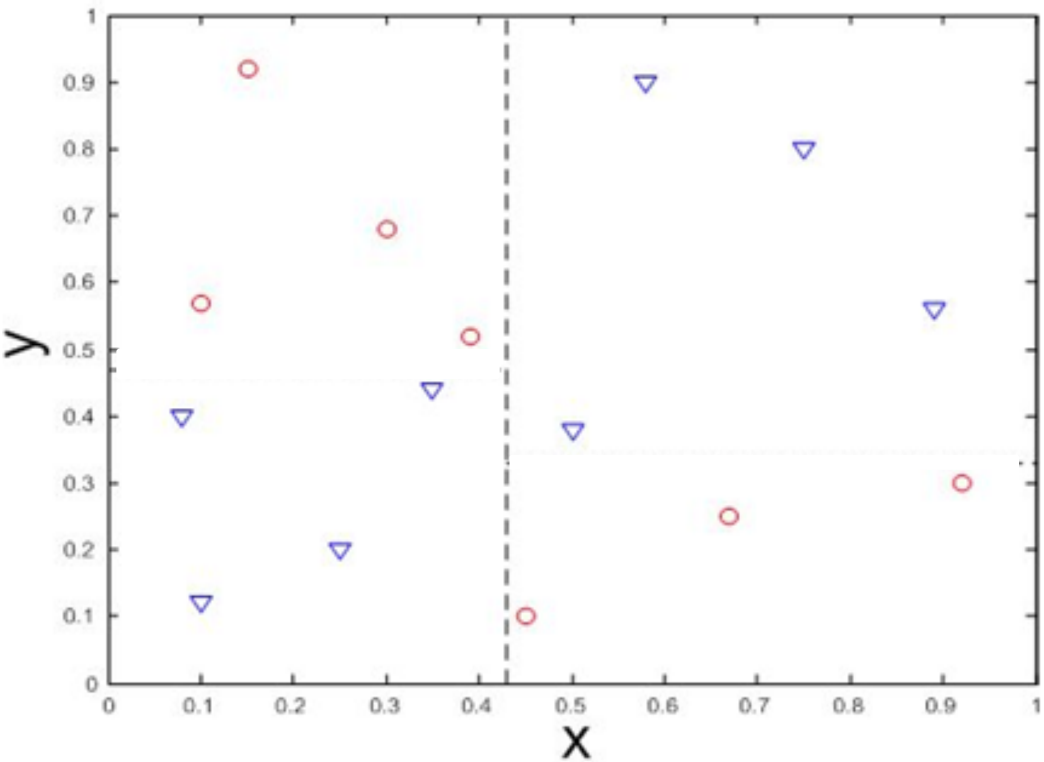
Decision Boundary for Decision Trees

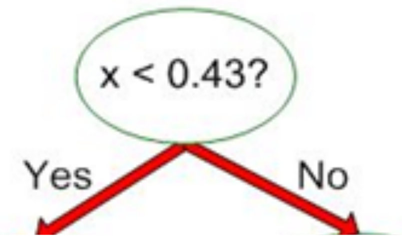
- Border line of between two neighbouring regions of different classes is known as decision boundary
- Decision boundary in decision trees is parallel to axes because test condition involves single attribute at-a-time



Decision Boundary for Decision Trees

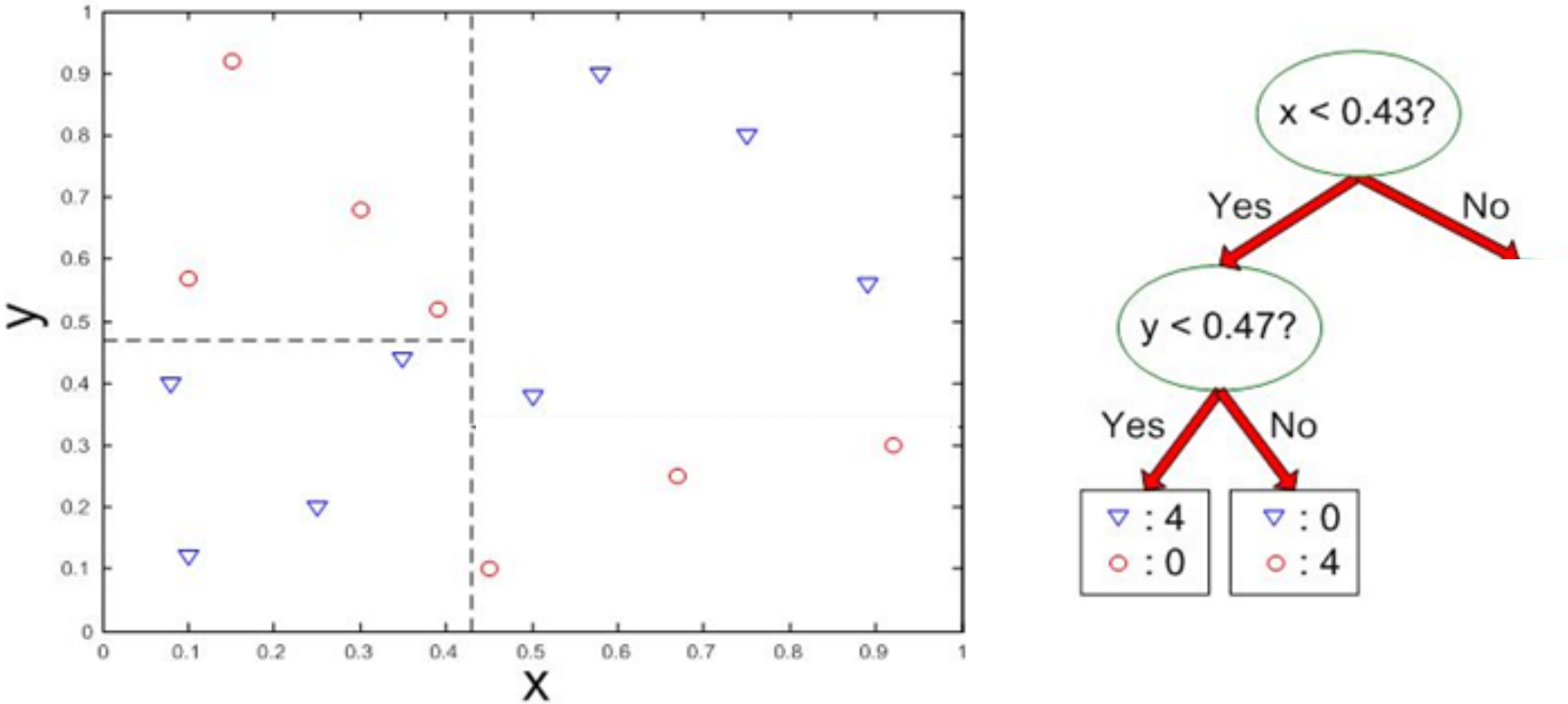
- Border line of between two neighbouring regions of different classes is known as decision boundary
- Decision boundary in decision trees is parallel to axes because test condition involves single attribute at-a-time





