

Predictive Analytics, Classification, and Decision Trees

Session Outline

- Introduction to predictive analytics
- Introduction to classification
- Decision Tree Classifier
- Hands-on Lab: Building a decision tree classifier using R

Session Outline

- **Introduction to predictive analytics**
- Introduction to classification
- Decision Tree Classifier
- Hands-on Lab: Building a decision tree classifier using R

Family and Personal Life

- **Location:** Microsoft and Nokia predict future location based on cellular phone and location data.
- **Friendship and connection:** Facebook and LinkedIn
- **Love:**
 - **Match.com:** Predict potential matches
 - **OkCupid:** Which message content is most likely to elicit a response
- **Pregnancy:** Target predicts customer pregnancy
- **Divorce and infidelity:** University and clinical researchers can predict this as well!

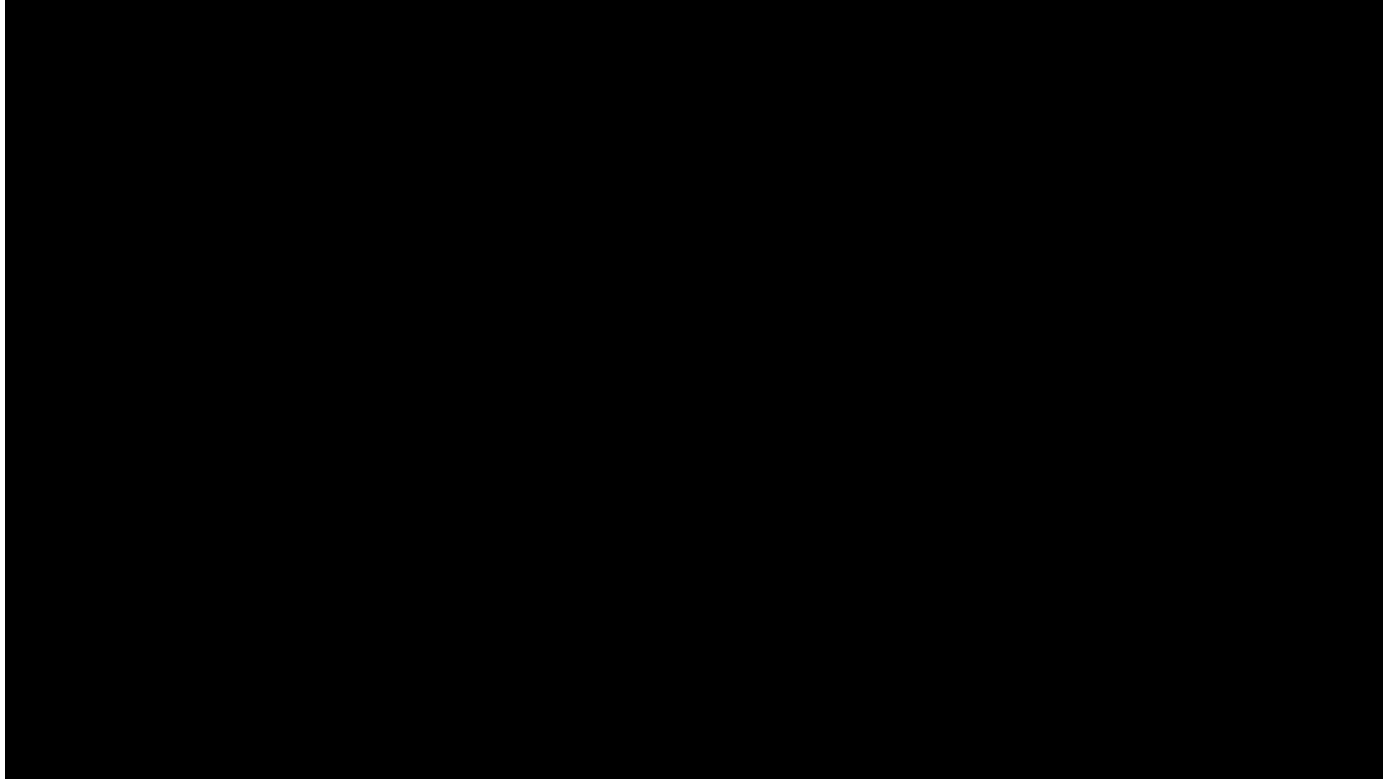
Direct Marketing

- **Cox Communication:** Tripled direct mail responses by predicting propensity to buy
- **Harrah's Las Vegas:** The casino predicts how much a customer will spend over the long term
- **Target:** Increased revenue 15-30 percent with predictive models
- **PREMIER Bankcard:** Reduced mailing cost by \$12 million

Telcos, Retail, and More

- **Fedex:** predicts defection to a competitor with 65-90% accuracy
- **Telcos:** Optus (Australia), Sprint, Telenor(Norway), 2degrees (New Zealand)
- **Amazon:** 35% sales come from product recommendation

Even In Law Enforcement...



Quick Review

- Supervised Learning
- Unsupervised Learning

Unsupervised Learning

- Unsupervised learning:

- Target values unknown
- Training data unlabeled
- Goal: Discover information hidden in the data
- May precede supervised learning

Supervised Learning

- Supervised learning:
 - Target values known
 - Training data labeled with target values
 - Goal: Find a way to map attributes to target value
 - Classification & Regression

