# Decision Tree Classifier

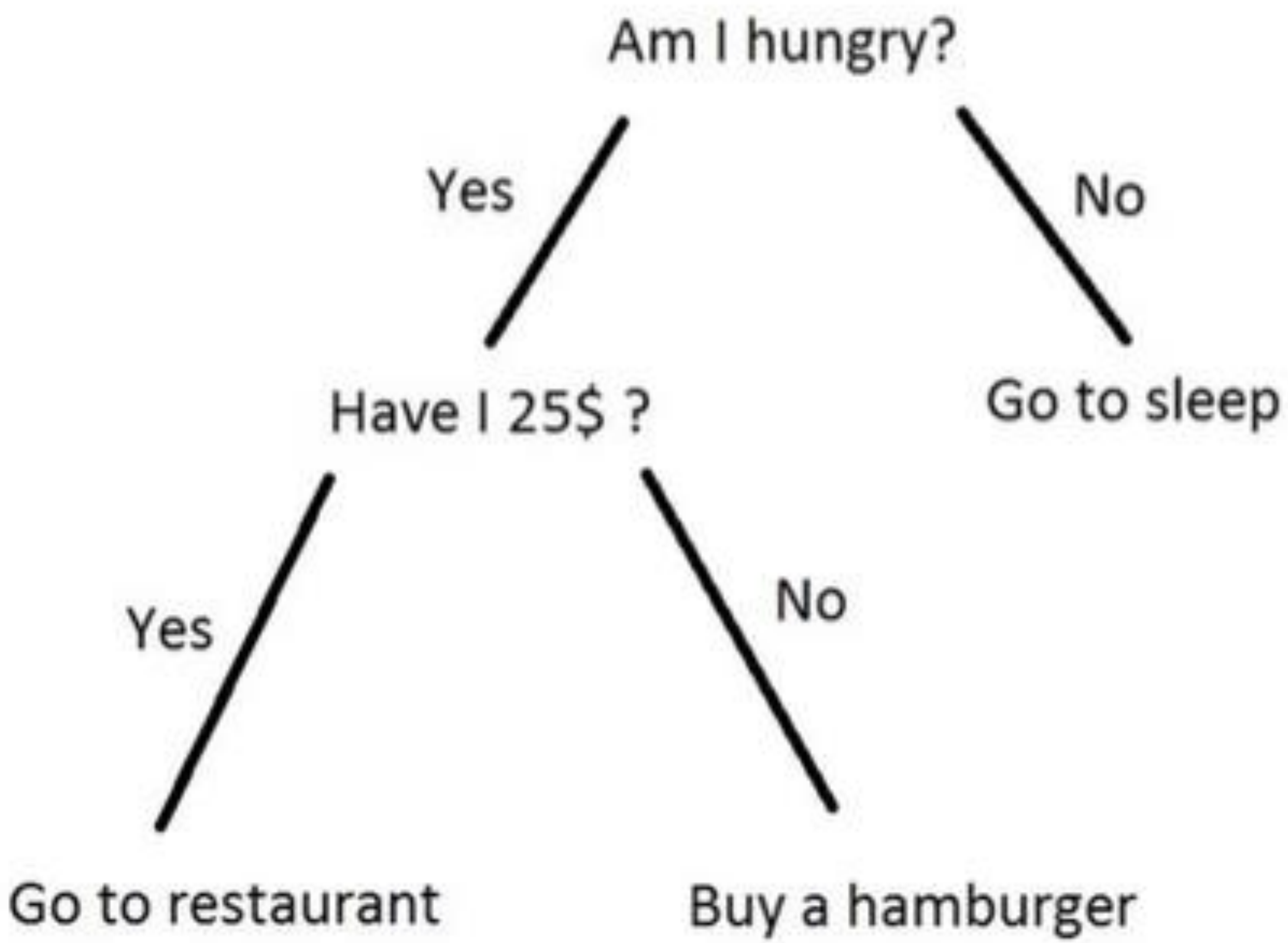
BCSE 0105 MACHINE LEARNING

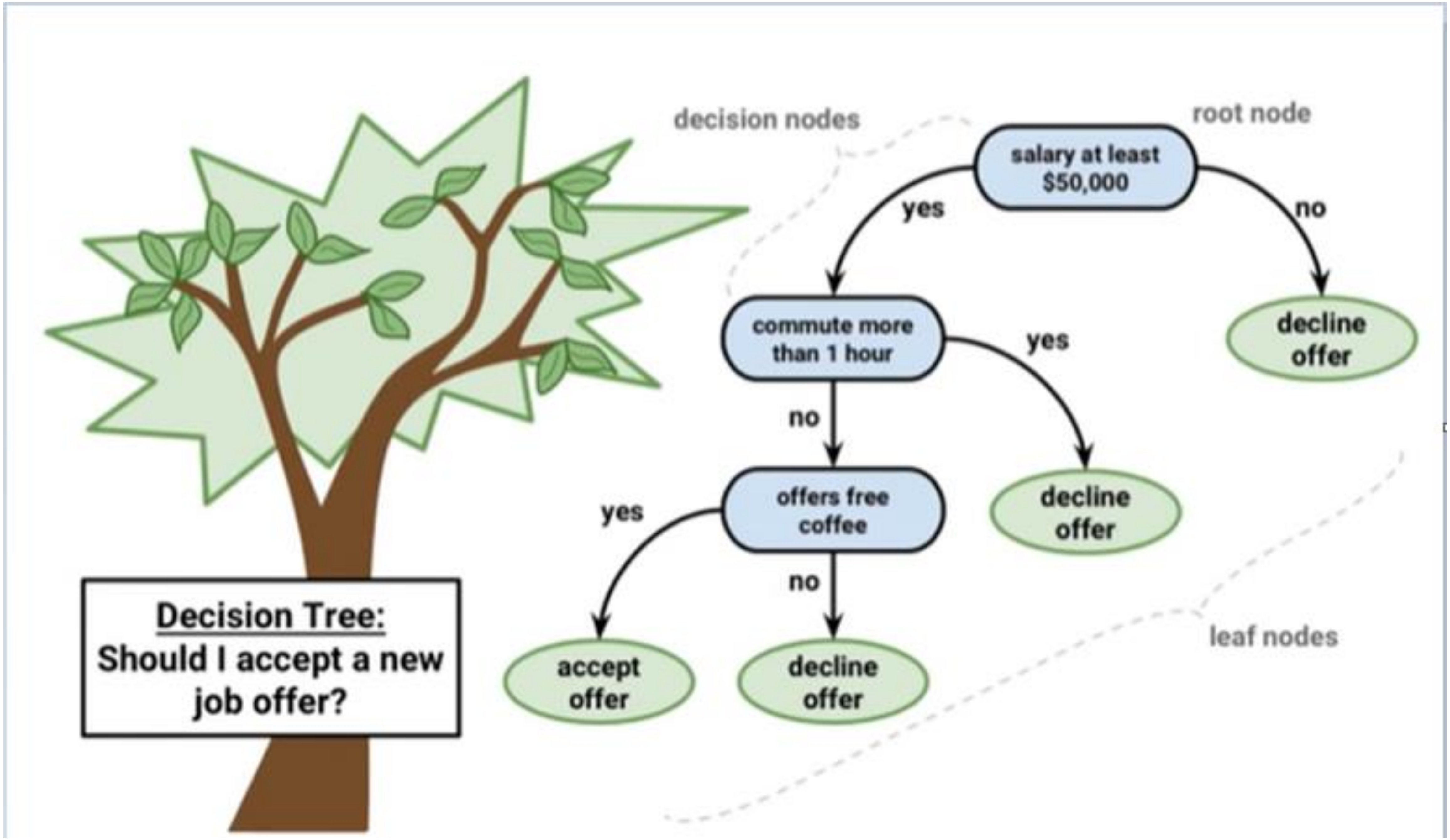
# Introduction

"A **decision tree** is a graphical representation of all the possible solutions to a decision based on certain conditions"

## Decision Tree

- Graphical representation of all the possible solutions to a decision
- Decisions are based on some conditions
- Decision made can be easily explained

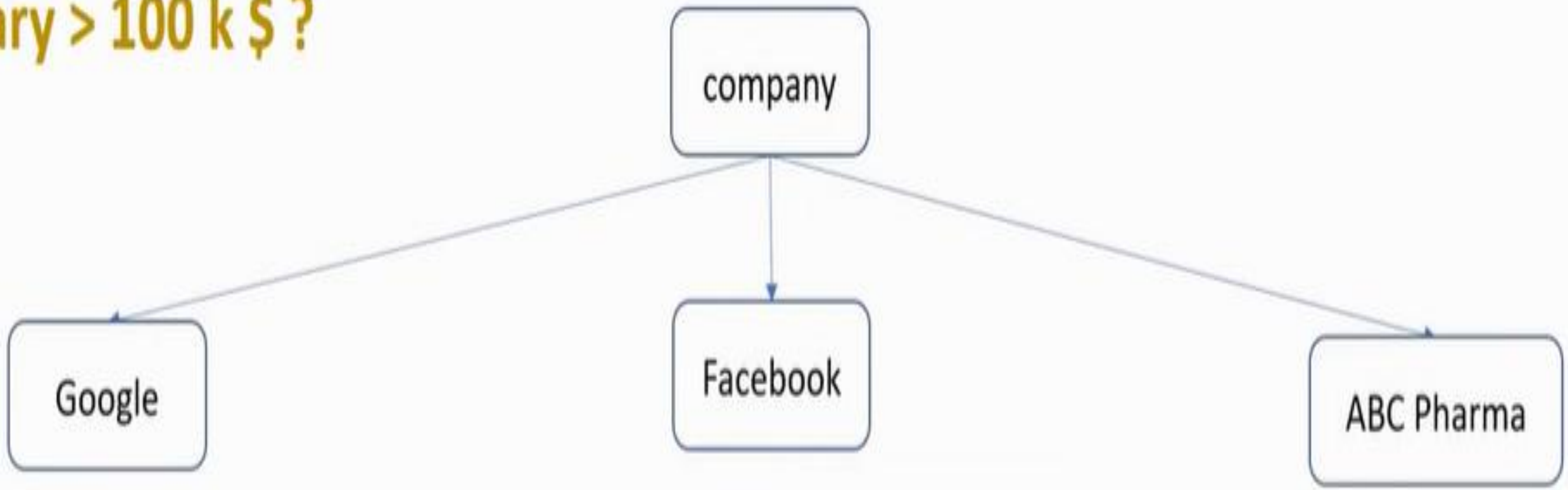

Am I hungry?
Yes
No
Have I 25$ ?
Go to sleep
Yes
No
Go to restaurant
Buy a hamburger



decision nodes
root node
salary at least $50,000
yes
no
commute more than 1 hour
yes
decline offer
no
offers free coffee
decline offer
yes
no
accept offer
decline offer
leaf nodes
Decision Tree:
Should I accept a new job offer?

| Company | Job | Degree | Salary_more_then_100k |
|---|---|---|---|
| google | sales executive | bachelors | 0 |
| google | sales executive | masters | 0 |
| google | business manager | bachelors | 1 |
| google | business manager | masters | 1 |
| google | computer programmer | bachelors | 0 |
| google | computer programmer | masters | 1 |
| abc pharma | sales executive | masters | 0 |
| abc pharma | computer programmer | bachelors | 0 |
| abc pharma | business manager | bachelors | 0 |
| abc pharma | business manager | masters | 1 |
| facebook | sales executive | bachelors | 1 |
| facebook | sales executive | masters | 1 |
| facebook | business manager | bachelors | 1 |
| facebook | business manager | masters | 1 |
| facebook | computer programmer | bachelors | 1 |
| facebook | computer programmer | masters | 1 |

# Salary > 100 k $ ?




| | | |
|---|---|---|
| google | sales executive | bachelors |
| google | sales executive | masters |
| google | business manager | bachelors |
| google | business manager | masters |
| google | computer programmer | bachelors |
| google | computer programmer | masters |

?

