

# Decision Tree Classifier

# Topics

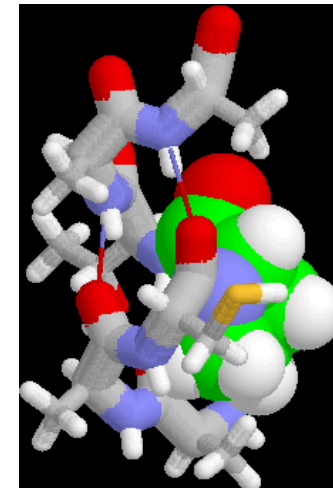
- Classification Basics
- Introduction to Decision Tree Classifier
- Basic Algorithms
  - CART
  - ID3
  - C4.5
- Issues of Decision Trees

# Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: previously unseen records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.

# Examples of Classification Task

- Predicting tumor cells as benign or malignant
- Classifying credit card transactions as legitimate or fraudulent
- Classifying secondary structures of protein as alpha-helix, beta-sheet, or random coil
- Categorizing news stories as finance, weather, entertainment, sports, etc



# Classification Techniques

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

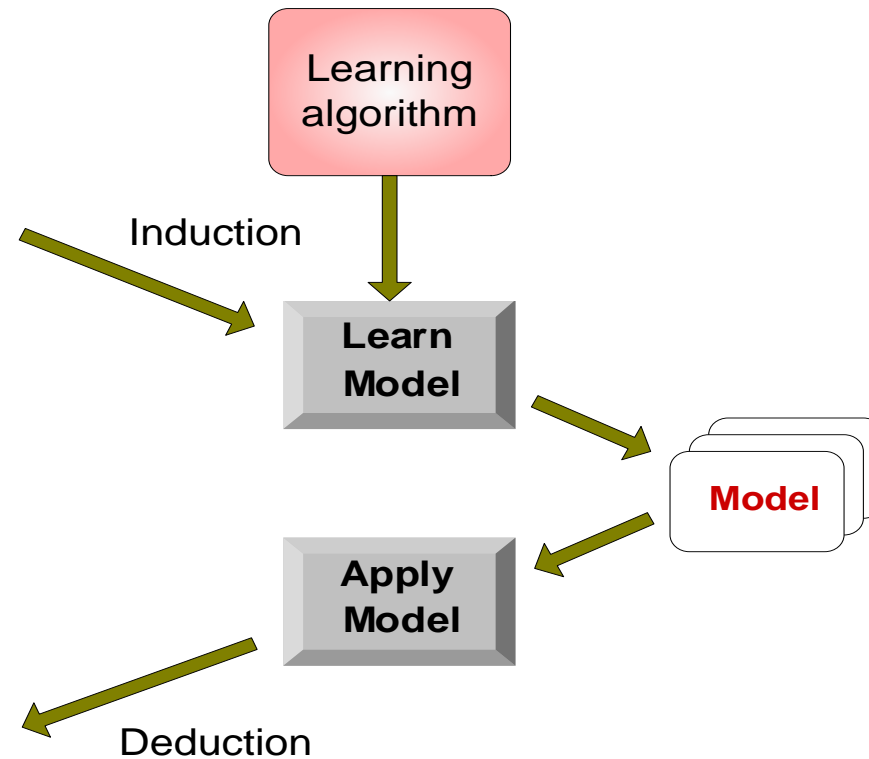
# Illustrating a basic Classification Task

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

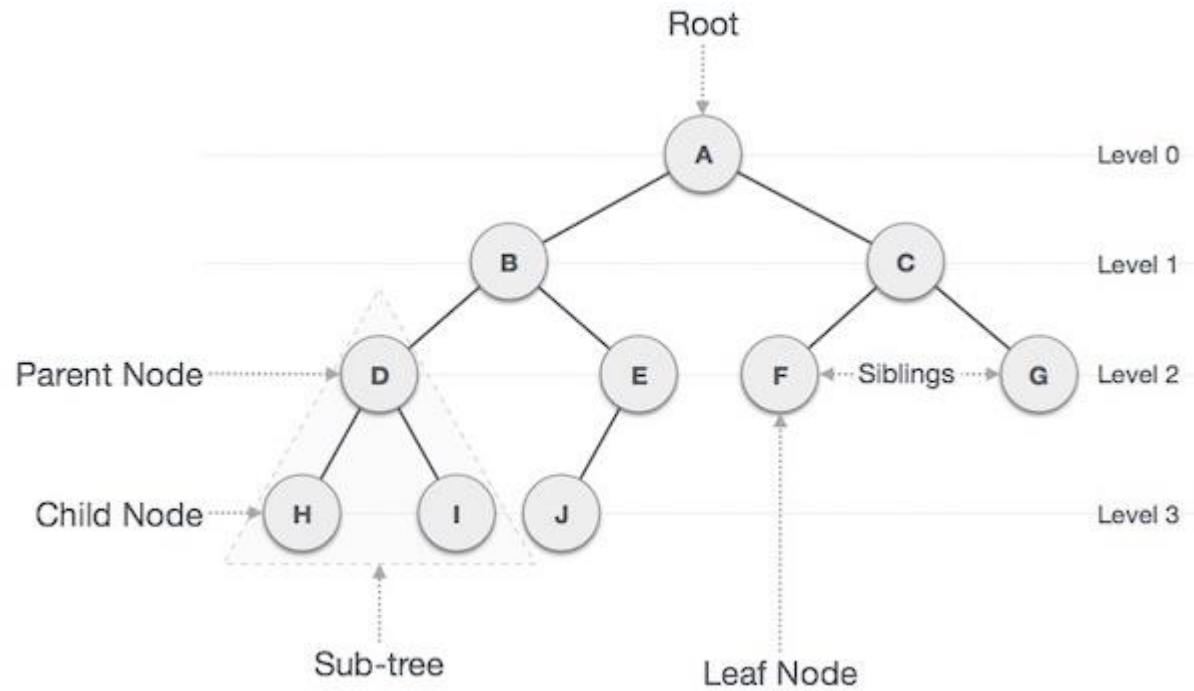
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