Decision Boundary for Decision Trees

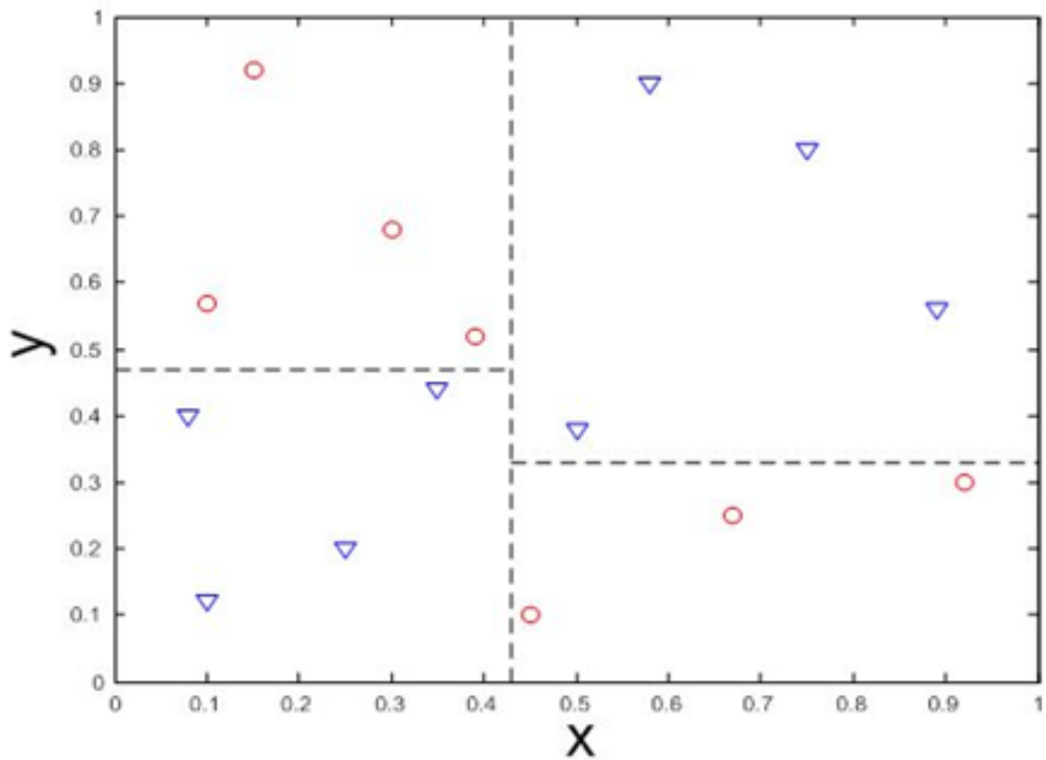
- Border line of between two neighbouring regions of different classes is known as decision boundary
- Decision boundary in decision trees is parallel to axes because test condition involves single attribute at-a-time

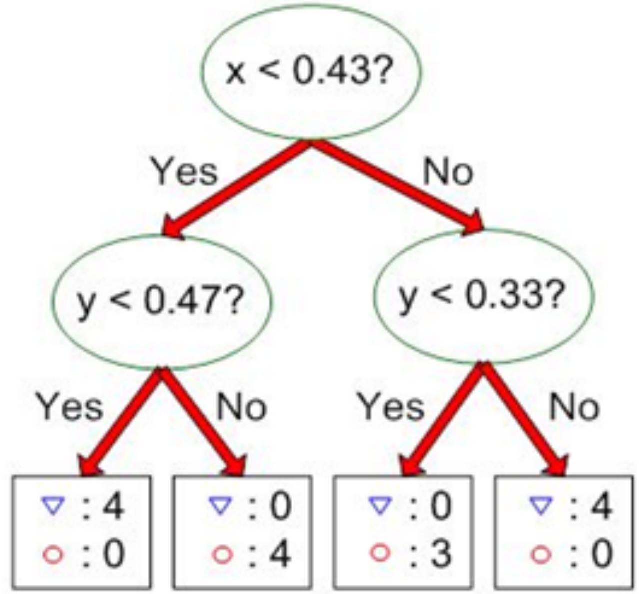


http://images.slideplayer.com/15/4848727/slides/slide_44.jpg

Decision Boundary for Decision Trees

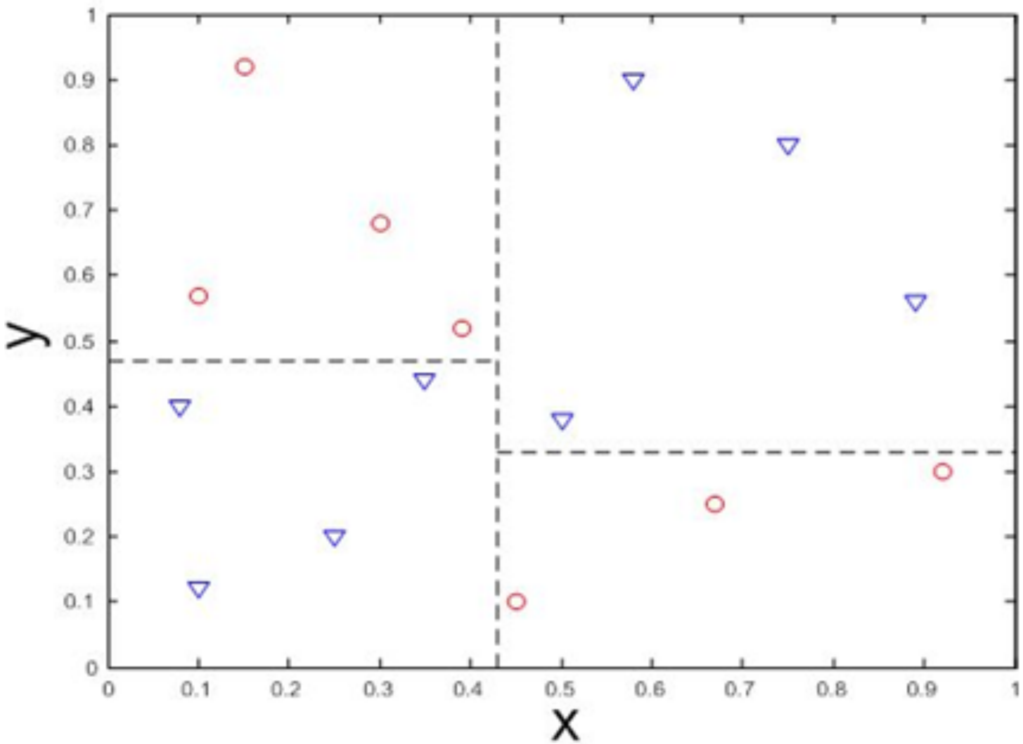
- Border line of between two neighbouring regions of different classes is known as decision boundary
- Decision boundary in decision trees is parallel to axes because test condition involves single attribute at-a-time