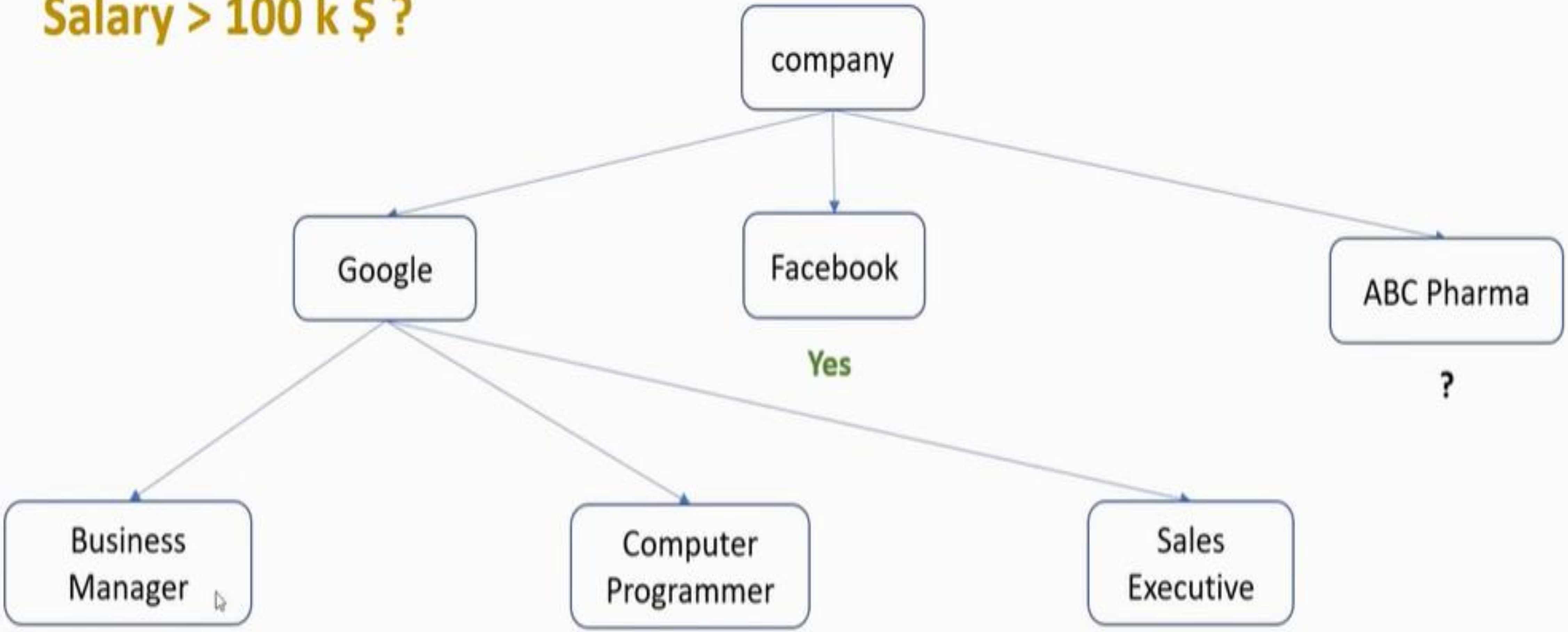
| | | |
|---|---|---|
| facebook | sales executive | bachelors |
| facebook | sales executive | masters |
| facebook | business manager | bachelors |
| facebook | business manager | masters |
| facebook | computer programmer | bachelors |
| facebook | computer programmer | masters |

Yes

| | | |
|---|---|---|
| abc pharma | sales executive | masters |
| abc pharma | computer programmer | bachelors |
| abc pharma | business manager | bachelors |
| abc pharma | business manager | masters |

?

# Salary > 100 k $ ?




| | | |
|---|---|---|
| google | business manager | bachelors |
| google | business manager | masters |

Yes

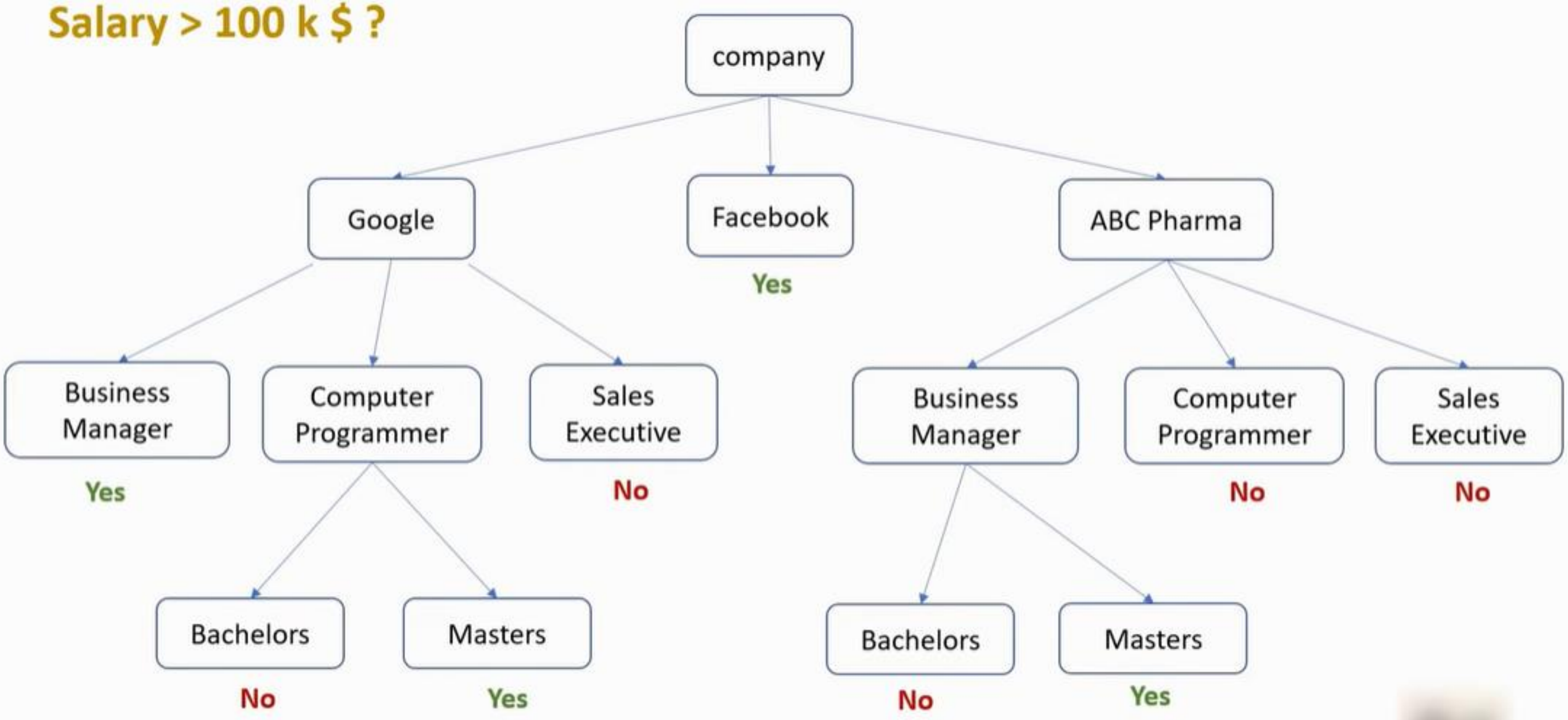
| | | |
|---|---|---|
| google | computer programmer | bachelors |
| google | computer programmer | masters |