Session Outline

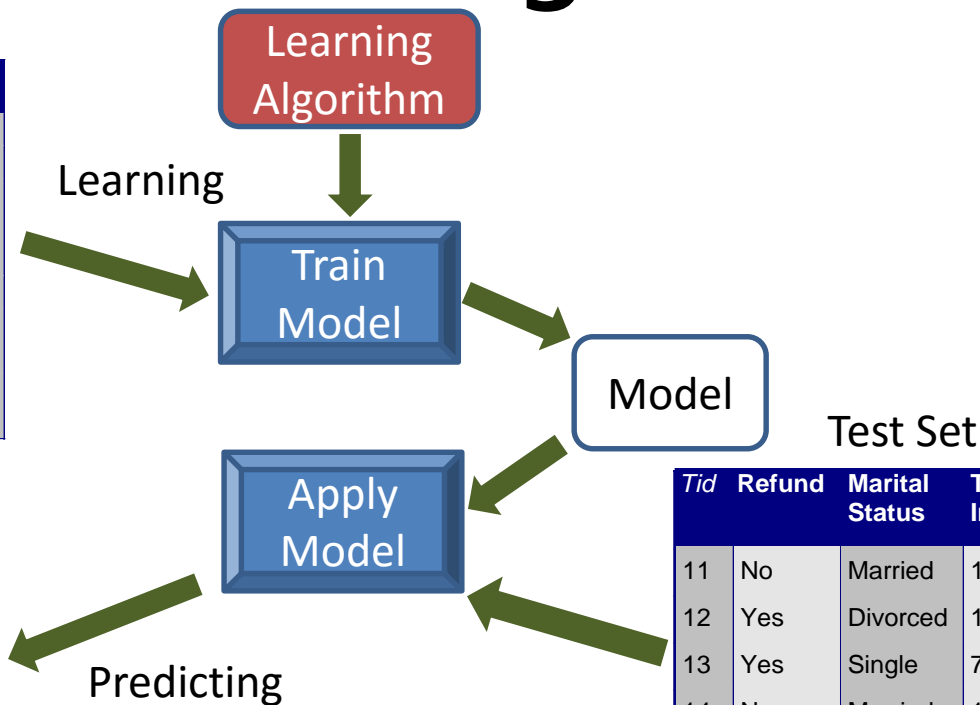
- Introduction to predictive analytics
- **Introduction to classification**
- Decision Tree Classifier
- Hands-on Lab: Building a decision tree classifier using R

Supervised Learning

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

Training Set

Tid	Cheat
11	No
12	Yes
13	No
14	No
15	Yes



Tid	Refund	Marital Status	Taxable Income	Cheat
11	No	Married	125K	?
12	Yes	Divorced	100K	?
13	Yes	Single	70K	?
14	No	Married	120K	?
15	No	Single	95K	?

The Classification Task

- Given a collection of records (training set)
 - Two attribute types: **predictors** and **class**
 - Find a model to map predictor set to class
 - Class is:
 - Categorical
 - Nominal (almost always)

The Classification Task

- Goal: Assign new records a correct class
 - **Training set** used to create model
 - **Test set** used to check
 - Predict test set classes to assess correctness
 - Split data into training and test sets
 - 70/30, 60/40, 50/50

Examples of Classification Tasks

- **Marketing:** Customer groups to target
- **Online:** Bot detection in web traffic
- **Medical:** Predicting tumor cells as benign or malignant
- **Finance:** Credit card fraud detection
- **Document Classification:** Categorizing news stories
- **Security/Surveillance:** Face and fingerprint recognition

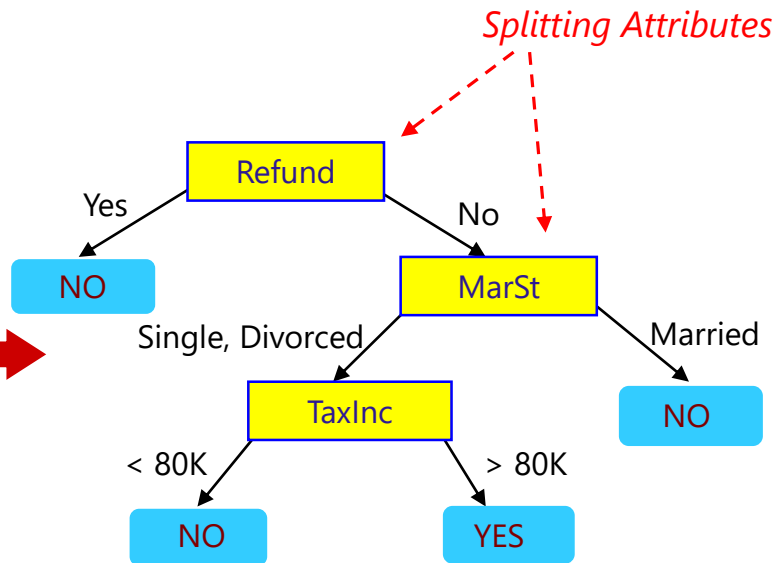
Session Outline

- Introduction to predictive analytics
- Introduction to classification
- **Decision Tree Learning**
- Hands-on Lab: Building a decision tree classifier using R

Decision Tree Classification

<i>Tid</i>	<i>Refund</i>	<i>Marital Status</i>	<i>Taxable Income</i>	<i>Cheat</i>
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