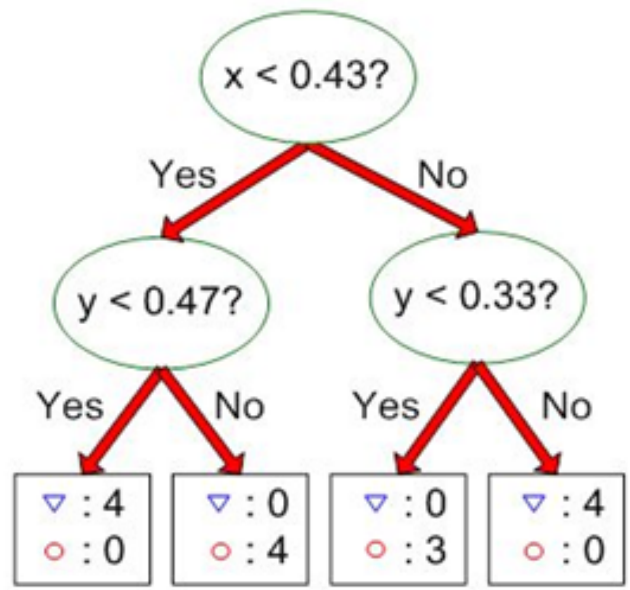




Hunt's algorithm: Intuition

- Recursively subdivide the training data into smaller subsets (sub-regions as axis-parallel rectangles)
- We would like each subset to be 'pure': label each rectangle with one class





Tree Induction Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

THE UNIVERSITY OF MELBOURNE

Tree Induction Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

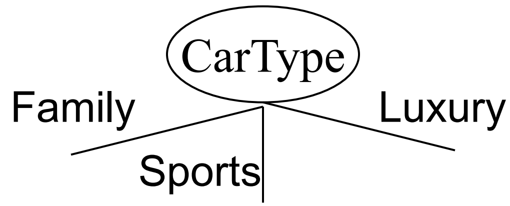
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

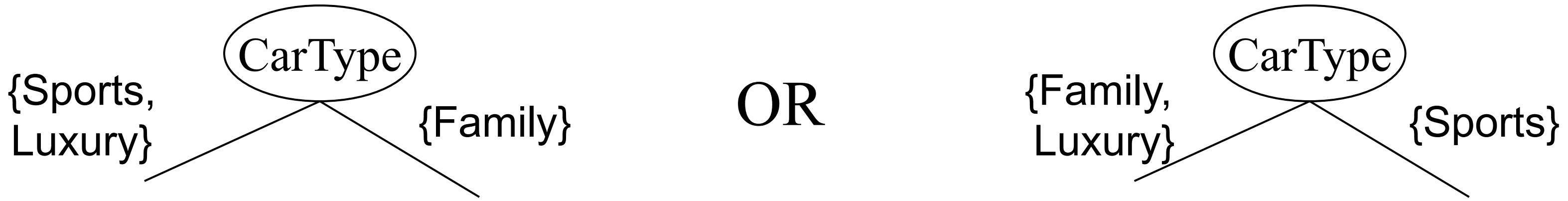
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

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

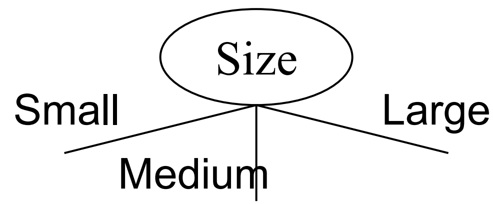


- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.

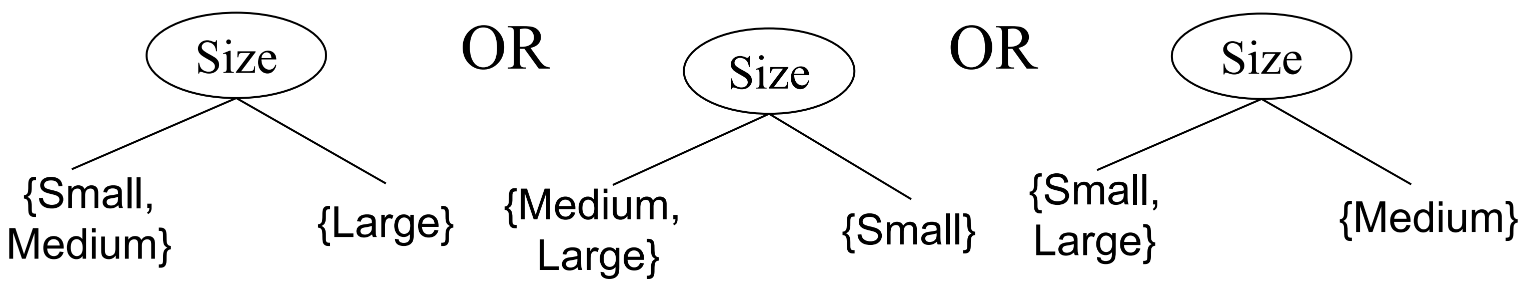


Splitting Based on Ordinal Attributes

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



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

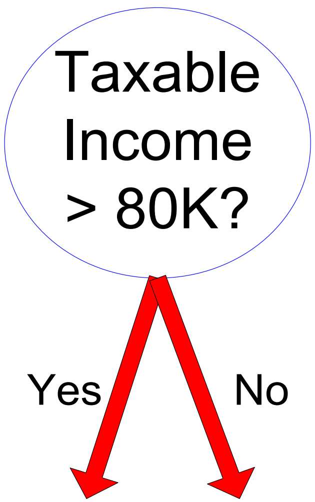


- Which split should we choose?

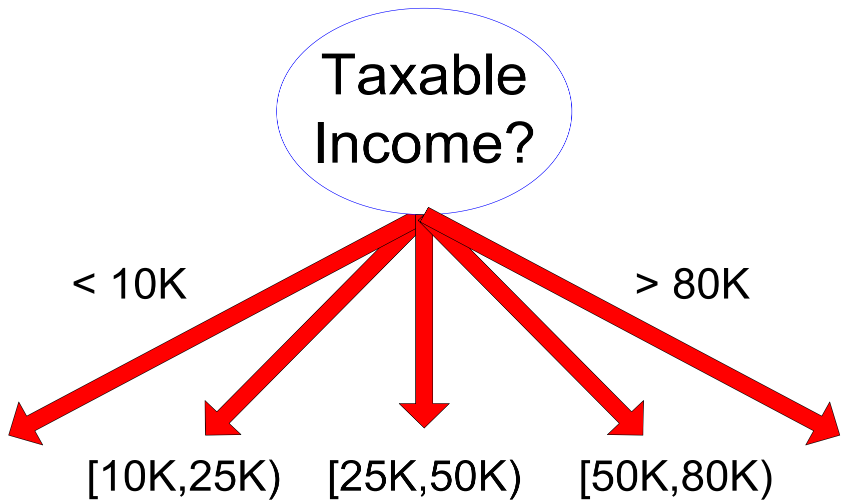
Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more compute intensive