?

| | | |
|---|---|---|
| google | sales executive | bachelors |
| google | sales executive | masters |

No

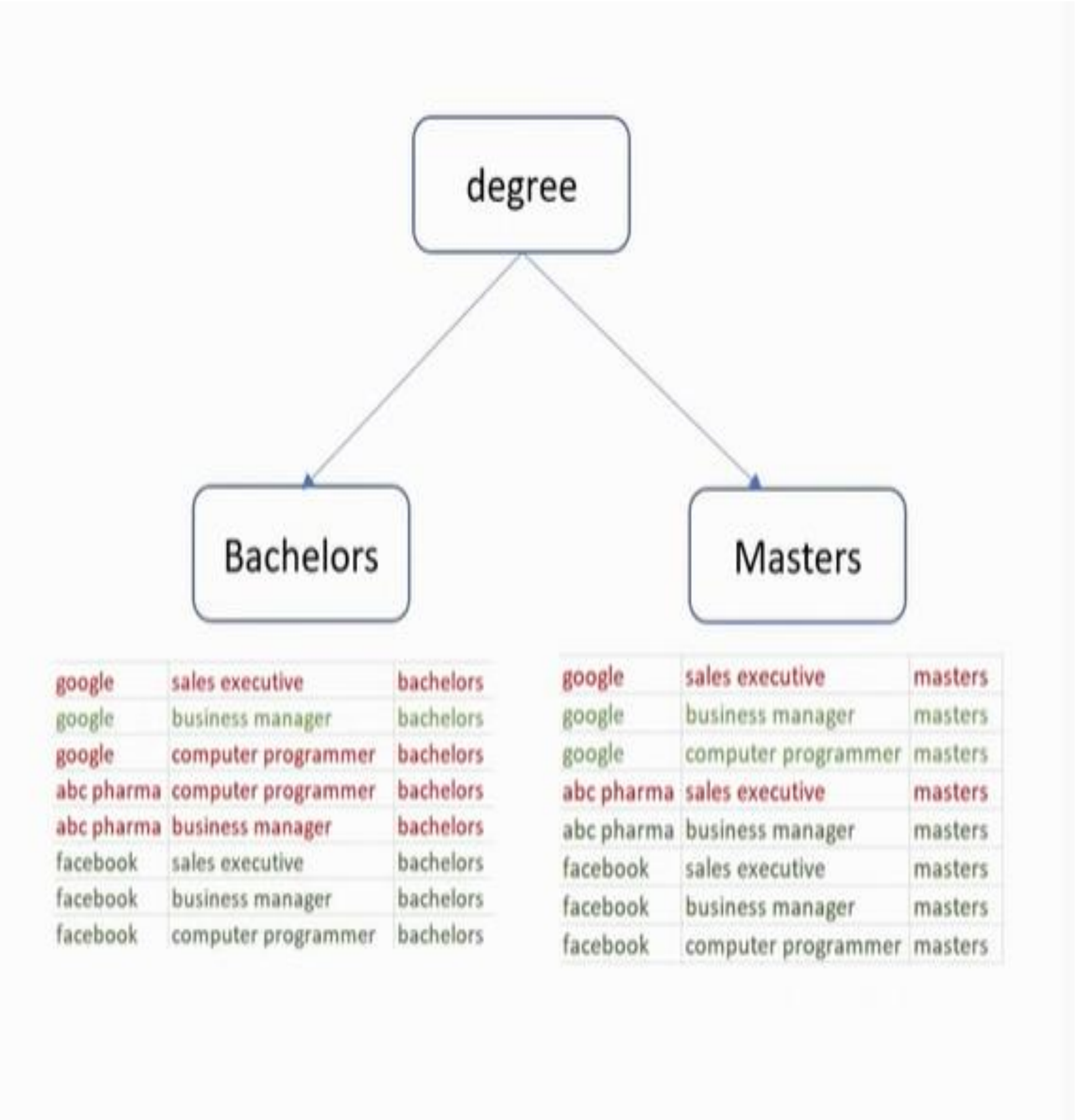
## Salary > 100 k $ ?


company
Google
Facebook
ABC Pharma
Yes
Business Manager
Computer Programmer
Sales Executive
Business Manager
Computer Programmer
Sales Executive
Yes
No
No
No
Bachelors
Masters
Bachelors
Masters
No
Yes
No
Yes

- It matters in which order you have split the tree

- Which feature to be chosen first will effect the performance of the algorithm

# Alternatively



Classification Problem: Whether John will play Tennis or not?

| Day | Outlook | Temperature | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| 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 |

# To make a decision tree:
# Need to select a feature as root node

Which one among them should you pick first?