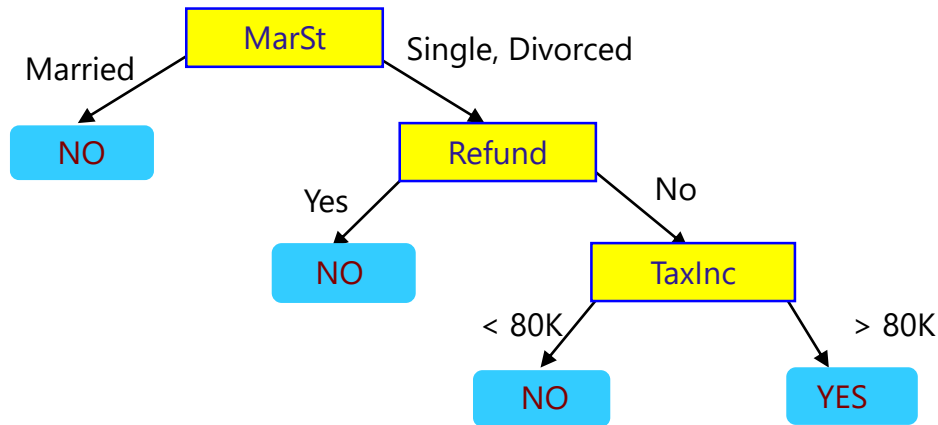


Model: Decision Tree

A Different Decision Tree

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



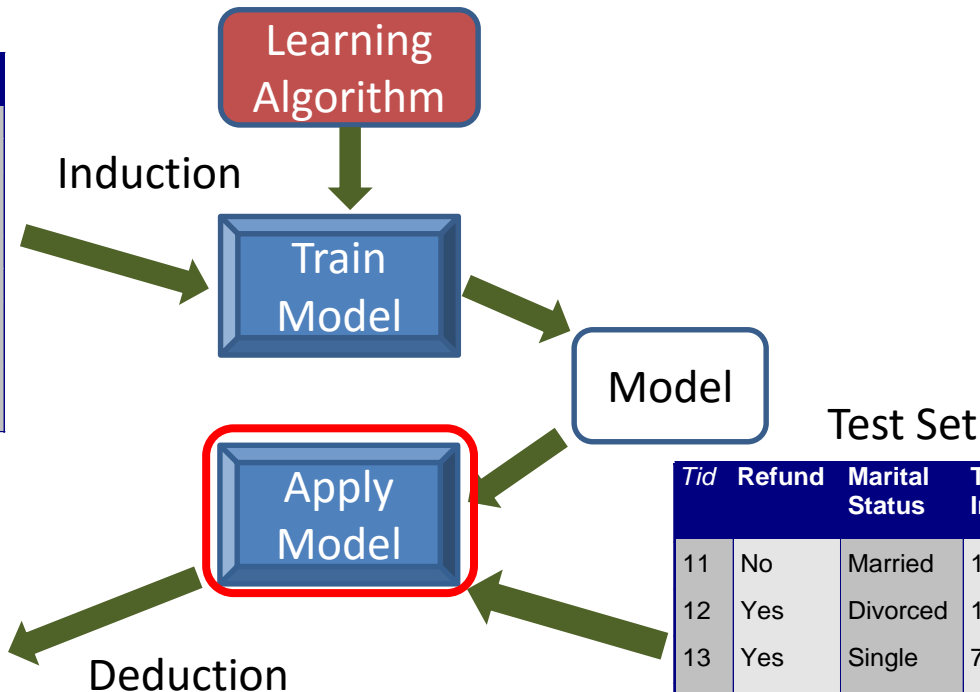
There could be more than one tree
that fits the same data!

Decision Tree Application

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

Training Set

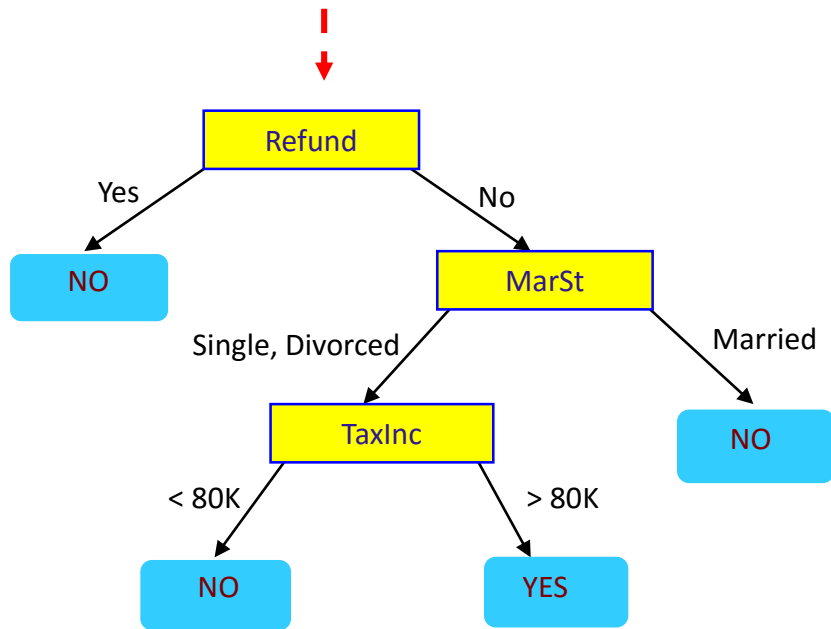
Tid	Cheat
11	No
12	Yes
13	No
14	No
15	Yes



Tid	Refund	Marital Status	Taxable Income	Cheat
11	No	Married	125K	?
12	Yes	Divorced	100K	?
13	Yes	Single	70K	?
14	No	Married	120K	?
15	No	Single	95K	?

Apply Model to Test Data

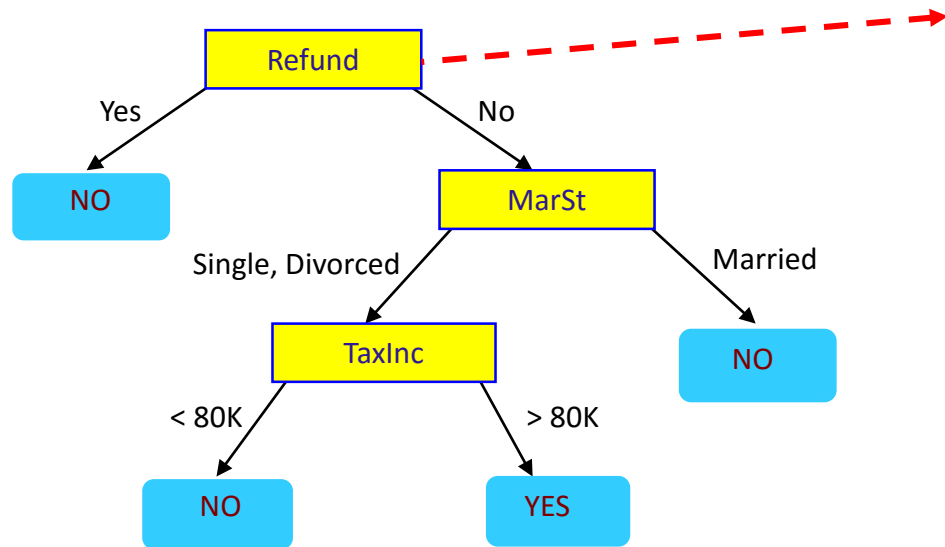
Start from the root of tree.