Splitting Based on Continuous Attributes



(i) Binary split



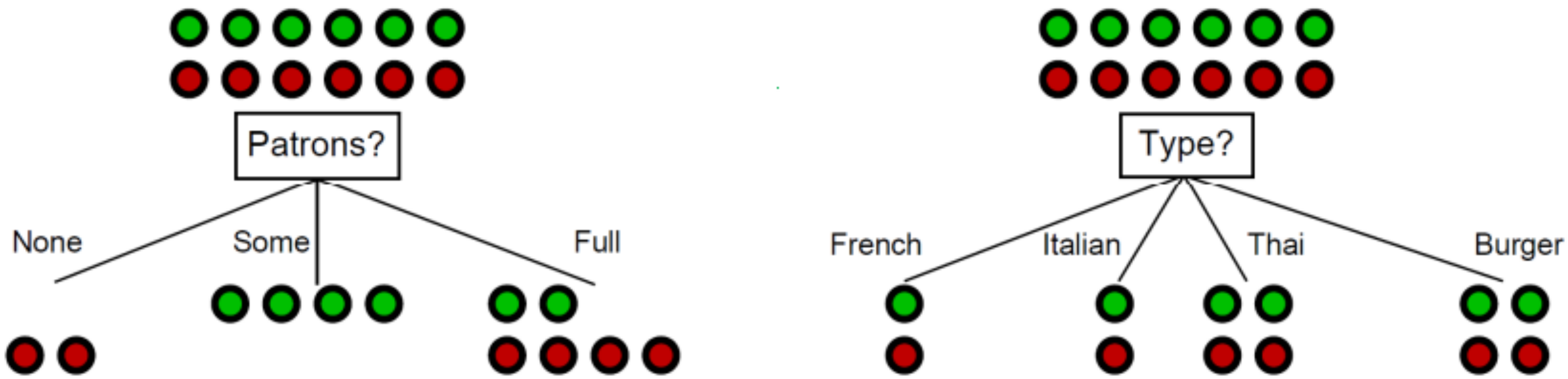
(ii) Multi-way split

Tree Induction Issues

- Determine how to split the records
  - How to specify the attribute test condition?
  - How to determine the best split?
- Determine when to stop splitting

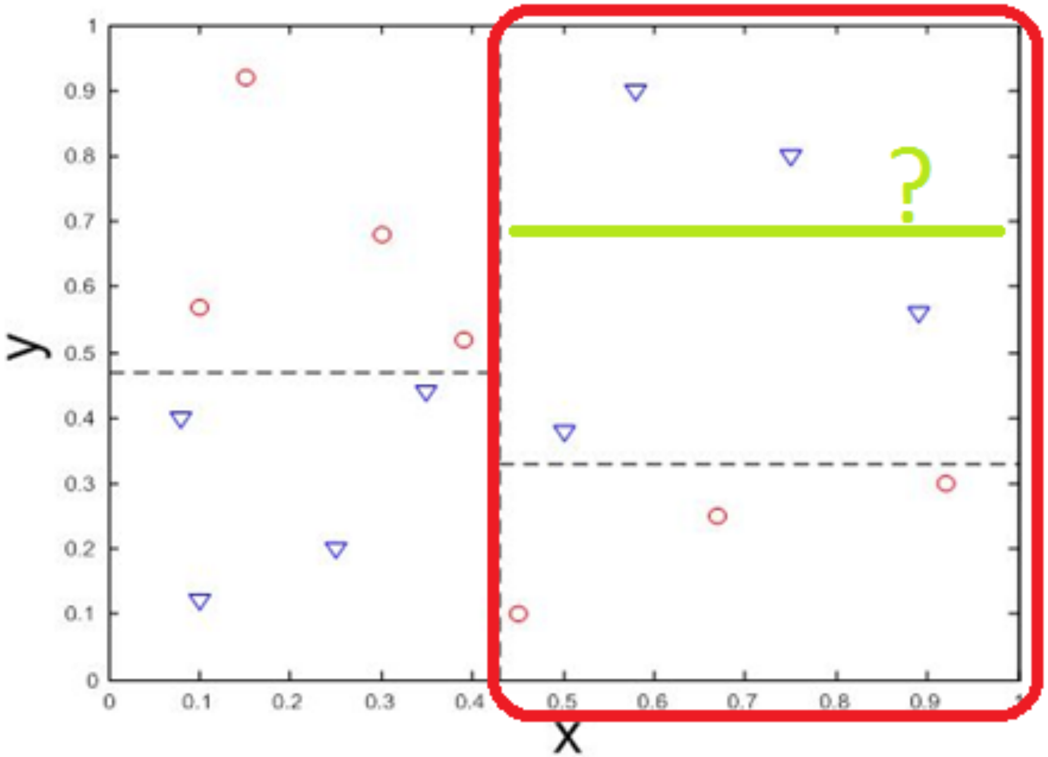
How to Determine the Best Split – Example 1

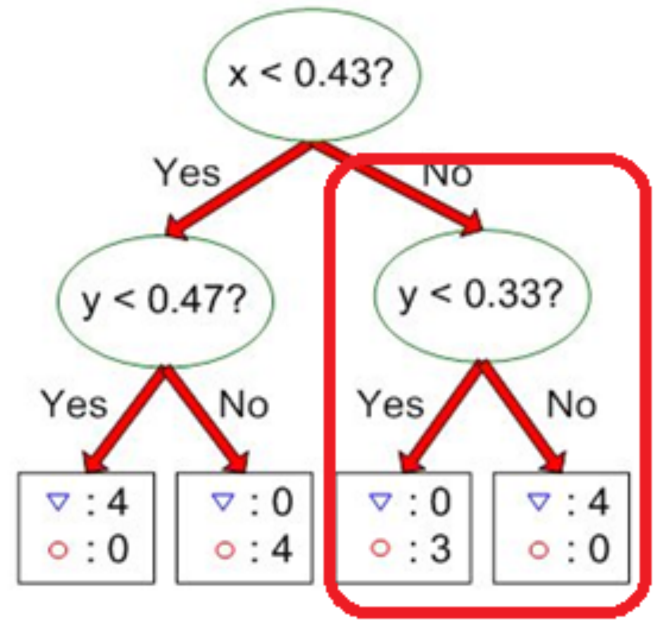
- Before Splitting: 6 records of class C0,
  6 records of class C1



Which test condition is the best?

How to Determine the Best Split – Example 2





How to Determine the Best Split

- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

C0: 5 C1: 5	C0: 9 C1: 1
Non-homogeneous, High degree of impurity	Homogeneous, Low degree of impurity

Measures of Node Impurity

- Misclassification Error
- Entropy
- Gini index

Node Impurity Based on Classification Error

- Classification error at a node t :

$$Error(t) = 1 - \max_i P(i \mid t)$$

- *P(i|t)*: proportion of points in the i-th class

- Measures misclassification error made by a node.
  - Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

Examples for Computing Error

$$Error(t) = 1 - \max_i P(i \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	**2**
C2	**4**

?