**Answer:** Determine the attribute that best classifies the training data

But How do we choose the best attribute?

Or

How does a tree decide where to split?

# How Does A Tree Decide Where To Split?

- Entropy (to measure the impurity)
- Information Gain (decision tree split)
- Gini Index/Gini Impurity

# Algorithms to construct Decision Tree

- ID3 Algorithm (using entropy and Information Gain)
- CART Algorithm (using Gini Index/Gini Impurity)

# Entropy

- Entropy can be defined as **a measure of the purity of the split** or measure of impurity of a node.
- Entropy always lies between 0 to 1.
- A node having multiple classes is impure whereas a node having only one class is pure.

$$\textbf{Entropy(s)} = - P(yes)\log_2 P(yes) - P(no)\log_2 P(no)$$

Where,

- S is the total sample space,
- P(yes) is probability of yes

# Equal yes and no

$E(S) = -P(Yes)\log_2 P(Yes)$

When $P(Yes) = P(No) = 0.5$ ie $YES + NO = Total\ Sample(S)$

$E(S) = 0.5\ \log_2 0.5 - 0.5\ \log_2 0.5$

$E(S) = 0.5(\log_2 0.5 -\ \log_2 0.5)$

$E(S) = 1$

# All yes or All no

$E(S) = -P(Yes)\log_2 P(Yes)$

When $P(Yes) = 1$ ie YES = Total Sample(S)

$E(S) = 1 \ \log_2 1$

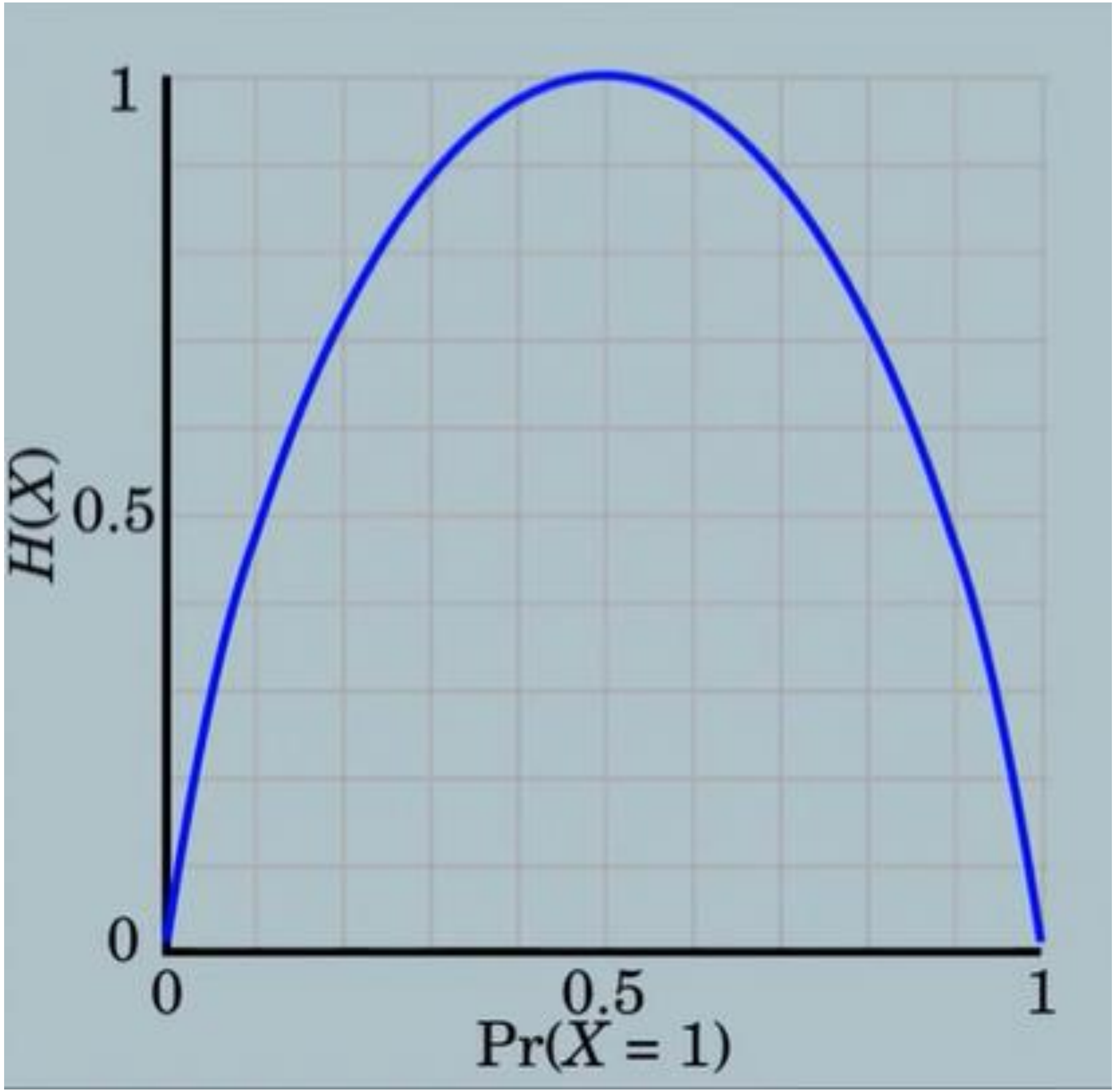
$E(S) = 0$

$E(S) = -P(No)\log_2 P(No)$

When $P(No) = 1$ ie No = Total Sample(S)

$E(S) = 1 \ \log_2 1$

$E(S) = 0$



1
0.5
0
H(X)
0
0.5
1
Pr(X = 1)

| Day | Outlook | Temperature | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| 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 |

# Entropy of independent categorical attribute/after splitting , (eg.  E(Outlook))

$$E(A) = \sum \frac{|S_v|}{|S|} Entropy(S_v)$$

Where

A is attribute (eg. Outlook)

v is values of attribute A (eg. sunny, overcast, rainy)

$\frac{|S_v|}{|S|}$ is probability of v i.e. P(v)

# Eg. E(Outlook)

E (Outlook)=P(Sunny)*E(Sunny)+P(Overcast)*E(Overcast)+P(Rainy)*E(Rainy)

= P(Sunny)*E(3,2)+P(Overcast)*E(4,0)+P(Rainy)*E(2,3)

= $5/14 \times 0.971 + 4/14 \times 0 + 5/14 \times 0.971$

= 0.693

# Information Gain

- Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification.
- Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.
- Information gain computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values.
- ID3 (Iterative Dichotomiser) decision tree algorithm uses information gain to split a node.