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

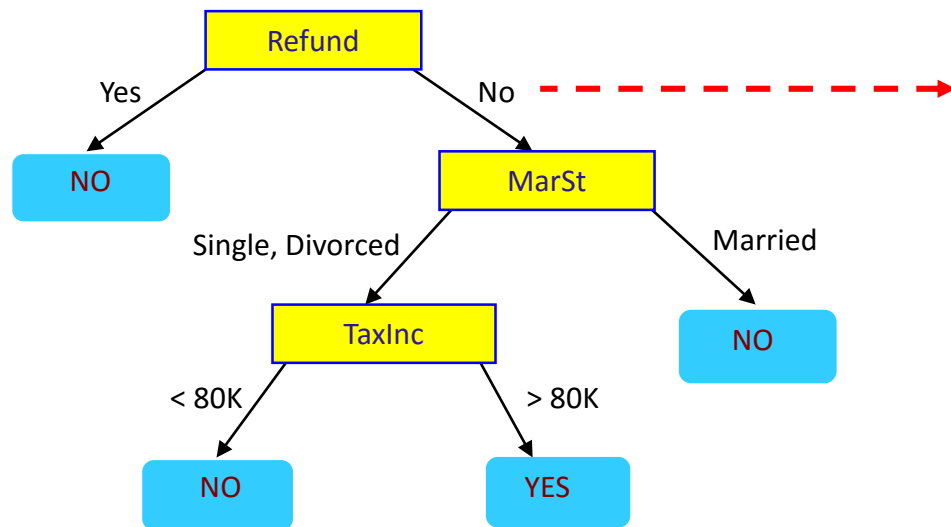
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

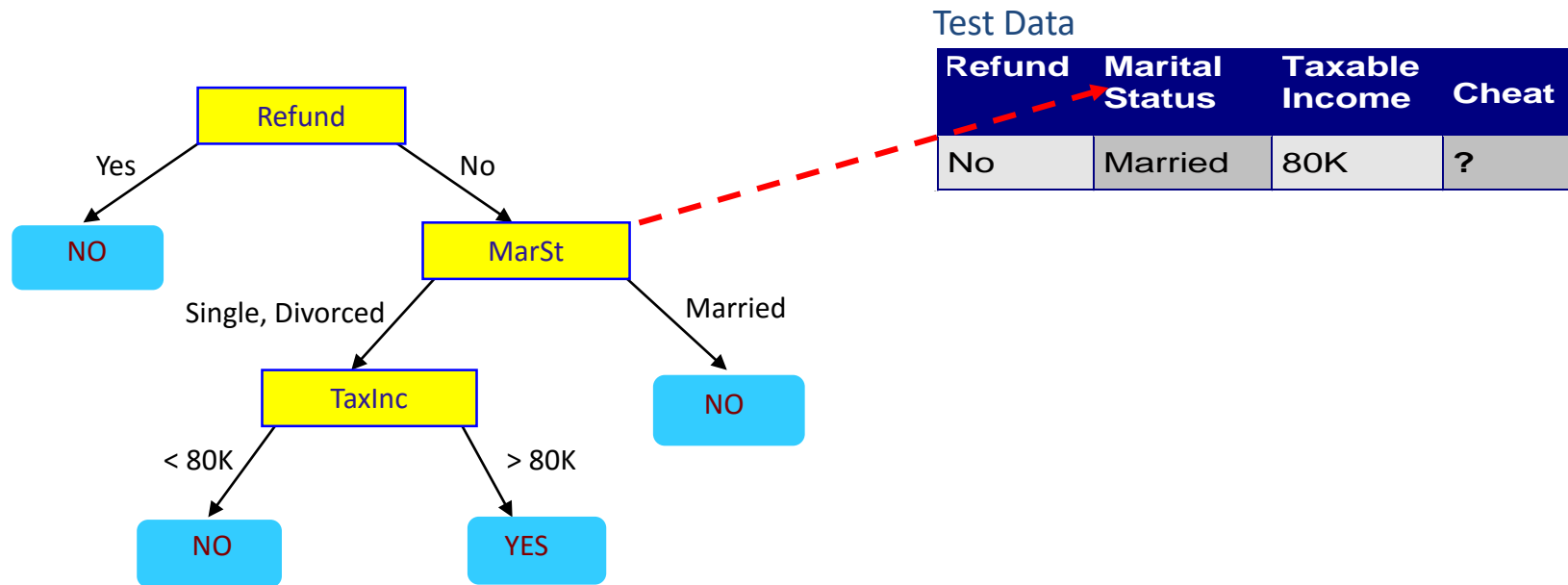
Apply Model to Test Data



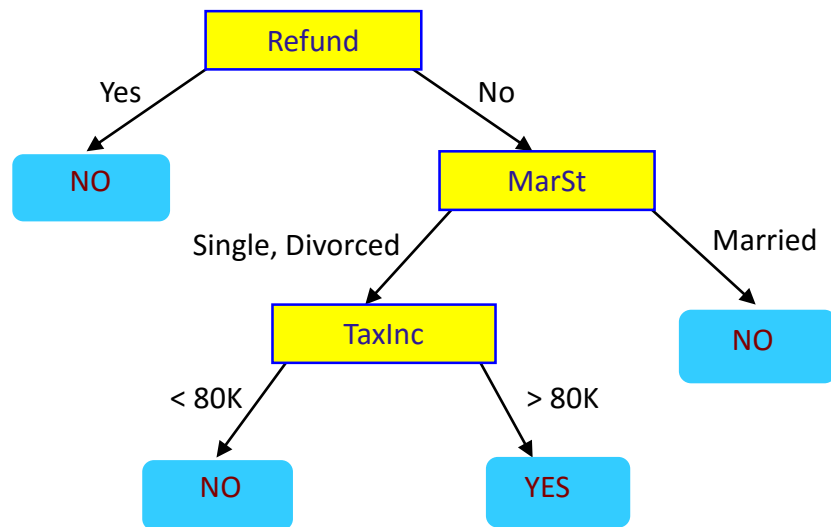
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