Examples for Computing Error

$$Error(t) = 1 - \max_i P(i \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Node Impurity Criteria Based on Entropy

- Entropy at a given node t:

$$Entropy(t) = -\sum_{j} p(j \mid t) \log p(j \mid t)$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Measures homogeneity of a node.
  - Maximum ($\log n_c$) when records are equally distributed among all classes implying least information, where n_c is the number of classes.
  - Minimum (0.0) when all records belong to one class, implying most information

Examples for computing Entropy

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

?

Examples for computing Entropy

$$Entropy(t) = -\sum_j p(j\,|\,t) \log p(j\,|\,t)$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Measure of Impurity: GINI

- Gini Index for a given node t :

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

  - Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information
  - Minimum (0.0) when all records belong to one class, implying most interesting information

C1	**0**
C2	**6**
Gini=0.000	

C1	**1**
C2	**5**
Gini=0.278	

C1	**2**
C2	**4**
Gini=0.444	

C1	**3**
C2	**3**
Gini=0.500	

Examples for computing GINI

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0 P(C2) = 6/6 = 1

Gini = 1 – P(C1)^2 – P(C2)^2 = 1 – 0 – 1 = 0

C1	**1**
C2	**5**

P(C1) = 1/6 P(C2) = 5/6

Gini = 1 – (1/6)^2 – (5/6)^2 = 0.278

C1	**2**
C2	**4**

P(C1) = 2/6 P(C2) = 4/6

Gini = 1 – (2/6)^2 – (4/6)^2 = 0.444

How good is a Split?

- Compare the impurity of parent node (before splitting)
- With the impurity of the children nodes (after splitting)

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

- $I(v_j)$: impurity measure of node v_j
- j: children node index
- $N(v_j)$: number of data points in child node v_j
- N: number of data points in parent node
- The larger the gain, the better

How good is a Split?

- For I(v) being the entropy function: Information gain

$$\Delta = I(\text{parent}) - \sum_{j=1}^{k} \frac{N(v_j)}{N} I(v_j),$$

- Where I() is the entropy function H()
- Note: the information gain is equivalent to the mutual information between the class variable and the test attribute
- Thus splitting using the information gain is to choose the attribute with highest information shared with the class variable

Splitting Based on Gain Ratio

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO} \qquad SplitINFO = -\sum_{i=1}^{k} \frac{n_i}{n} \log \frac{n_i}{n}$$

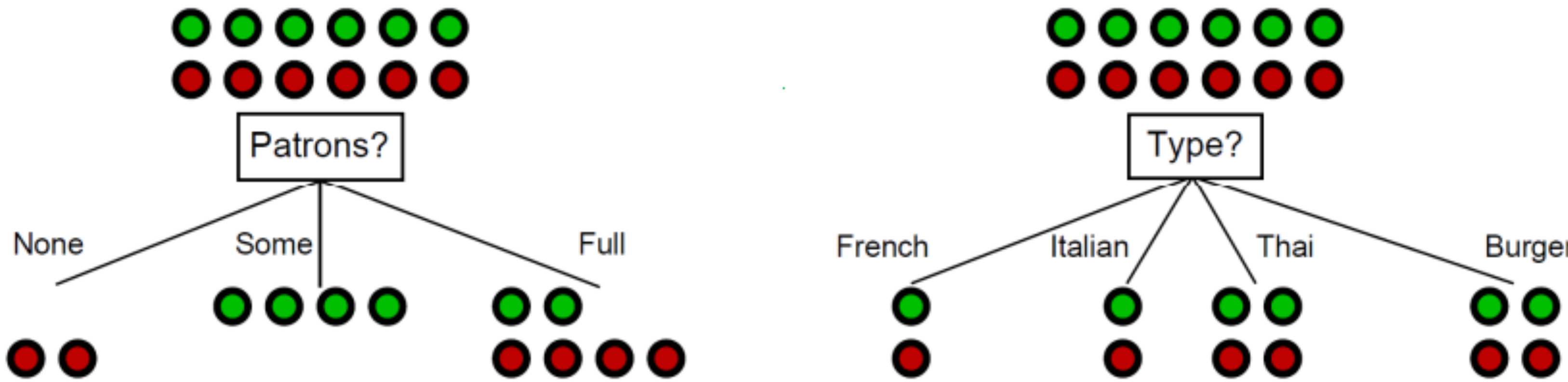
Parent Node, p is split into k partitions

n_i is the number of records in partition i

  – Adjusts Information Gain by the entropy of the partitioning (SplitINFO). Higher entropy partitioning (large number of small partitions) is penalized!
  – Used in C4.5
  – Designed to overcome the disadvantage of Information Gain

How to Determine the Best Split?

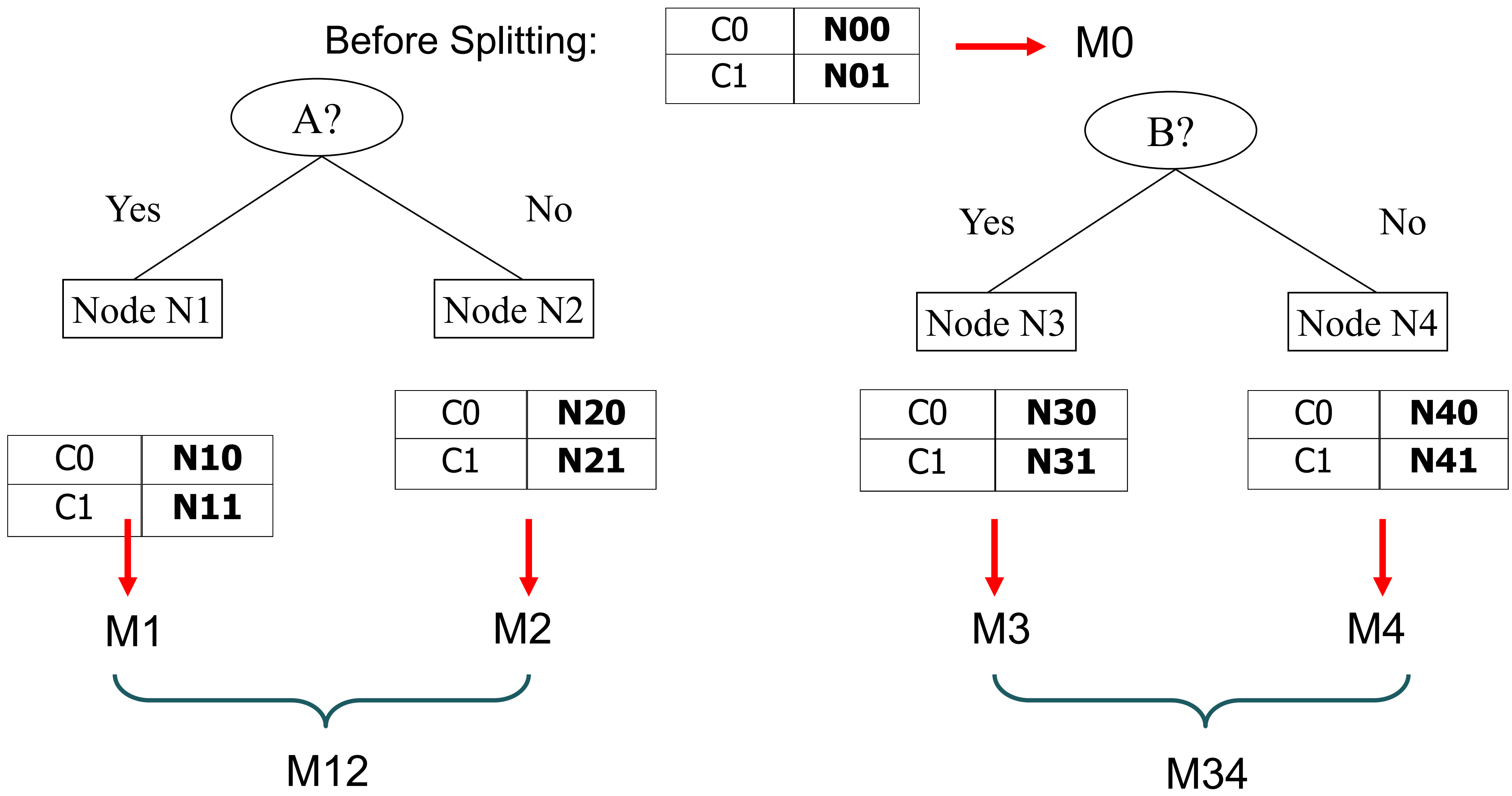
Before Splitting: 6 records of class 0,
6 records of class 1