# Information Gain

$$Gain(S,A) = Entropy(S) - \sum \frac{|S_v|}{|S|} Entropy(S_v),$$

where

S is the given set or sample space

Or, Gain(S,A)=E(S)-E(A)

Eg.

Gain(S, Outlook)=E(S)-E(Outlook)= $0.94 - 0.693 = 0.247$

# Let’s Build Our Decision Tree

Construct a decision tree of the given training set using ID3 algorithm

| Day | Outlook | Temperature | Humidity | Wind | Play Tennis |
|---|---|---|---|---|---|
| 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 |

# Step 1: Compute the entropy for the Data set

Out of 14 instances we have 9 YES and 5 NO

*So we have the formula,*

$E(S) = -P(Yes) \log_2 P(Yes) - P(No) \log_2 P(No)$

$E(S) = -(9/14)* \log_2 9/14 - (5/14)* \log_2 5/14$

$E(S) = 0.41+0.53 = 0.94$

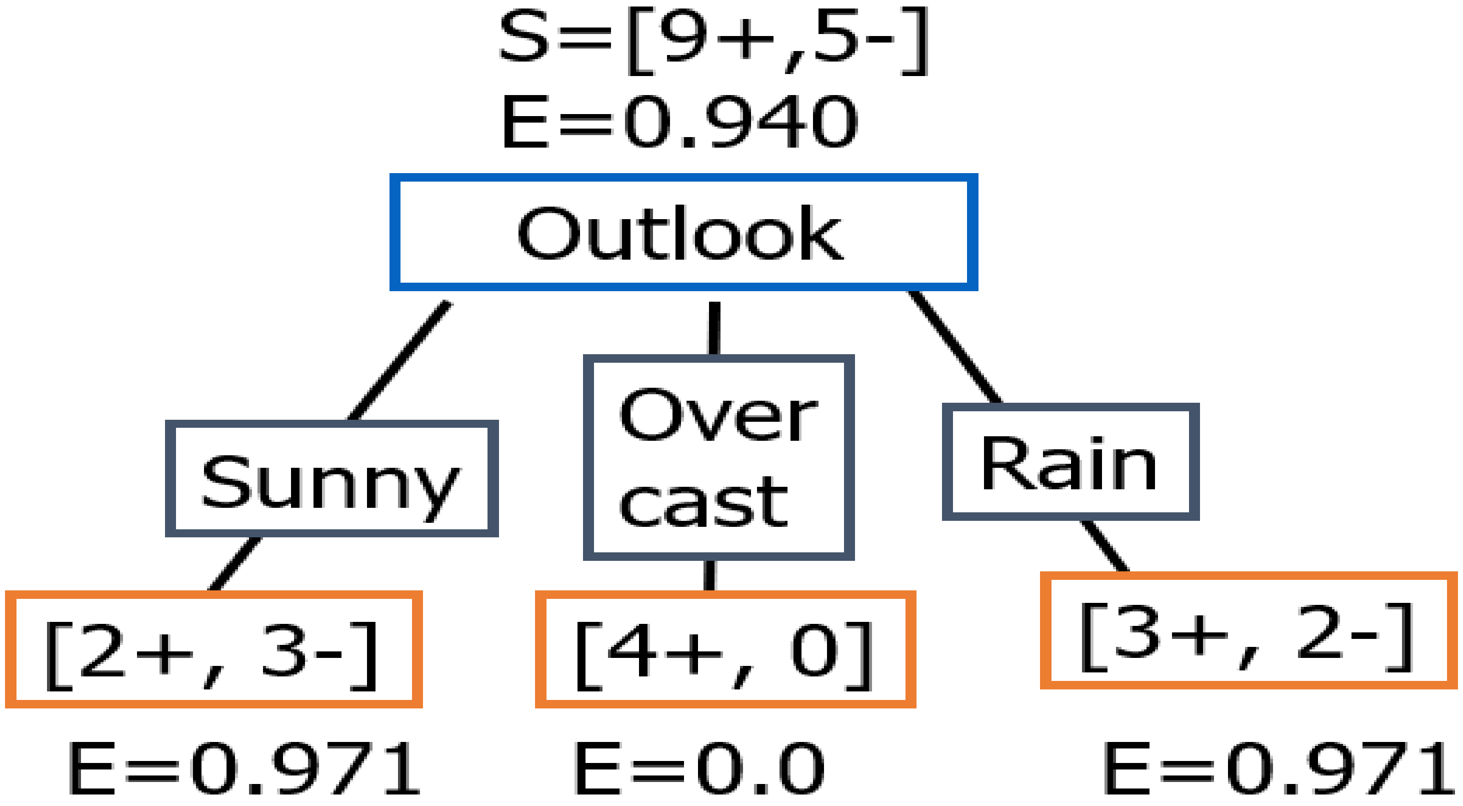
# Which Node To Select As Root Node?

Outlook ?
Humidity ?
Temperature ?
Wind ?