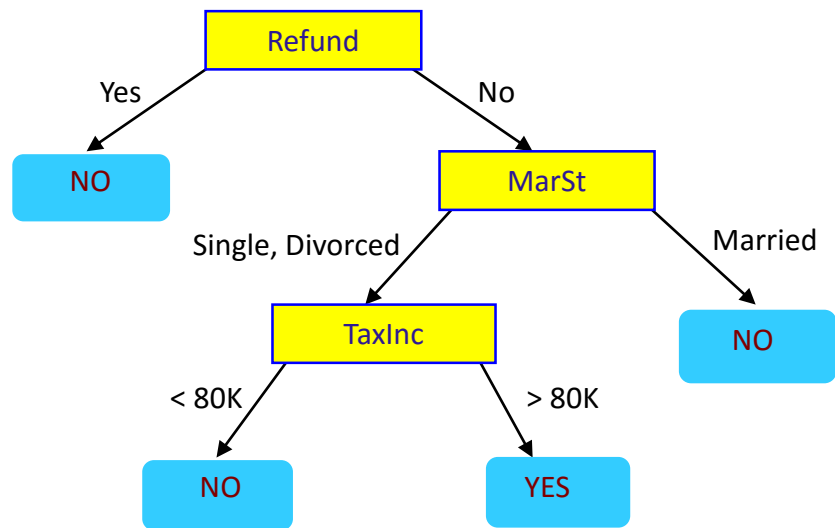
Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

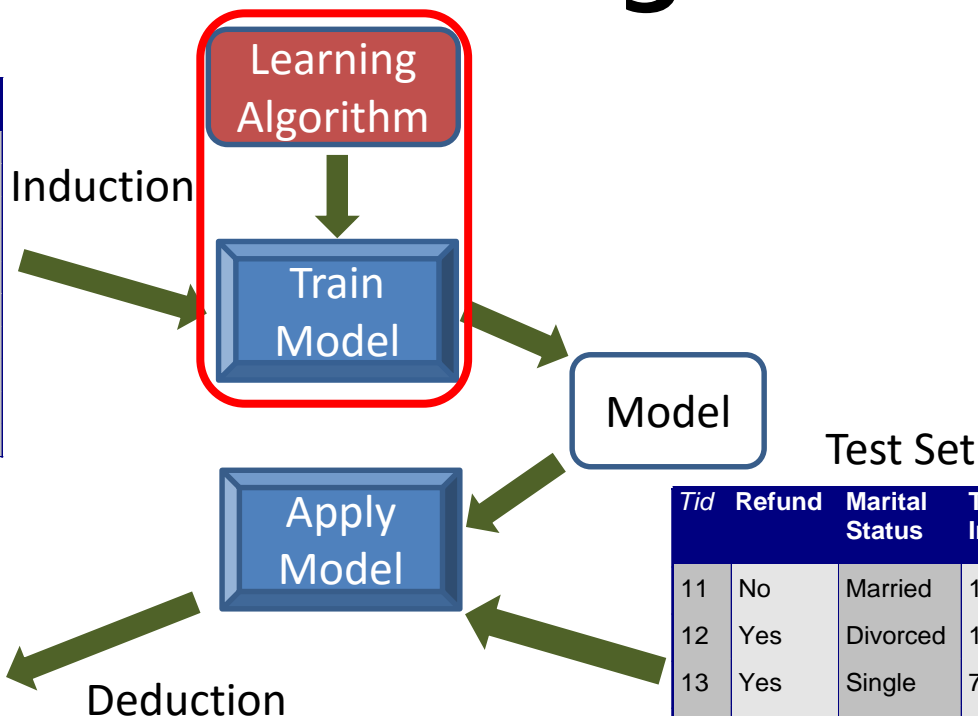
Cheat = "No"

Decision Tree Training

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

Training Set

Tid	Cheat
11	No
12	Yes
13	No
14	No
15	Yes



Tid	Refund	Marital Status	Taxable Income	Cheat
11	No	Married	125K	?
12	Yes	Divorced	100K	?
13	Yes	Single	70K	?
14	No	Married	120K	?
15	No	Single	95K	?

How Do We Get A Tree?

- Exponentially many decision trees are possible
- Finding the optimal tree is **infeasible**
- Greedy methods that find sub-optimal solutions do exist

Tree Induction

- Greedy strategy

- Split based attribute test that optimizes a criterion

- Issues

- How to split the records
 - How to specify the attribute test condition?
 - How to determine the best split?
 - When do we stop?

Tree Induction

- Greedy strategy

- Split based attribute test that optimizes a criterion

- Issues

- How to split the records
 - **What attribute test criterion?**
 - How to determine the best split?
 - When do we stop?

How to Specify Test Condition?

- Attribute types

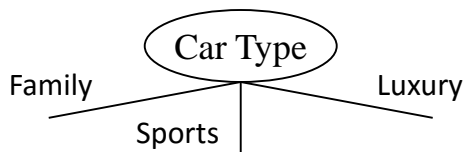
- Nominal
- Ordinal
- Continuous

- Order of split

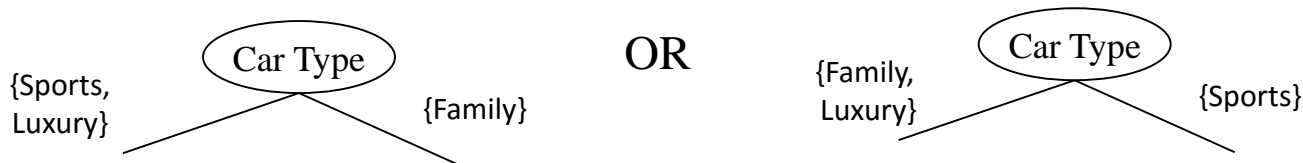
- 2-way split
- Multi-way split

Splitting: Nominal Attributes

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

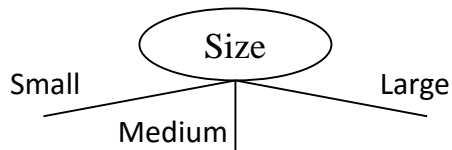


- Binary split: Divide values into two subsets. Need to find optimal partitioning.