Which test condition is the best?

- Compute the gain of all splits
- Choose the one with largest gain

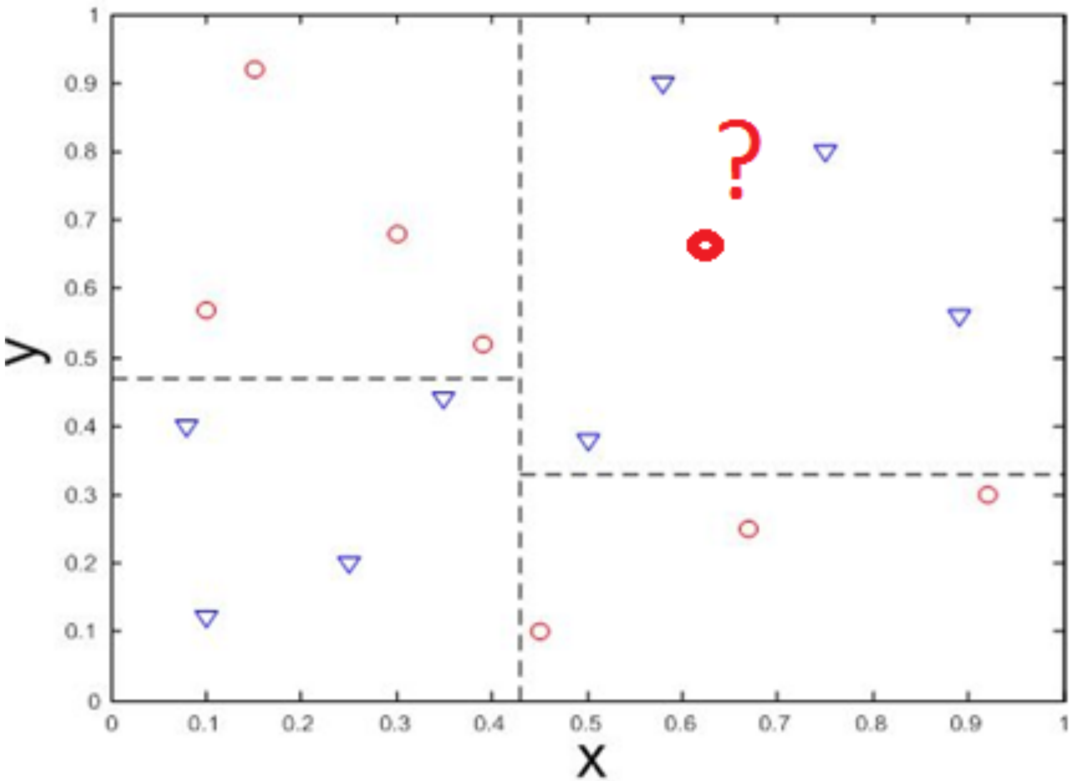
How to Find the Best Split

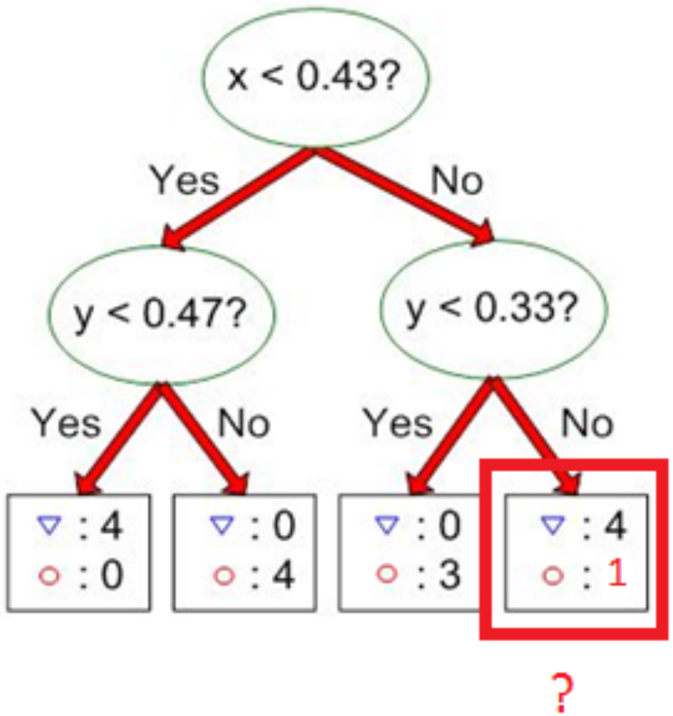


Gain = M0 – M12 vs M0 – M34M: some impurity measure;
M12: weighted average of impurity