# Decision Tree

## Tree Introduction

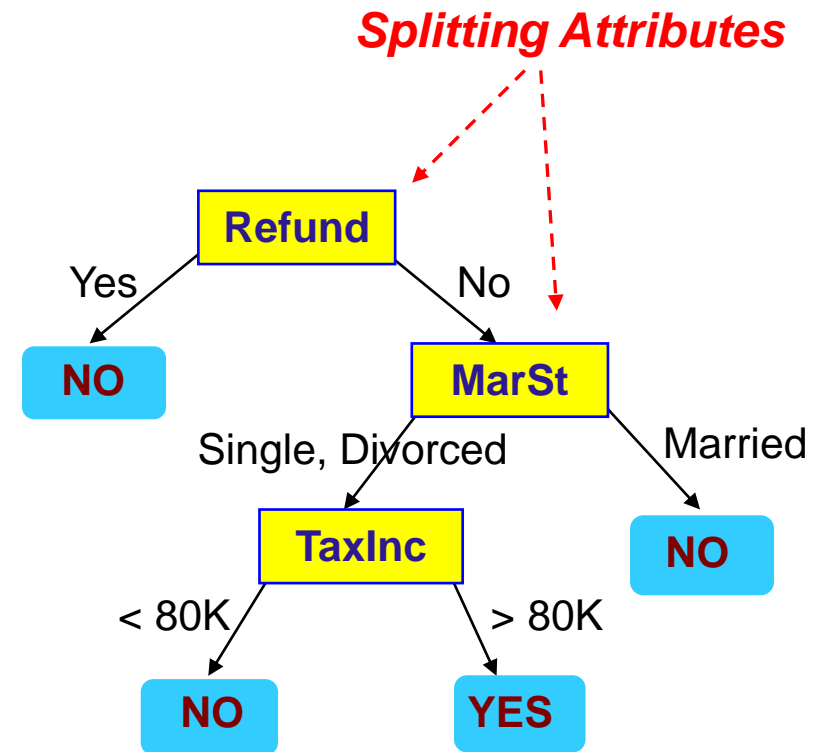


# Example of a Decision Tree

<i>Tid</i>	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

categorical  
categorical  
continuous  
class

Training Data



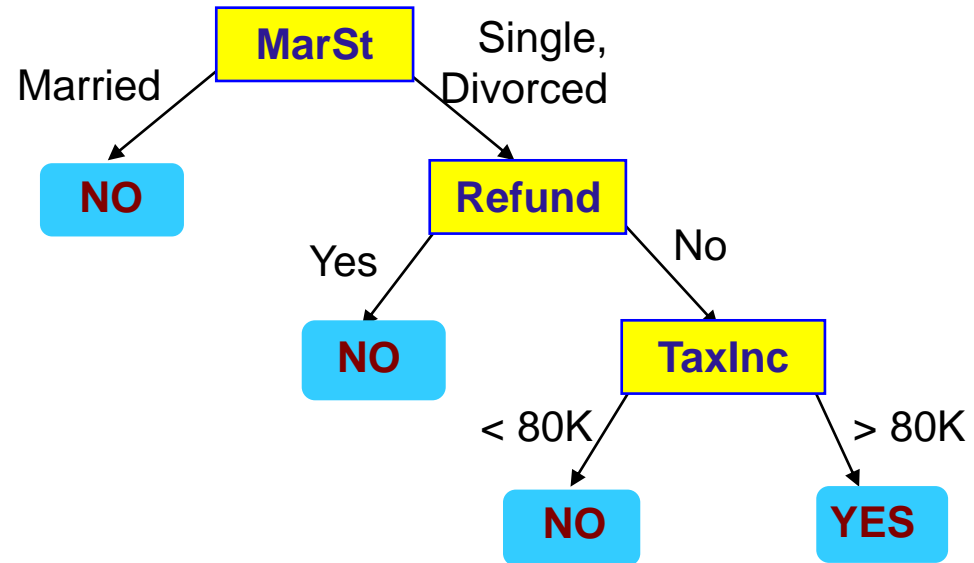
Model: Decision Tree



# Another Example of Decision Tree

<i>Tid</i>	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

categorical  
categorical  
continuous  
class



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

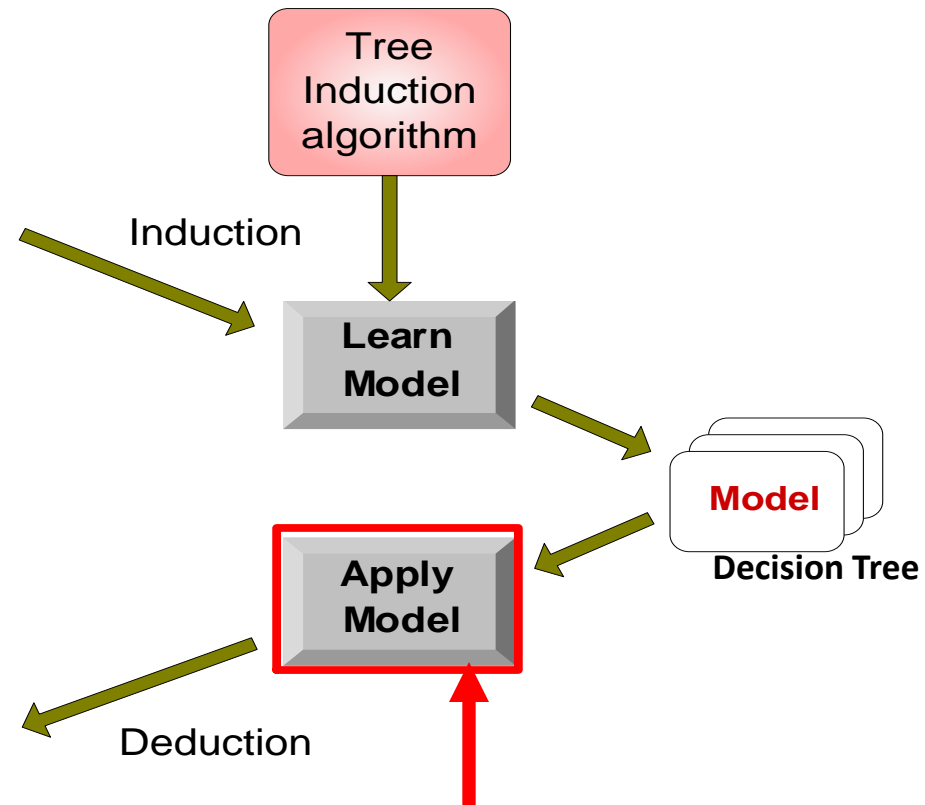
# Decision Tree Classification Task

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

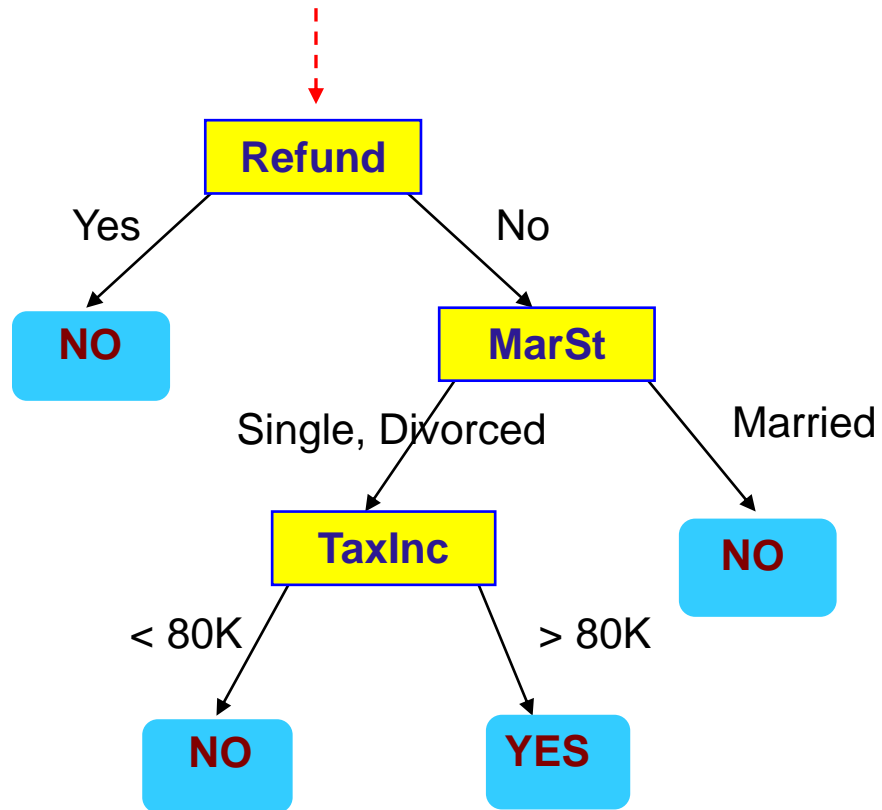
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



# Apply Model to Test Data

Start from the root of tree.



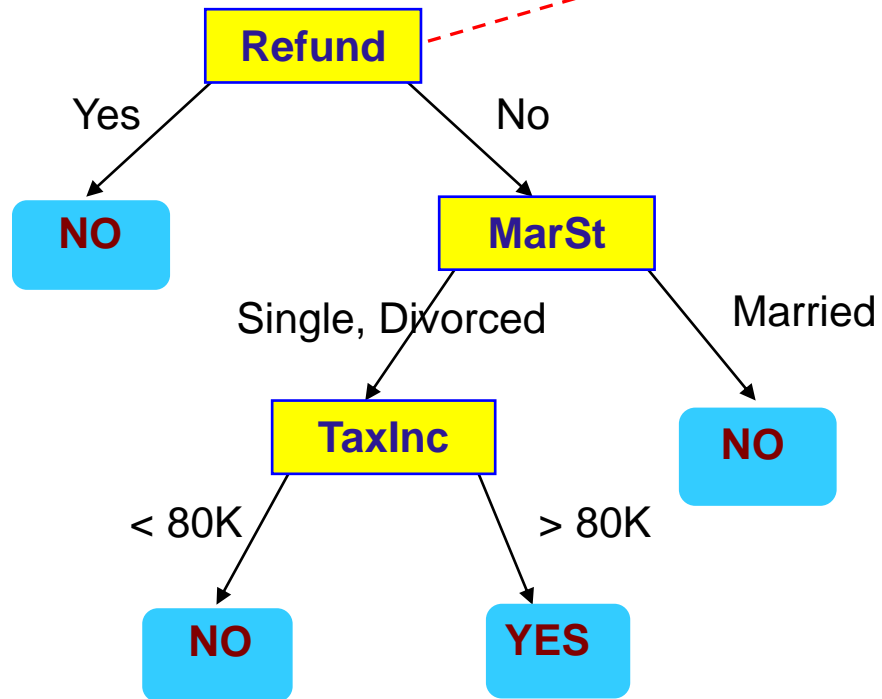
## Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

# Apply Model to Test Data

Test Data

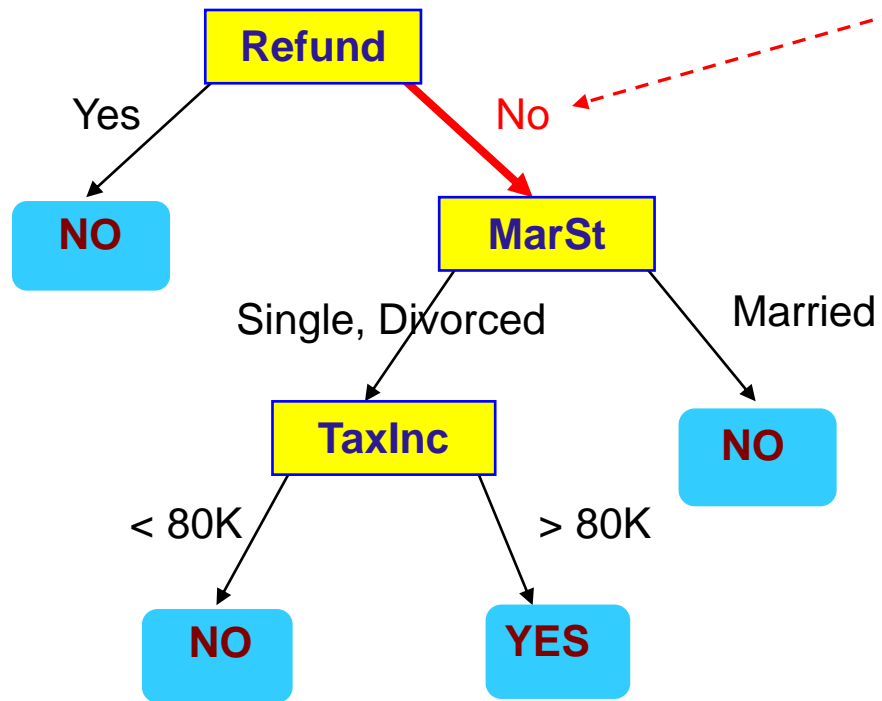
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