# # First iteration

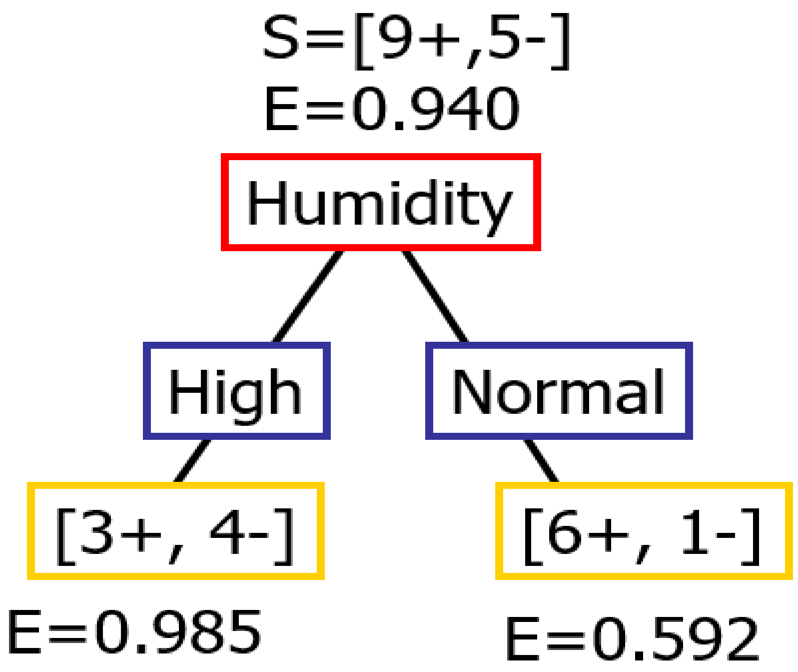
- At first iteration, we need to know which is best attribute to be chosen as top root in our decision tree.
- To do that, ID3 will find the best attribute which has maximum information gain.
- The information gain for each attribute:
- attributes = [Outlook, Humidity, Wind, Temperature]
- G(S, Outlook) = 0.247
- G(S, Humidity) = ?
- G(S, Wind) = ?
- G(S, Temperature) = ?

# Information Gain of Outlook

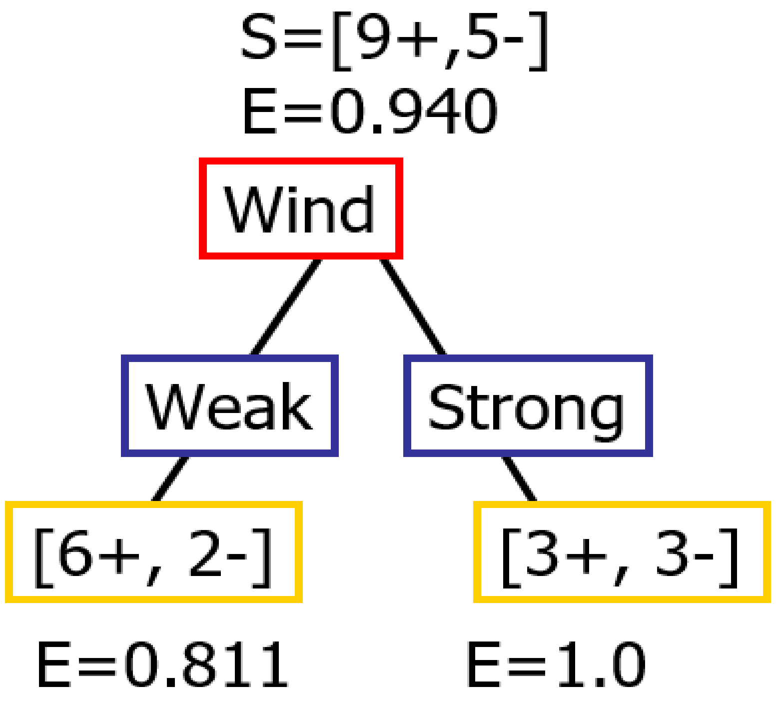



Gain(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247

# Information Gain of Humidity, windy




Gain(S,Humidity)
=0.940-(7/14)*0.985
– (7/14)*0.592
=0.151




Gain(S,Wind)
=0.940-(8/14)*0.811
– (6/14)*1.0
=0.048

Humidity provides greater info. gain than Wind, w.r.t target classification.

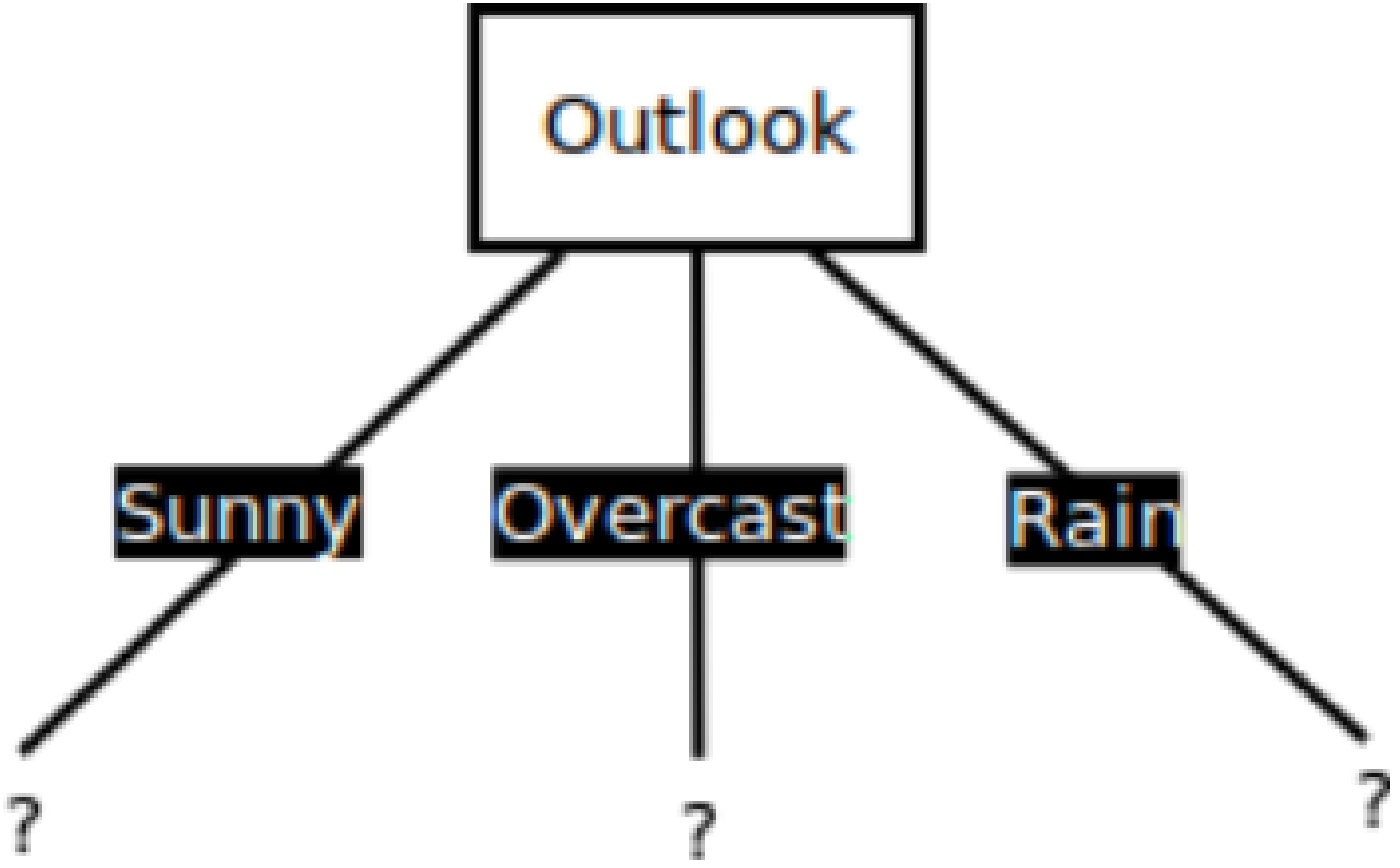
**Similarly, we can find the Information Gain of Temperature**