Determine When to Stop Splitting

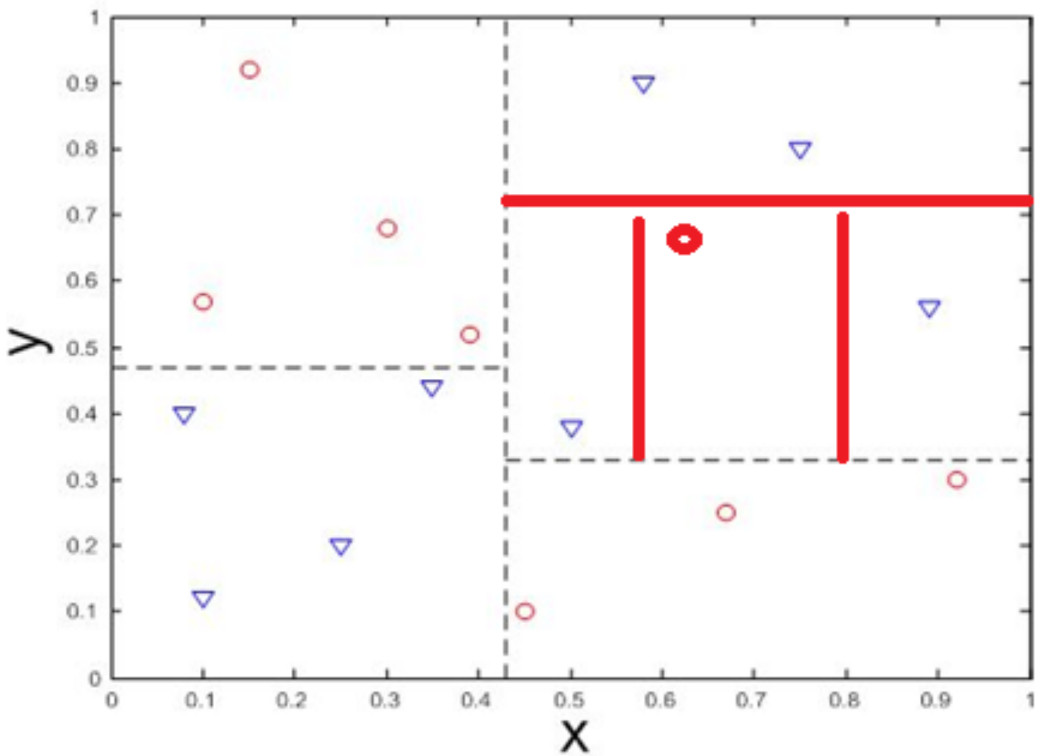
- When a node is homogenous
- When the subsample size is smaller than a threshold

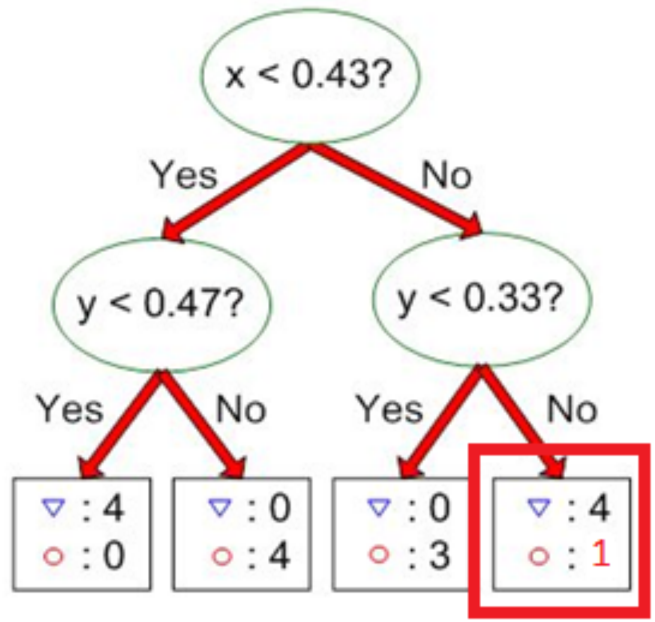




Determine When to Stop Splitting

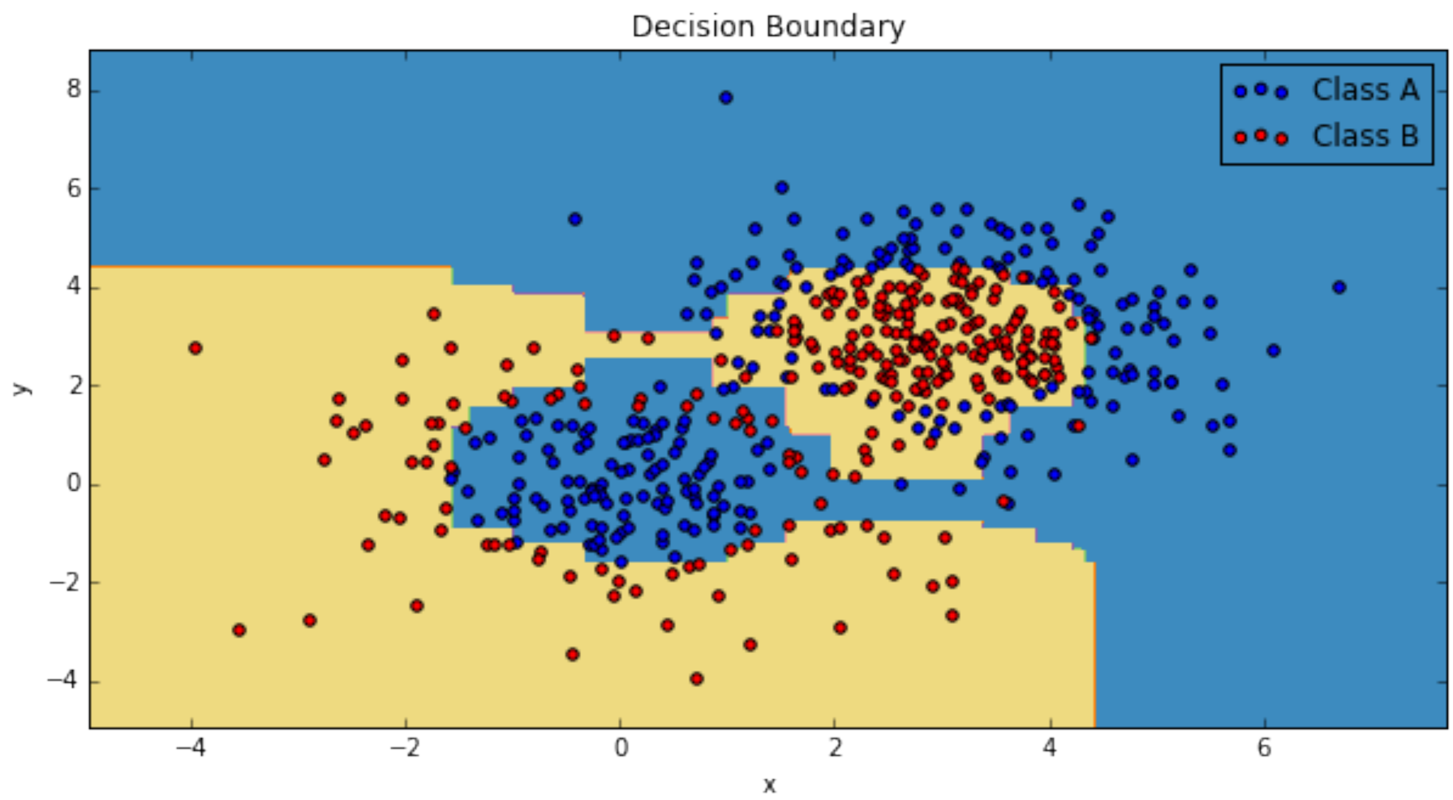
- Remember: our goal is not to sub-divide the data perfectly
- Over-subdivision leads to a complicated decision boundary (over-fitting)