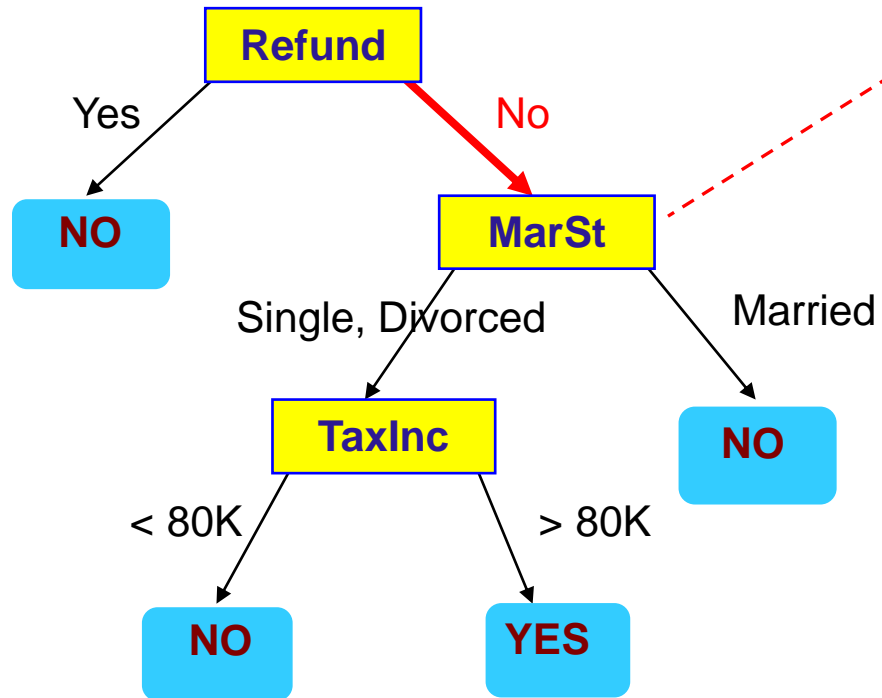
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

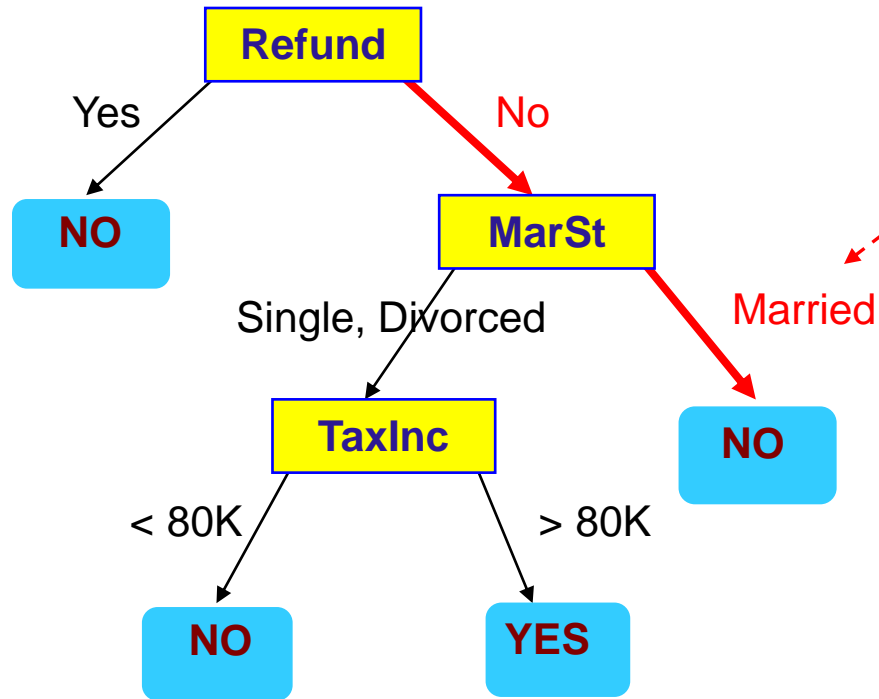
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

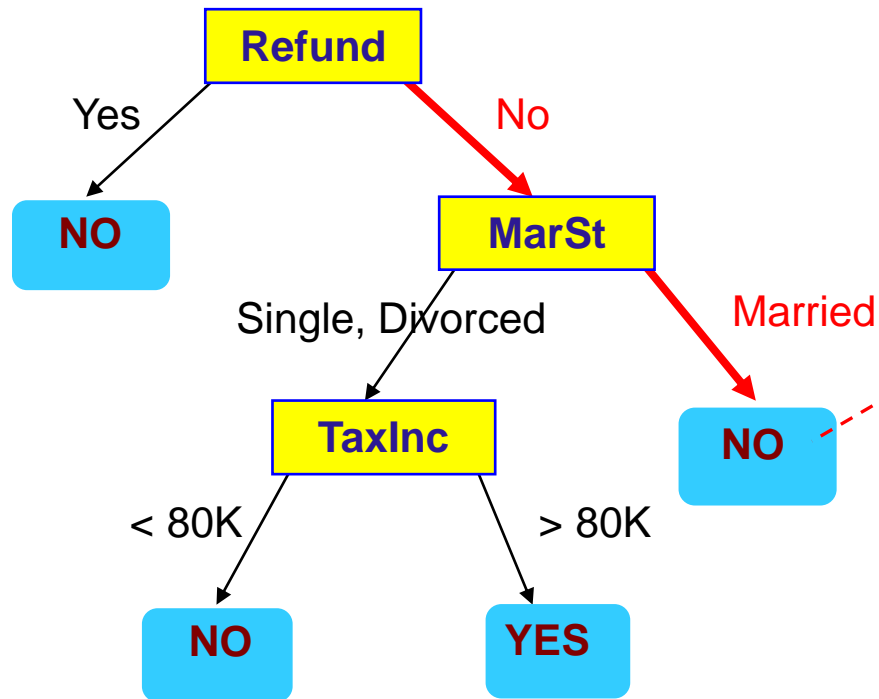
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



# Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"



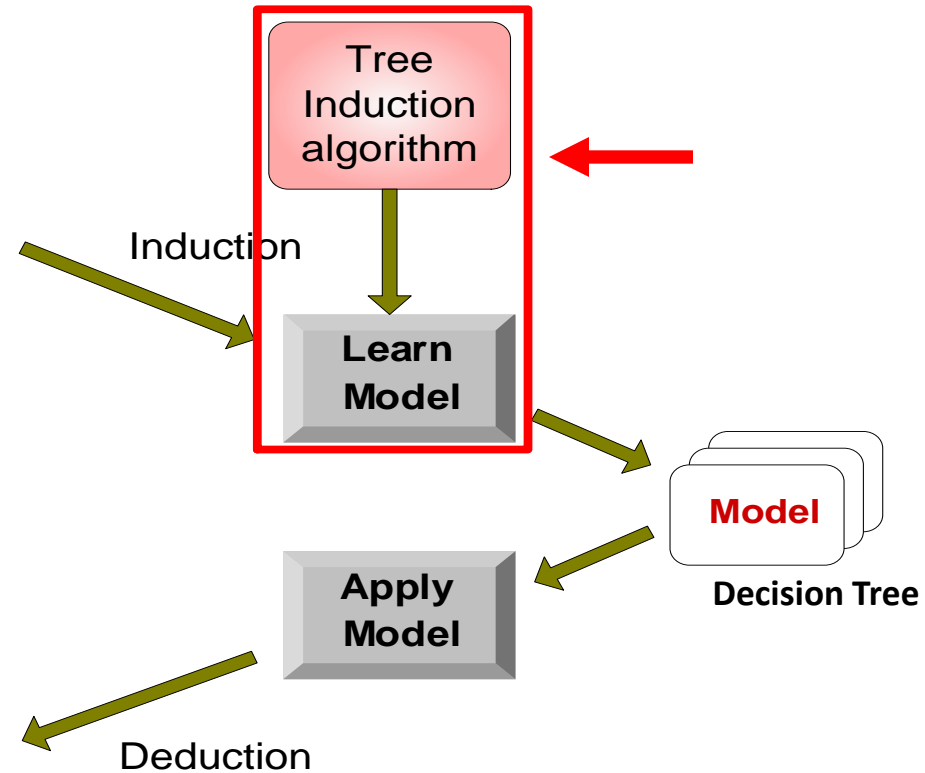
# Decision Tree Classification Task

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



# Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- 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