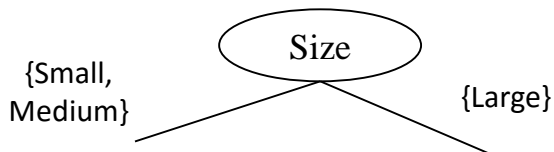


Splitting: Ordinal Attributes

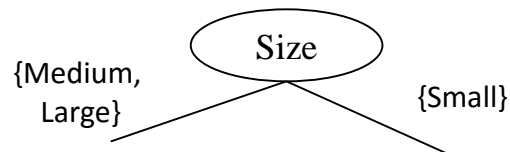
- Multi-way split: As many partitions as distinct values.



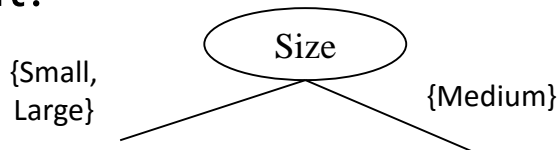
- Binary split: Divides values into two subsets. Need to find optimal partitioning.



OR



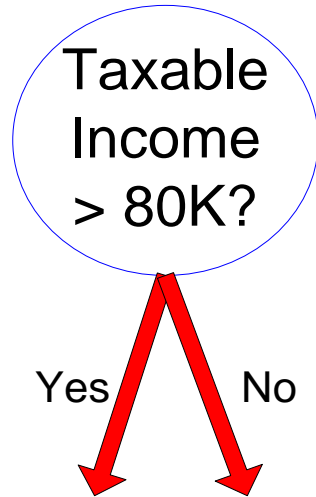
- What about this split?



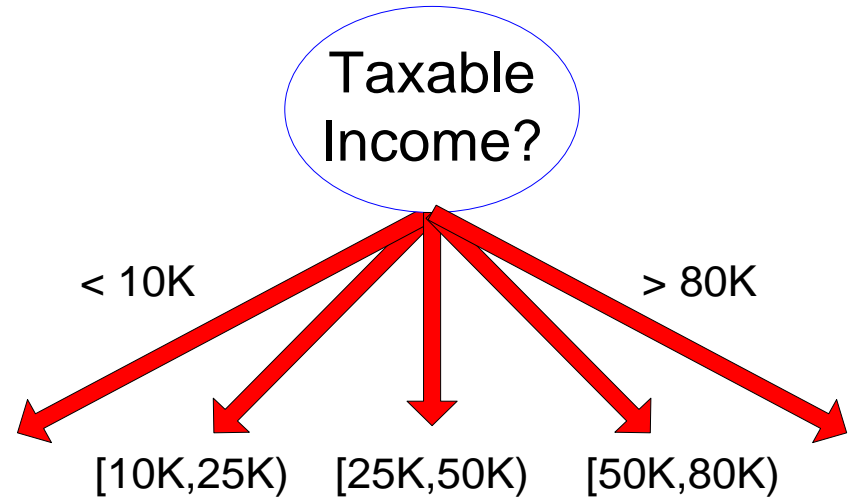
Splitting: Continuous Attributes

- Discretize: transform to ordinal categorical attribute
 - Static – “bucket” once at the beginning
 - Dynamic – “bucket” at each node
 - Equal interval bucketing
 - Equal frequency bucketing (percentiles)
 - clustering.
- Binary Decision: $(A < v)$ or $(A \geq v)$
 - Consider all possible splits and finds the best cut value
 - Can be more computationally intensive

Splitting on Continuous Attributes



(i) Binary split



(ii) Multi-way split

Tree Induction

- Greedy strategy

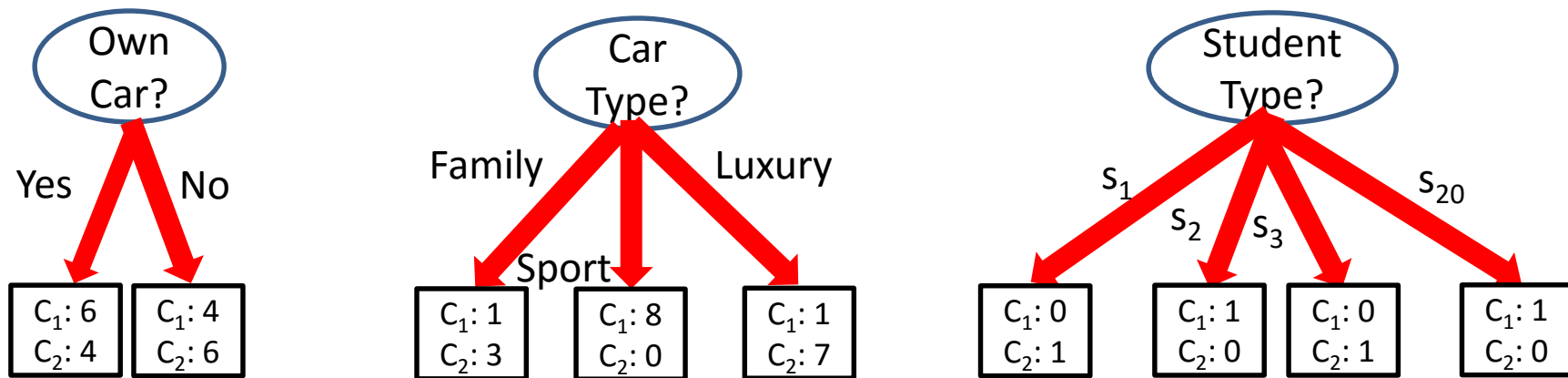
- Split based attribute test that optimizes a criterion

- Issues

- How to split the records
 - What attribute test criterion?
 - **How to determine the best split?**
 - When do we stop?

What is The Best Split?

Before Splitting: 10 records of class 1, 10 records of class 2



Which test condition is the best?

What is The Best Split?