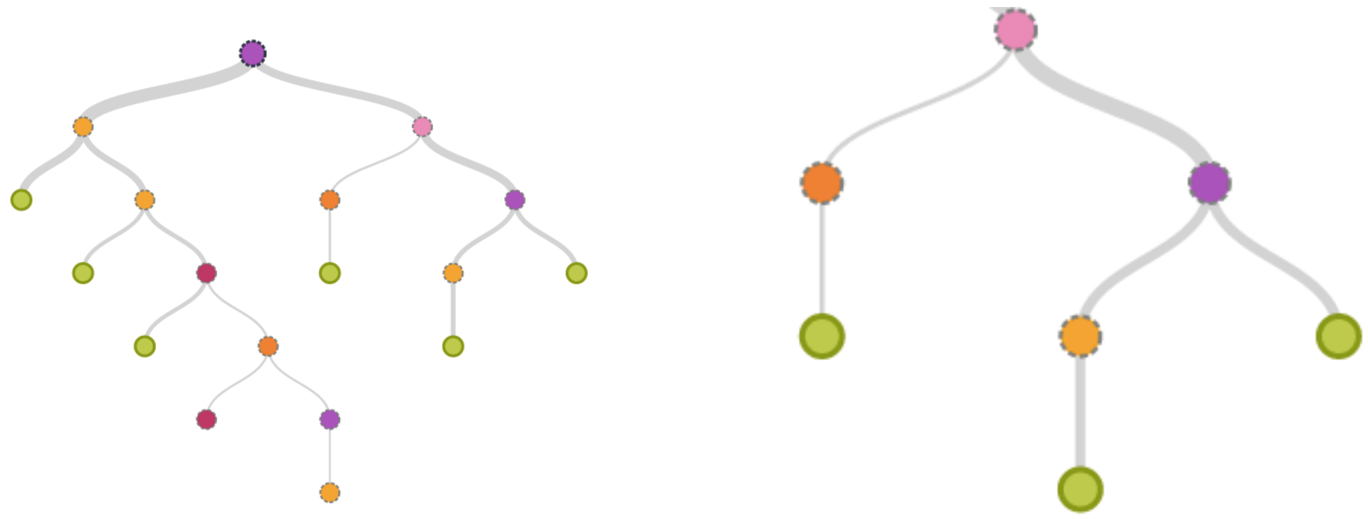
Decision Boundary Complexity

- A perfect decision boundary that models the training data perfectly?
- A good-enough decision boundary that likely generalizes to unseen test data?



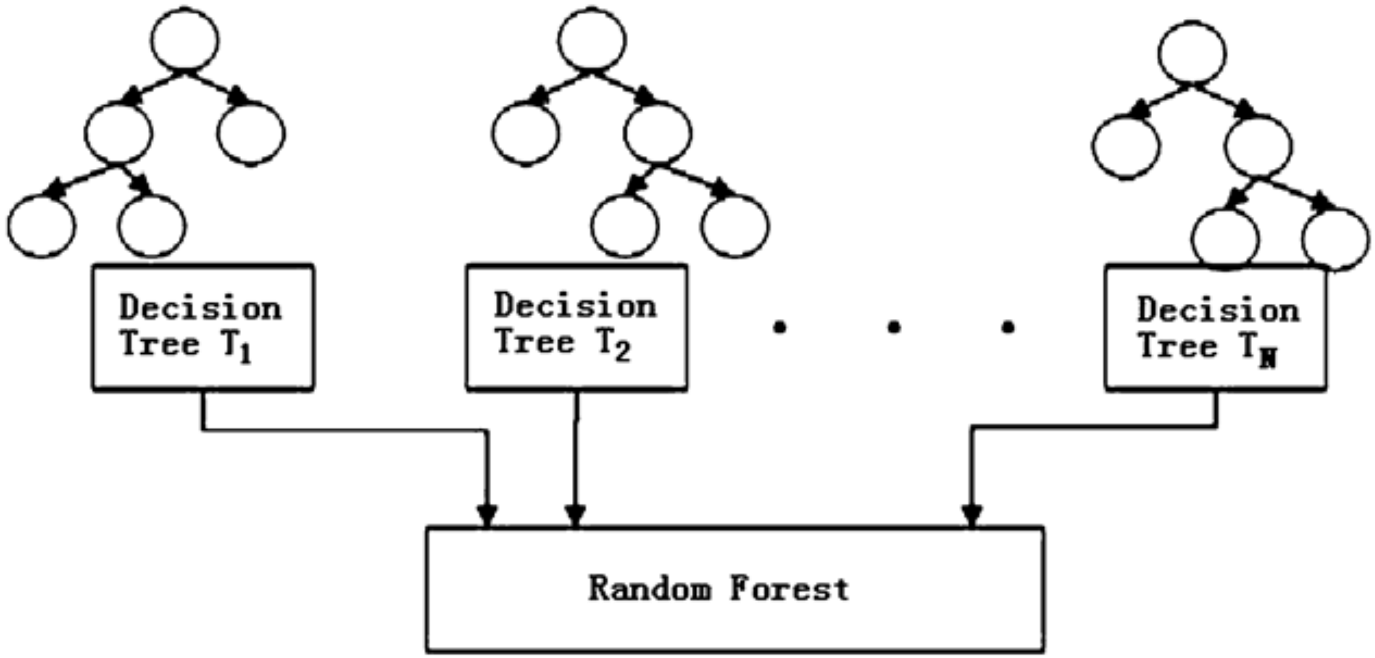
Decision Tree Parameters

- Used to control the complexity of the tree
  - Total number of nodes
  - Tree depth
  - Minimum number of data points for a split
- How to set parameters? Cross-validation



Random Forest

- Random Forest: Community of Experts
  - Train multiple decision trees on random subsets of samples
  - Decision via majority voting



Random Forest

- Each tree is built on a random subset of records of the data: Tree bagging
- Each tree is built on a random subset of features of the data: Random subspace
- In practice: RFs are widely popular and readily implemented in many ML packages

  (Weka, Scikit-learn, Matlab…)

References

This lecture was prepared using some material adapted from:

- https://www-users.cs.umn.edu/~kumar/dmbook/ch4.pdf
- CS059 - Data Mining -- Slides
- http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap4_basic_classification.ppt