# Tree Induction

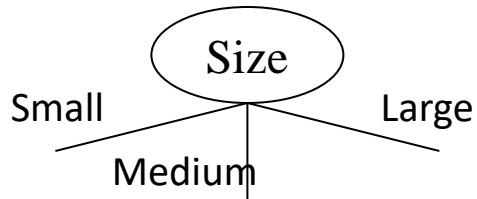
- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

# How to Specify Test Condition?

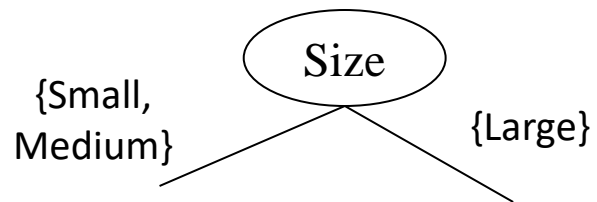
- Depends on attribute types
  - Nominal: Nominal scale is a naming scale, where variables are simply “named” or labeled, with no specific order.
  - Ordinal: Ordinal scale has all its variables in a specific order.
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

# Splitting Based on Ordinal Attributes

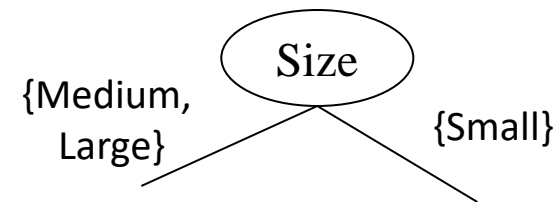
- **Multi-way split:** Use as many partitions as distinct values.



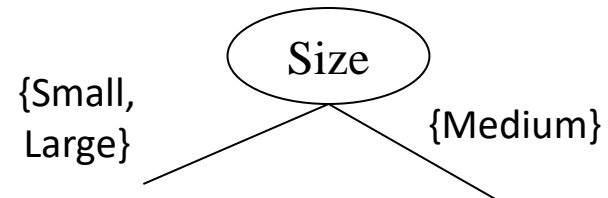
- **Binary split:** Divides values into two subsets.  
Need to find optimal partitioning.



OR

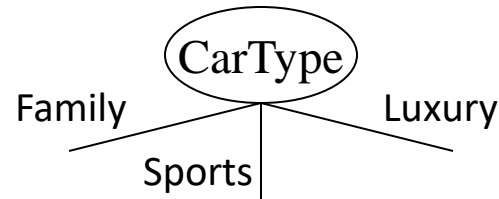


- What about this split?

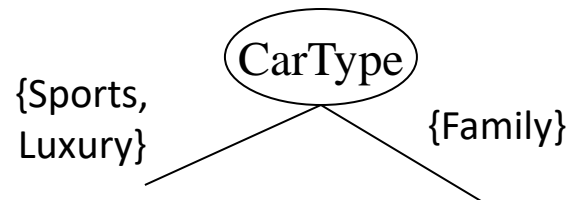


# Splitting Based on Nominal Attributes

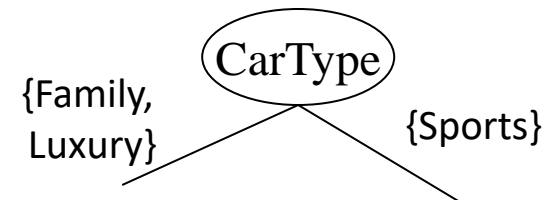
- **Multi-way split:** Use as many partitions as distinct values.



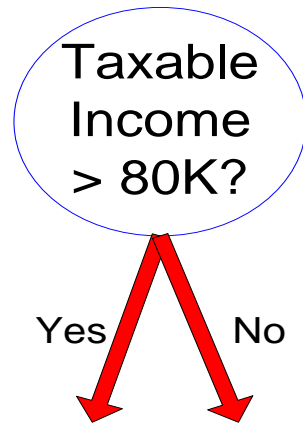
- **Binary split:** Divides values into two subsets.  
Need to find optimal partitioning.



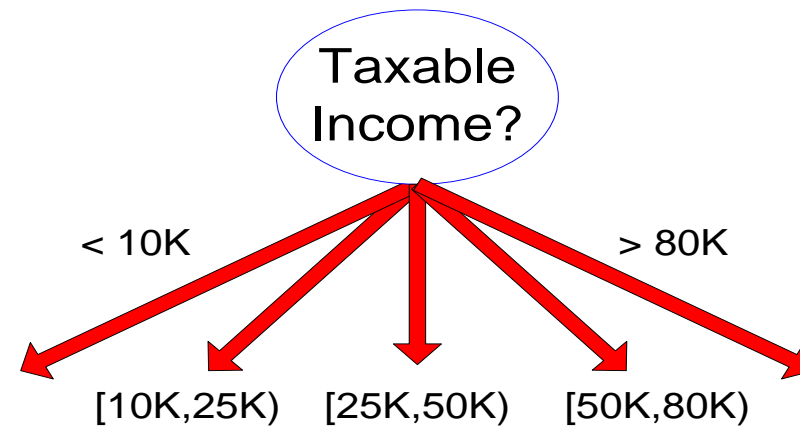
OR



# Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

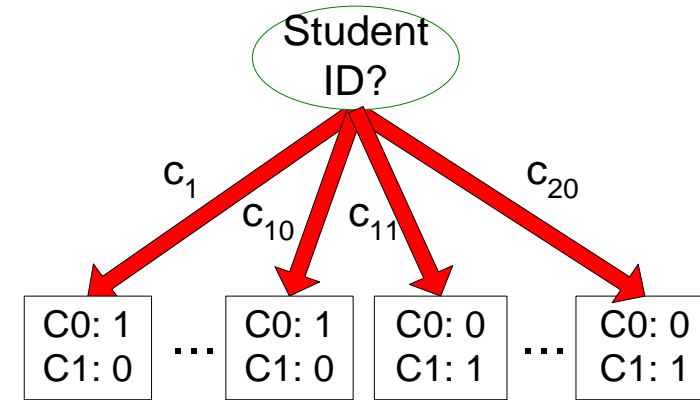
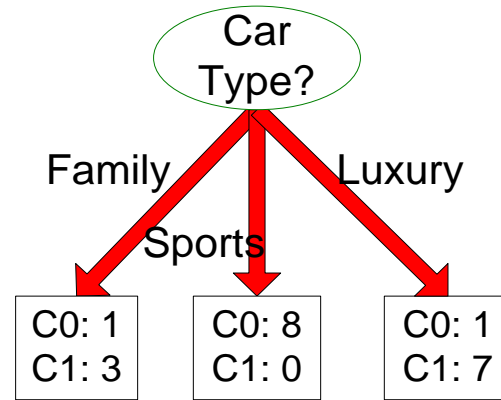
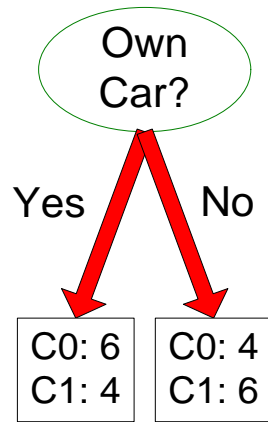
# Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- 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

**Before Splitting: 10 records of class 0,  
10 records of class 1**



**Which test condition is the best?**

**Which one should be the root node?**

# How to determine the Best Split

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

If all the elements belong to a single class, then it can be called **pure**.

C0: 5
C1: 5

**Non-homogeneous,  
High degree of impurity**

C0: 9
C1: 1

**Homogeneous,  
Low degree of impurity**

# Measure of Impurity: GINI

- Gini Index for a given node  $t$  :

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

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

The degree of Gini index varies between 0 and 0.5, where,

- 0 denotes that all elements belong to a certain class or if there exists only one class
- 0.5 denotes equally distributed elements into some classes.