- Greedy approach:
 - Homogeneous class distribution preferred
- Need a measure of **node impurity**:

C0: 5
C1: 5

Non-homogeneous

High degree of impurity

C0: 9
C1: 1

Homogeneous

Low degree of impurity

Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

Impurity Measure: GINI

- Gini Index for a given node t :

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

- $p(j | t)$ 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
 n_c =number of classes
- Minimum (0.0) when all records belong to one class, implying most interesting information

C_1	0
C_2	6
Gini=0.000	

C_1	1
C_2	5
Gini=0.278	

C_1	2
C_2	4
Gini=0.444	

C_1	3
C_2	3
Gini=0.500	

Impurity Measure: GINI

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

C ₁	0
C ₂	6

$$P(C_1) = 0/6 = 0 \quad P(C_2) = 6/6 = 1$$

$$Gini = 1 - P(C_1)^2 - P(C_2)^2 = 1 - 0 - 1 = 0$$

C ₁	1
C ₂	5

$$P(C_1) = 1/6 \quad P(C_2) = 5/6$$

$$Gini = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

C ₁	2
C ₂	4

$$P(C_1) = 2/6 \quad P(C_2) = 4/6$$

$$Gini = 1 - (2/6)^2 - (4/6)^2 = 0.444$$

Impurity Measure: GINI

- When a node p is split into k partitions (children), the quality of split is computed as:

$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

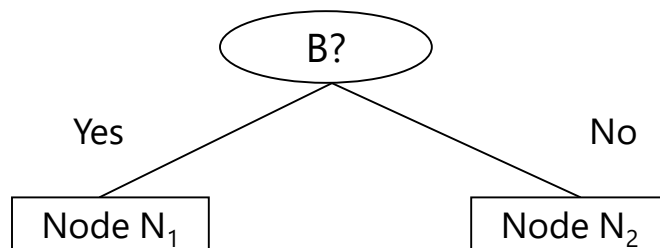
where,

n_i = number of records at child i ,

n = number of records at node p

Impurity Measure: Computing GINI Index

- Splits into two partitions
- Effect of Weighing partitions:
 - Larger and Purer Partitions are sought after



$$\begin{aligned} \text{Gini}(N_1) &= 1 - (5/6)^2 - (2/6)^2 \\ &= 0.194 \end{aligned}$$

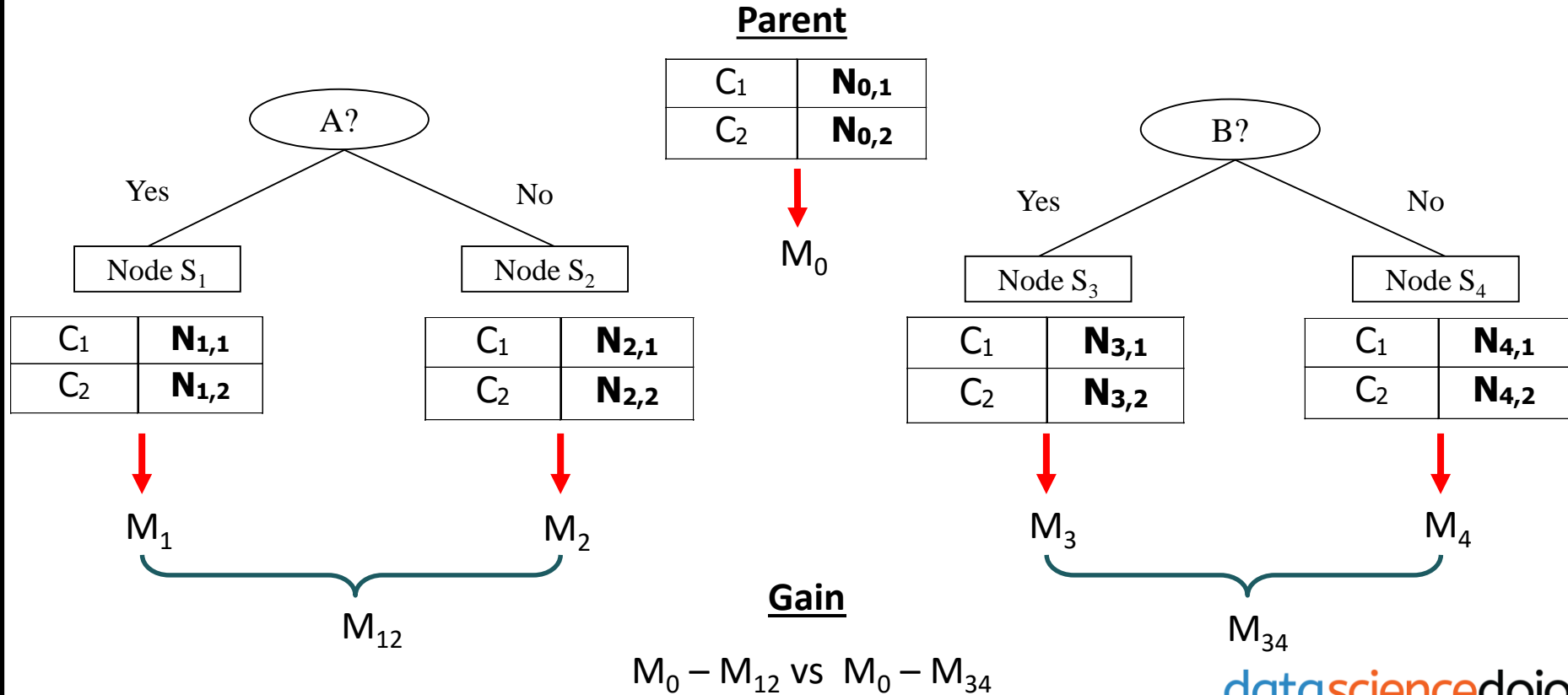
$$\begin{aligned} \text{Gini}(N_2) &= 1 - (1/6)^2 - (4/6)^2 \\ &= 0.528 \end{aligned}$$

	N ₁	N ₂
C ₁	5	1
C ₂	2	4
Gini=0.333		

	Parent
C ₁	6
C ₂	6
Gini = 0.500	

$$\begin{aligned} \text{Gini}(N_1 + N_2) &= 7/12 * 0.194 + \\ &\quad 5/12 * 0.528 \\ &= 0.333 \end{aligned}$$

A or B?