C1	<b>0</b>
C2	<b>6</b>
<b>Gini=0.000</b>	

C1	<b>1</b>
C2	<b>5</b>
<b>Gini=0.278</b>	

C1	<b>2</b>
C2	<b>4</b>
<b>Gini=0.444</b>	

C1	<b>3</b>
C2	<b>3</b>
<b>Gini=0.500</b>	

# Examples for computing GINI

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

C1	<b>0</b>
C2	<b>6</b>

$$P(C1) = 0/6 = 0 \quad P(C2) = 6/6 = 1$$

$$GINI = 1 - P(C1)^2 - P(C2)^2 = 1 - 0 - 1 = 0$$

C1	<b>1</b>
C2	<b>5</b>

$$P(C1) = 1/6 \quad P(C2) = 5/6$$

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

C1	<b>2</b>
C2	<b>4</b>

$$P(C1) = 2/6 \quad P(C2) = 4/6$$

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

# Splitting Based on GINI

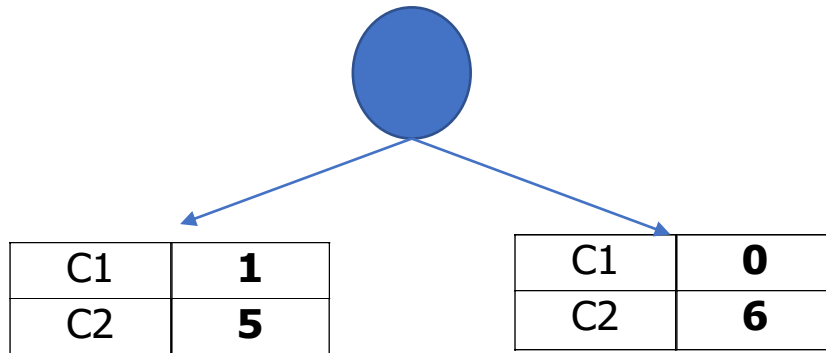
- When a node  $p$  is split into  $k$  partitions (children), the **quality of split** is computed by taking the weighted average of the individual Gini indexes,

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

where,

$n_i$  = number of records at child  $i$ ,

$n$  = number of records at node  $p$ .



# CART-algorithm

- In the decision tree, the nodes are split into sub-nodes on the basis of a threshold value of an attribute.
- The CART algorithm does that by searching for the best homogeneity for the sub-nodes, with the help of the Gini Index criterion.
  - The root node is taken as the training set and is split into two by considering the best attribute and threshold value.
  - Further, the subsets are also split using the same logic.
  - This continues till the last pure sub-set is found in the tree or the maximum number of leaves possible in that growing tree. This is also known as Tree Pruning.

# CART-greedy approach

- The input variables and the split points are selected through a greedy algorithm. Constructing a binary decision tree is a technique of splitting up the input space. A predetermined ending condition, such as a minimum number of training examples given to each leaf node of the tree, is used to halt tree building (**Stopping Criteria**).
- The input space is divided using the Greedy approach. This is known as recursive binary splitting. This is a numerical method in which all of the values are aligned and several split points are tried and assessed using a cost function, with the split with the lowest cost being chosen.

# CART- A Step by Step Example

Given,  
Planning Tennis  
Dataset

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Weak	Yes
14	Rain	Mild	High	Strong	No

**Task**  
Construct Decision  
Tree using CART.

Q: Which node (attribute) will be the root node?



# Outlook

Outlook is a nominal feature. It can be sunny, overcast or rain. I will summarize the final decisions for outlook feature.

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

$$\text{Gini(Outlook=Sunny)} = 1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 = 0.48$$
$$\text{Gini(Outlook=Overcast)} = 1 - (4/4)^2 - (0/4)^2 = 0$$
$$\text{Gini(Outlook=Rain)} = 1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 = 0.48$$

Then, we will calculate weighted sum of gini indexes for outlook feature.

$$\text{Gini(Outlook)} = (5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$$

$$GINI(t) = 1 - \sum_j [p(j | t)]^2$$
$$GINI_{split} = \sum_{i=1}^k \frac{n_i}{n} GINI(i)$$

## Temperature

Similarly, temperature is a nominal feature and it could have 3 different values: Cool, Hot and Mild. Let's summarize decisions for temperature feature.

Temperature	Yes	No	Number of instances
Hot	2	2	4
Cool	3	1	4
Mild	4	2	6

$$\text{Gini(Temp=Hot)} = 1 - (2/4)^2 - (2/4)^2 = 0.5$$
$$\text{Gini(Temp=Cool)} = 1 - (3/4)^2 - (1/4)^2 = 1 - 0.5625 - 0.0625 = 0.375$$
$$\text{Gini(Temp=Mild)} = 1 - (4/6)^2 - (2/6)^2 = 1 - 0.444 - 0.111 = 0.445$$

We'll calculate weighted sum of gini index for temperature feature

$$\text{Gini(Temp)} = (4/14) \times 0.5 + (4/14) \times 0.375 + (6/14) \times 0.445 = 0.142 + 0.107 + 0.190 = 0.439$$

# Humidity

Humidity is a binary class feature. It can be high or normal.

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

$$\text{Gini(Humidity=High)} = 1 - (3/7)^2 - (4/7)^2 = 1 - 0.183 - 0.326 = 0.489$$

$$\text{Gini(Humidity=Normal)} = 1 - (6/7)^2 - (1/7)^2 = 1 - 0.734 - 0.02 = 0.244$$

Weighted sum for humidity feature will be calculated next

$$\text{Gini(Humidity)} = (7/14) \times 0.489 + (7/14) \times 0.244 = 0.367$$

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

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

# Wind

Wind is a binary class similar to humidity. It can be weak and strong.

Wind	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

$$\text{Gini(Wind=Weak)} = 1 - (6/8)^2 - (2/8)^2 = 1 - 0.5625 - 0.062 = 0.375$$

$$\text{Gini(Wind=Strong)} = 1 - (3/6)^2 - (3/6)^2 = 1 - 0.25 - 0.25 = 0.5$$

$$\text{Gini(Wind)} = (8/14) \times 0.375 + (6/14) \times 0.5 = 0.428$$

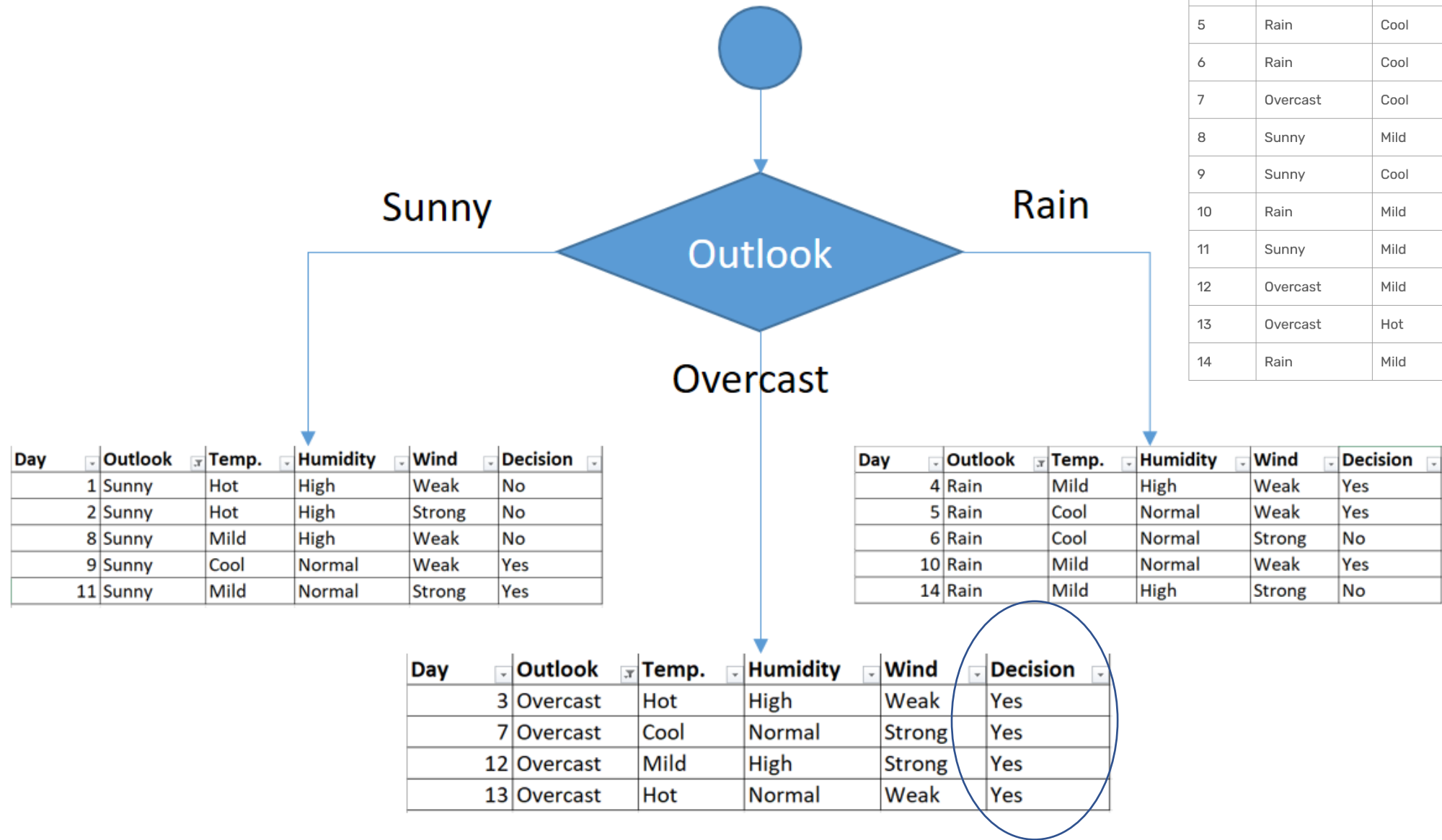
## Time to decide

We've calculated gini index values for each feature. The winner will be outlook feature because its cost is the lowest.

Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428

We'll put outlook decision at the top of the tree.

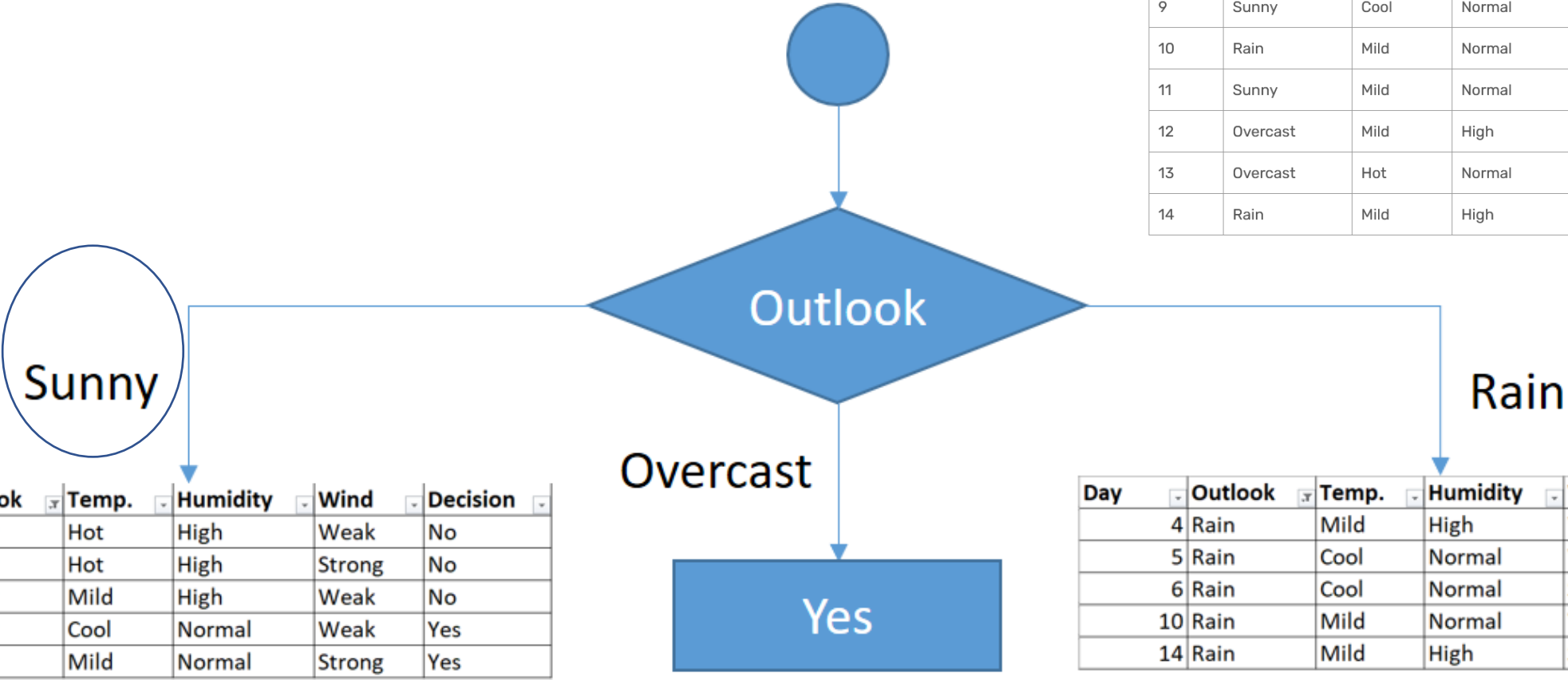
We'll put outlook decision at the top of the tree.



Day	Outlook	Temp.	Humidity	Wind	Decision
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	Weak	Yes
14	Rain	Mild	High	Strong	No

You might realize that sub dataset in the overcast leaf has only yes decisions. This means that overcast leaf is over.

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Weak	Yes
14	Rain	Mild	High	Strong	No



We will apply same principles to those sub datasets in the following steps.

Focus on the sub dataset for sunny outlook. We need to find the gini index scores for temperature, humidity and wind features respectively.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Weak	Yes
14	Rain	Mild	High	Strong	No

## Gini of temperature for sunny outlook

Temperature	Yes	No	Number of instances
Hot	0	2	2
Cool	1	0	1
Mild	1	1	2

$Gini(Outlook=Sunny \text{ and Temp.}=Hot) = 1 - (0/2)^2 - (2/2)^2 = 0$

$Gini(Outlook=Sunny \text{ and Temp.}=Cool) = 1 - (1/1)^2 - (0/1)^2 = 0$

$Gini(Outlook=Sunny \text{ and Temp.}=Mild) = 1 - (1/2)^2 - (1/2)^2 = 1 - 0.25 - 0.25 = 0.5$

$Gini(Outlook=Sunny \text{ and Temp.}) = (2/5) \times 0 + (1/5) \times 0 + (2/5) \times 0.5 = 0.2$

## Gini of wind for sunny outlook

Wind	Yes	No	Number of instances
Weak	1	2	3
Strong	1	1	2

$Gini(Outlook=Sunny \text{ and Wind}=Weak) = 1 - (1/3)^2 - (2/3)^2 = 0.266$

$Gini(Outlook=Sunny \text{ and Wind}=Strong) = 1 - (1/2)^2 - (1/2)^2 = 0.2$

$Gini(Outlook=Sunny \text{ and Wind}) = (3/5) \times 0.266 + (2/5) \times 0.2 = 0.466$

## Gini of humidity for sunny outlook

Humidity	Yes	No	Number of instances
High	0	3	3
Normal	2	0	2

$Gini(Outlook=Sunny \text{ and Humidity}=High) = 1 - (0/3)^2 - (3/3)^2 = 0$

$Gini(Outlook=Sunny \text{ and Humidity}=Normal) = 1 - (2/2)^2 - (0/2)^2 = 0$

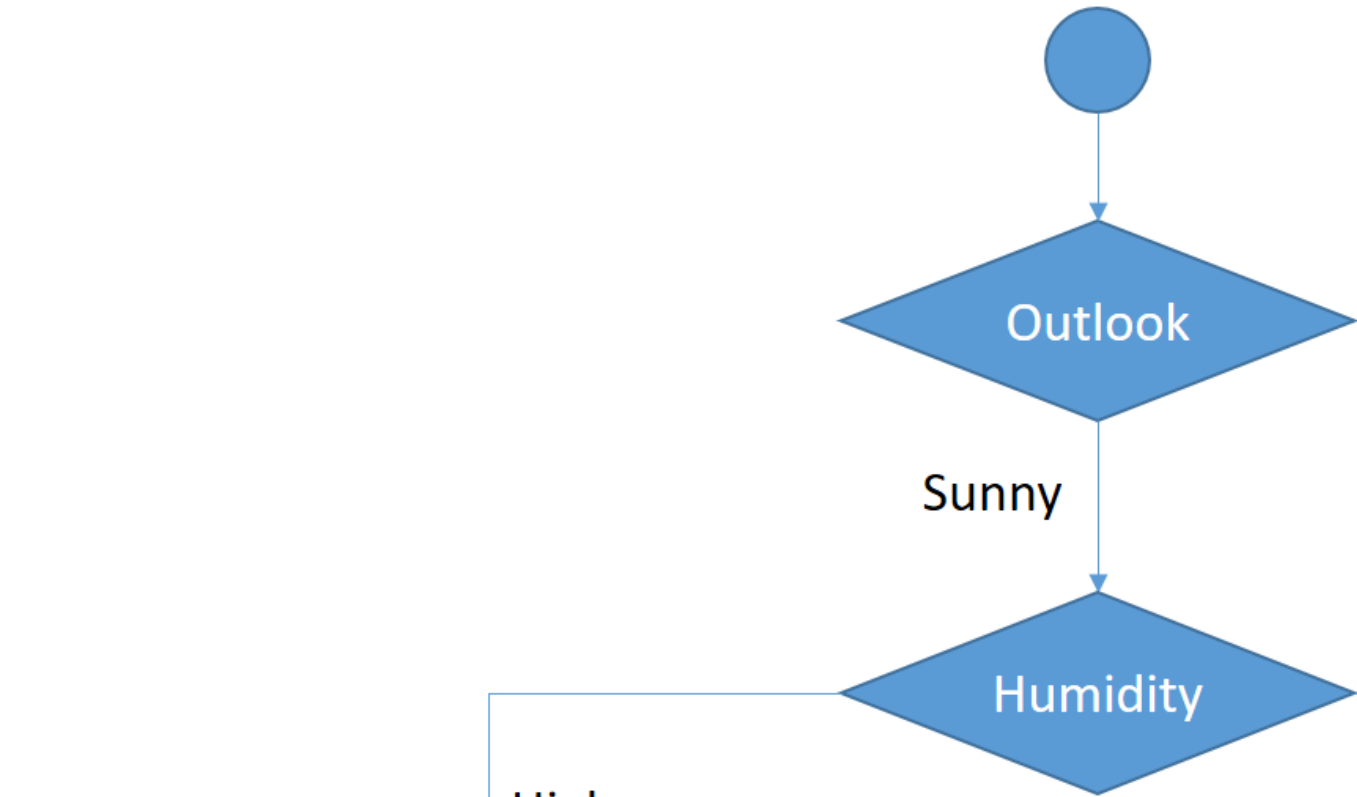
$Gini(Outlook=Sunny \text{ and Humidity}) = (3/5) \times 0 + (2/5) \times 0 = 0$

## Decision for sunny outlook

We’ve calculated gini index scores for feature when outlook is sunny. The winner is humidity because it has the lowest value.