Impurity Measure: Entropy

- Entropy at a given node t:

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

- $p(j | t)$ is the relative frequency of class j at node t
- Maximum: records equally distributed
- Minimum: all records belong to one class

Impurity Measure: Entropy

$$Entropy(t) = -\sum_j p(j | t) \log_2 p(j | t)$$

C ₁	0
C ₂	6

$$P(C_1) = 0/6 = 0 \quad P(C_2) = 6/6 = 1$$

$$Entropy = -0 \log 0 - 1 \log 1 = -0 - 0 = 0$$

C ₁	1
C ₂	5

$$P(C_1) = 1/6 \quad P(C_2) = 5/6$$

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

C ₁	2
C ₂	4

$$P(C_1) = 2/6 \quad P(C_2) = 4/6$$

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

Impurity Measure: Information

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^k \frac{n_i}{n} Entropy(i) \right)$$

- Node p is split into k partitions
 - n_i is number of records in partition i
- Measures Reduction in Entropy
- Choose split that maximizes GAIN
- Tends to prefer splits with large number of partitions

Impurity Measure: Information

- Gain Ratio:

$$GainRATIO_{split} = \frac{GAIN_{Split}}{SplitINFO}$$

$$SplitINFO = -\sum_{i=1}^k \frac{n_i}{n} \log \frac{n_i}{n}$$

- Node p is split into k partitions
- n_i is the number of records in partition i
- Penalizes GAIN metric for extra splits
- Counters tendency towards many splits

Impurity Measure: Classification Error

- Classification error at a node t :

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

- Maximum: records are equally distributed
- Minimum: all records belong to one class
- Similar to information gain
 - Less sensitive for > 2 or 3 splits
 - Less prone to overfitting

Impurity Measure: Classification Error

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

C ₁	0
C ₂	6

$$P(C_1) = 0/6 = 0 \quad P(C_2) = 6/6 = 1$$

$$Error = 1 - \max(0, 1) = 1 - 1 = 0$$

C ₁	1
C ₂	5

$$P(C_1) = 1/6 \quad P(C_2) = 5/6$$

$$Error = 1 - \max(1/6, 5/6) = 1 - 5/6 = 1/6$$

C ₁	2
C ₂	4

$$P(C_1) = 2/6 \quad P(C_2) = 4/6$$

$$Error = 1 - \max(2/6, 4/6) = 1 - 4/6 = 1/3$$

Tree Induction

- Greedy strategy

- Split based attribute test that optimizes a criterion

- Issues

- How to split the records
 - What attribute test criterion?
 - How to determine the best split?
 - **When do we stop?**

Sample Stopping Criteria

- All the records belong to the same class
- All the records have similar attribute values
- Fixed termination
 - Number of Levels
 - Number in Leaf Node

Decision Trees - PROS

■ Intuitive

- Easy interpretation for small trees

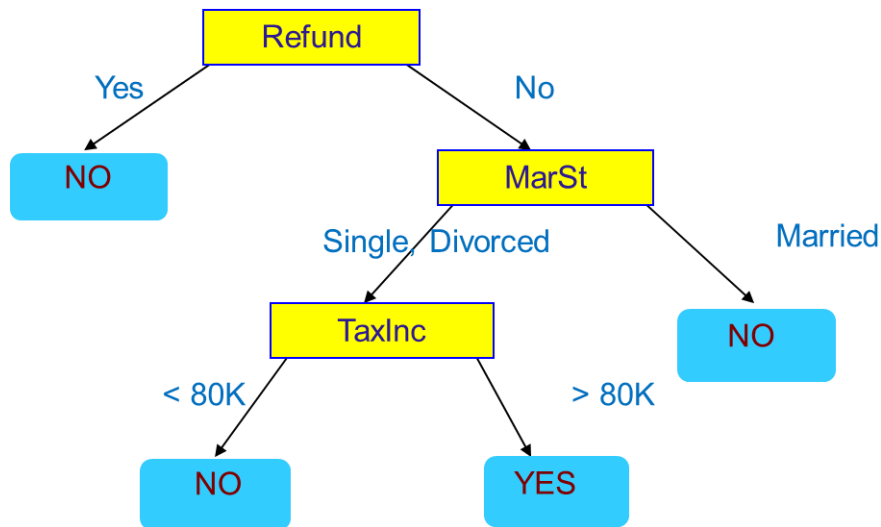
■ Non parametric:

- Incorporate both numeric and categorical attributes

■ Fast

- Once rules are developed, prediction is rapid

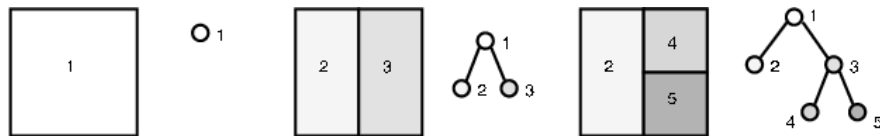
■ Robust to outliers



Decision Trees - CONS

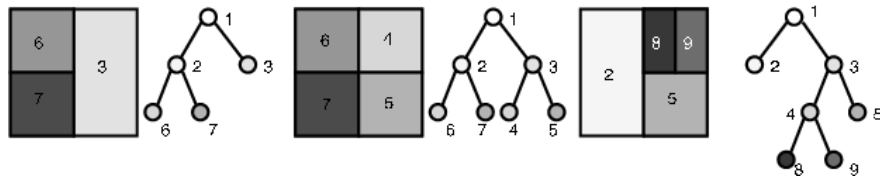
■ Overfitting

- Must be trained with great care



■ Rectangular Classification

- Recursive partitioning of data may not capture complex relationships



■ Tree replication

QUESTIONS

Session Outline

- Introduction to predictive analytics
- Introduction to classification
- Decision Tree Classifier
- **Hands-on Lab: Building a decision tree classifier using R**