Feature	Gini index
Temperature	0.2
Humidity	0
Wind	0.466

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Weak	Yes
14	Rain	Mild	High	Strong	No



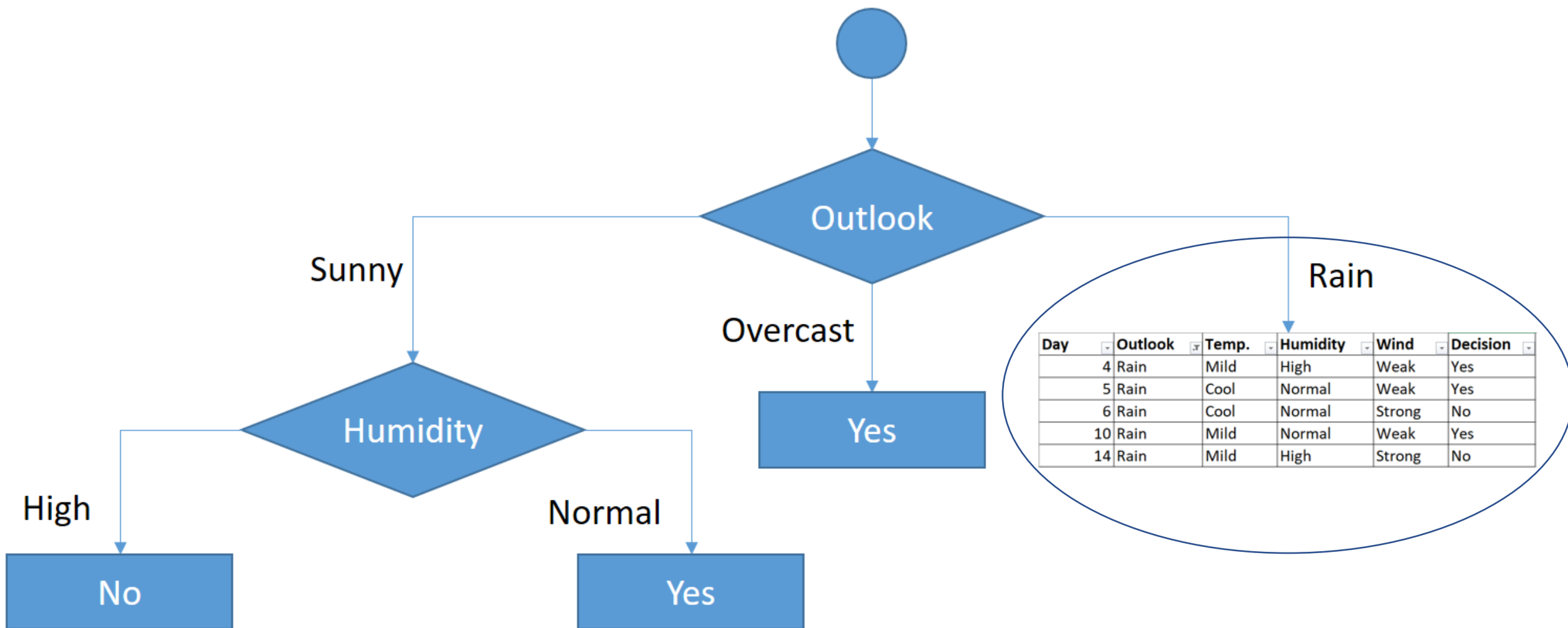
Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes



As seen, decision is always no for high humidity and sunny outlook.

On the other hand, decision will always be yes for normal humidity and sunny outlook. This branch is over.



Now, we need to focus on rain outlook.

## Rain outlook

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Weak	Yes
14	Rain	Mild	High	Strong	No

## Gini of temprature for rain outlook

Temperature	Yes	No	Number of instances
Cool	1	1	2
Mild	2	1	3

$Gini(Outlook=Rain \text{ and Temp.}=Cool) = 1 - (1/2)^2 - (1/2)^2 = 0.5$

$Gini(Outlook=Rain \text{ and Temp.}=Mild) = 1 - (2/3)^2 - (1/3)^2 = 0.444$

$Gini(Outlook=Rain \text{ and Temp.}) = (2/5) \times 0.5 + (3/5) \times 0.444 = 0.466$

## Gini of wind for rain outlook

Wind	Yes	No	Number of instances
Weak	3	0	3
Strong	0	2	2

$Gini(Outlook=Rain \text{ and Wind}=Weak) = 1 - (3/3)^2 - (0/3)^2 = 0$

$Gini(Outlook=Rain \text{ and Wind}=Strong) = 1 - (0/2)^2 - (2/2)^2 = 0$

$Gini(Outlook=Rain \text{ and Wind}) = (3/5) \times 0 + (2/5) \times 0 = 0$

## Gini of humidity for rain outlook

Humidity	Yes	No	Number of instances
High	1	1	2
Normal	2	1	3

$Gini(Outlook=Rain \text{ and Humidity}=High) = 1 - (1/2)^2 - (1/2)^2 = 0.5$

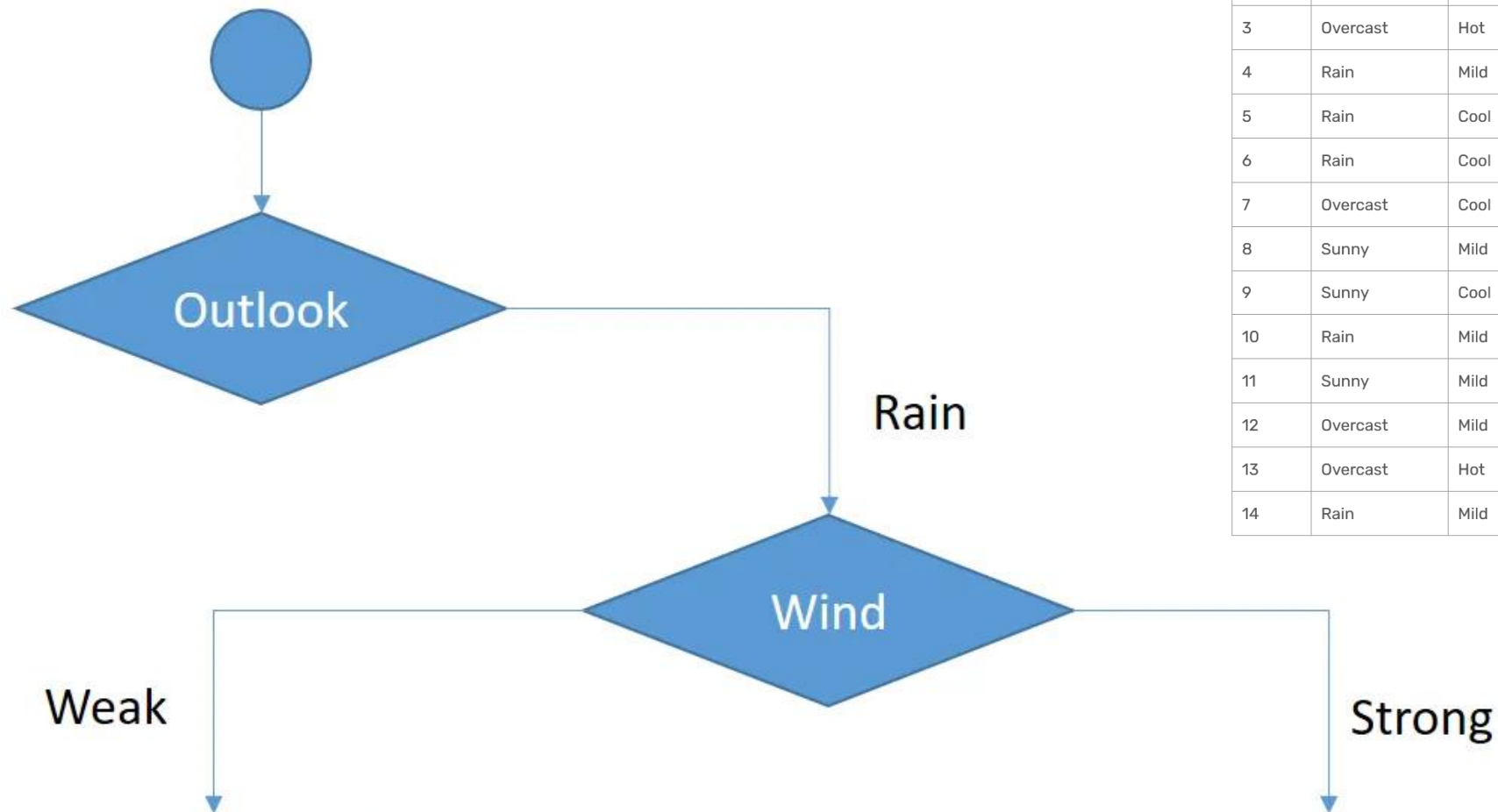
$Gini(Outlook=Rain \text{ and Humidity}=Normal) = 1 - (2/3)^2 - (1/3)^2 = 0.444$

$Gini(Outlook=Rain \text{ and Humidity}) = (2/5) \times 0.5 + (3/5) \times 0.444 = 0.466$

## Decision for rain outlook

The winner is wind feature for rain outlook because it has the minimum gini index score in features.

Feature	Gini index
Temperature	0.466
Humidity	0.466
Wind	0

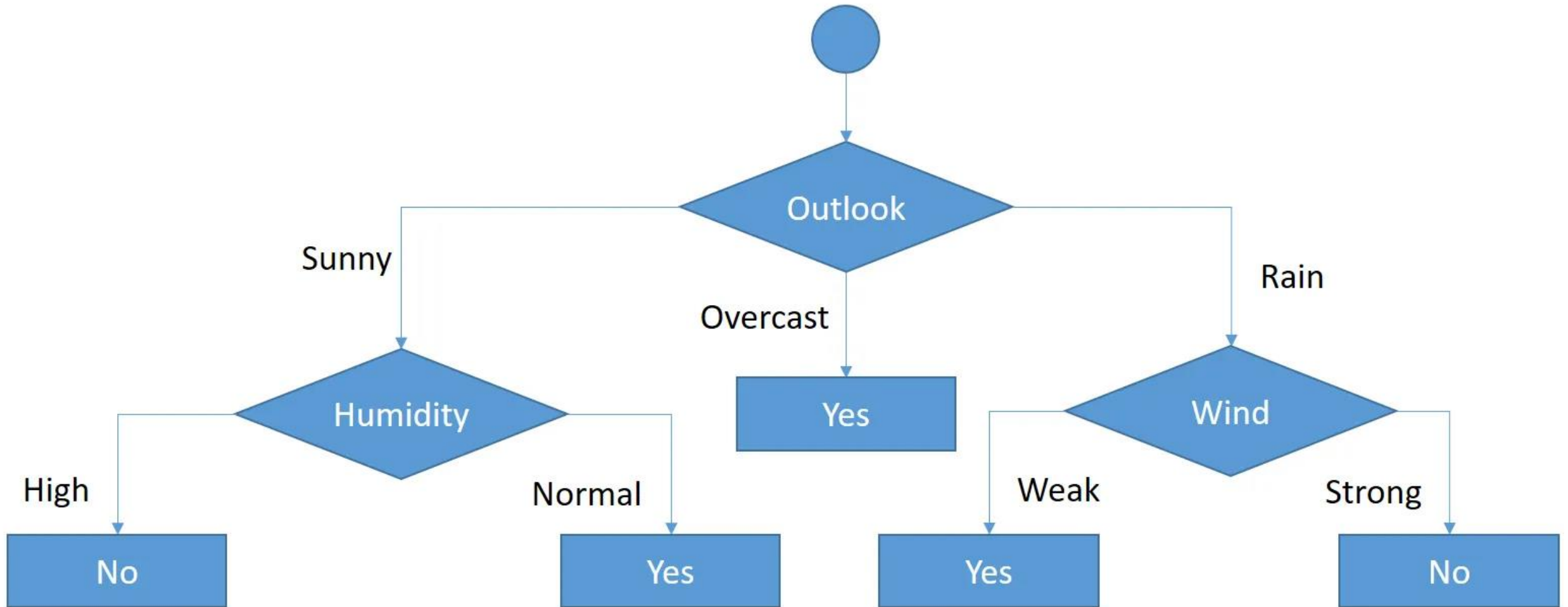


Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
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	Weak	Yes
14	Rain	Mild	High	Strong	No

As seen, decision is always yes when wind is weak. On the other hand, decision is always no if wind is strong. This means that this branch is over.



# How to define a node for Numeric Data?

	Weight	Heart Disease
Lowest	155	No
	180	Yes
	190	No
	220	Yes
Highest	225	Yes

Step 1) Sort the patients by weight, lowest to highest.

Weight	Heart Disease
155	No
<b>167.5</b>	
180	Yes
<b>185</b>	
190	No
<b>205</b>	
220	Yes
<b>222.5</b>	
225	Yes

Step 2) Calculate the average weight for all adjacent patients.

Weight	Heart Disease
155	No
<b>167.5</b>	
180	Yes
<b>185</b>	
190	No
<b>205</b>	
220	Yes
<b>222.5</b>	
225	Yes

Step 3) Calculate the impurity values for each average weight.

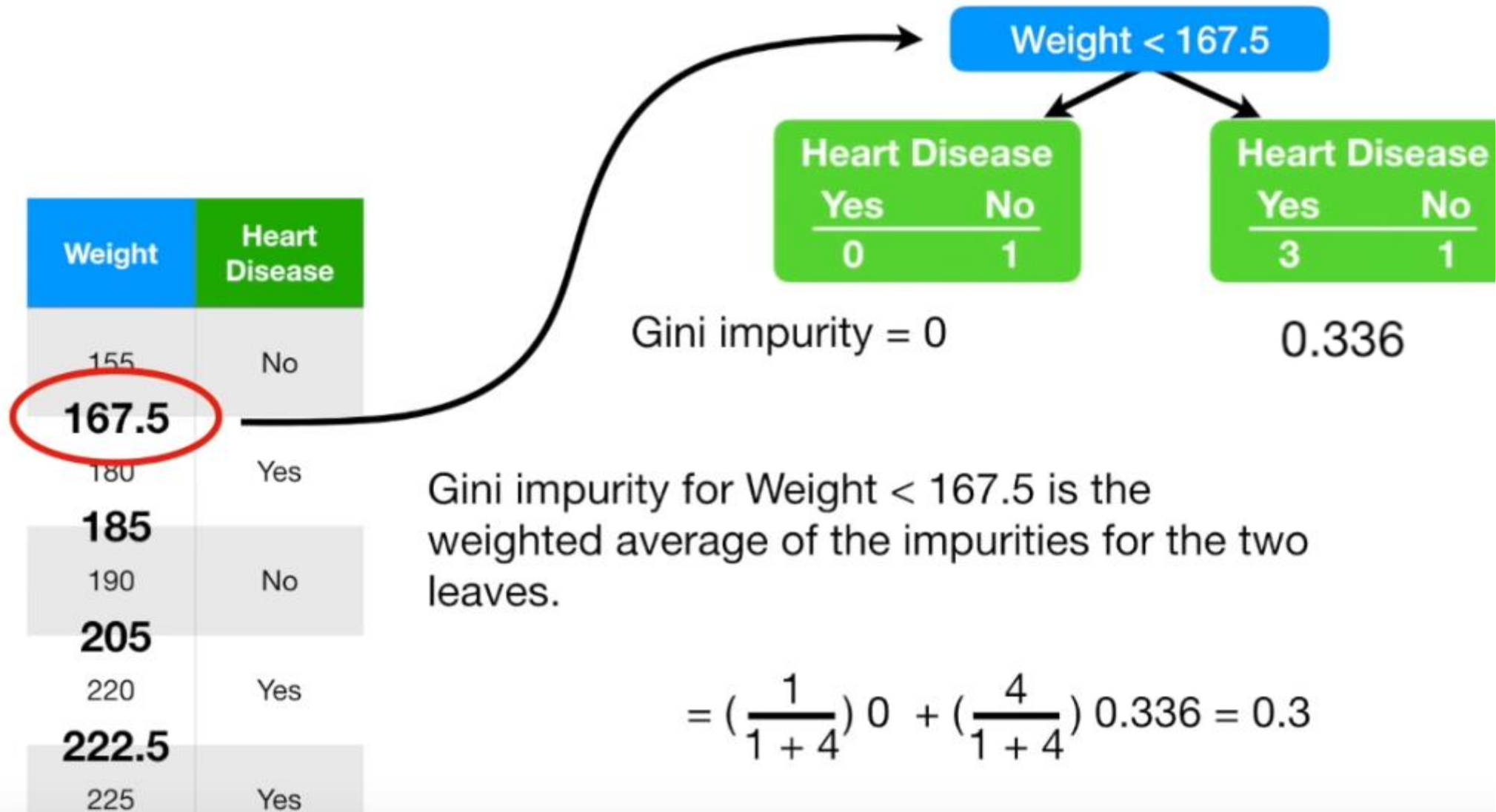
Gini impurity = ?

Gini impurity = ?

Gini impurity = ?

Gini impurity = ?





Weight	Heart Disease
155	No
<b>167.5</b>	
180	Yes
<b>185</b>	
190	No
<b>205</b>	
220	Yes
<b>222.5</b>	
225	Yes

→ Gini impurity = 0.3

→ Gini impurity = 0.47

→ Gini impurity = 0.27

→ Gini impurity = 0.4

The lowest impurity occurs when we separate using **weight < 205...**

...so this is the cutoff and impurity value we will use when we compare weight to chest pain or blocked arteries.