# # First iteration

- At first iteration, we need to know which is best attribute to be chosen as top root in our decision tree.
- To do that, ID3 will find the best attribute which is has maximum information gain.
- The information gain for each attribute:
- Gain(S,Outlook) =0.247
- Gain(S,Humidity) =0.151
- Gain(S,Wind) =0.048
- Gain(S,Temperature) =0.029

# # First iteration

- So, based on the information gains calculated
- we choose attribute Outlook as root node which has three branches Sunny, Rain, and Overcast.
- Our decision tree will look like an image below



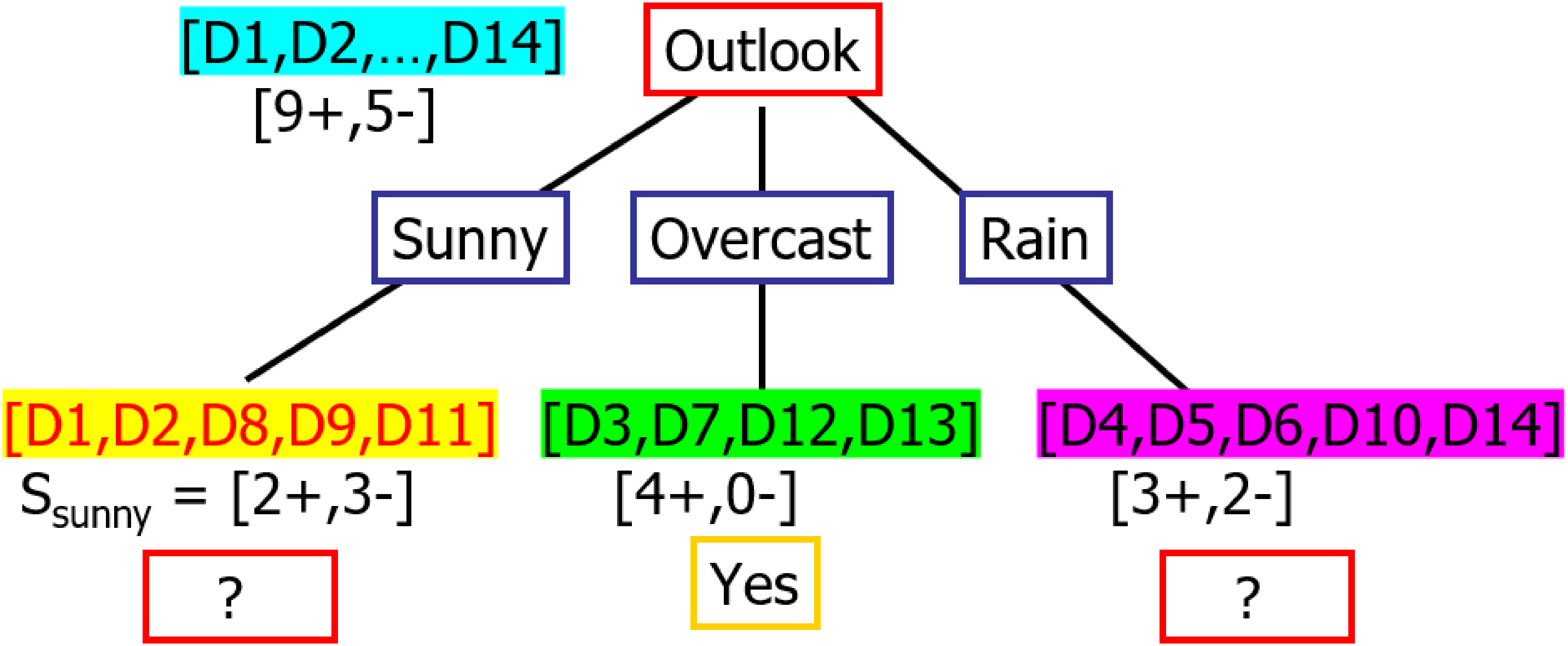

Initial decision tree

# # Second iteration

- Now we remove Outlook from attribute list.
- We want examine which is the best attribute for branch of Sunny.
- Remember, that new S is rows containing values of Sunny.

Now, attributes left are [Humidity, Wind, Temperature]

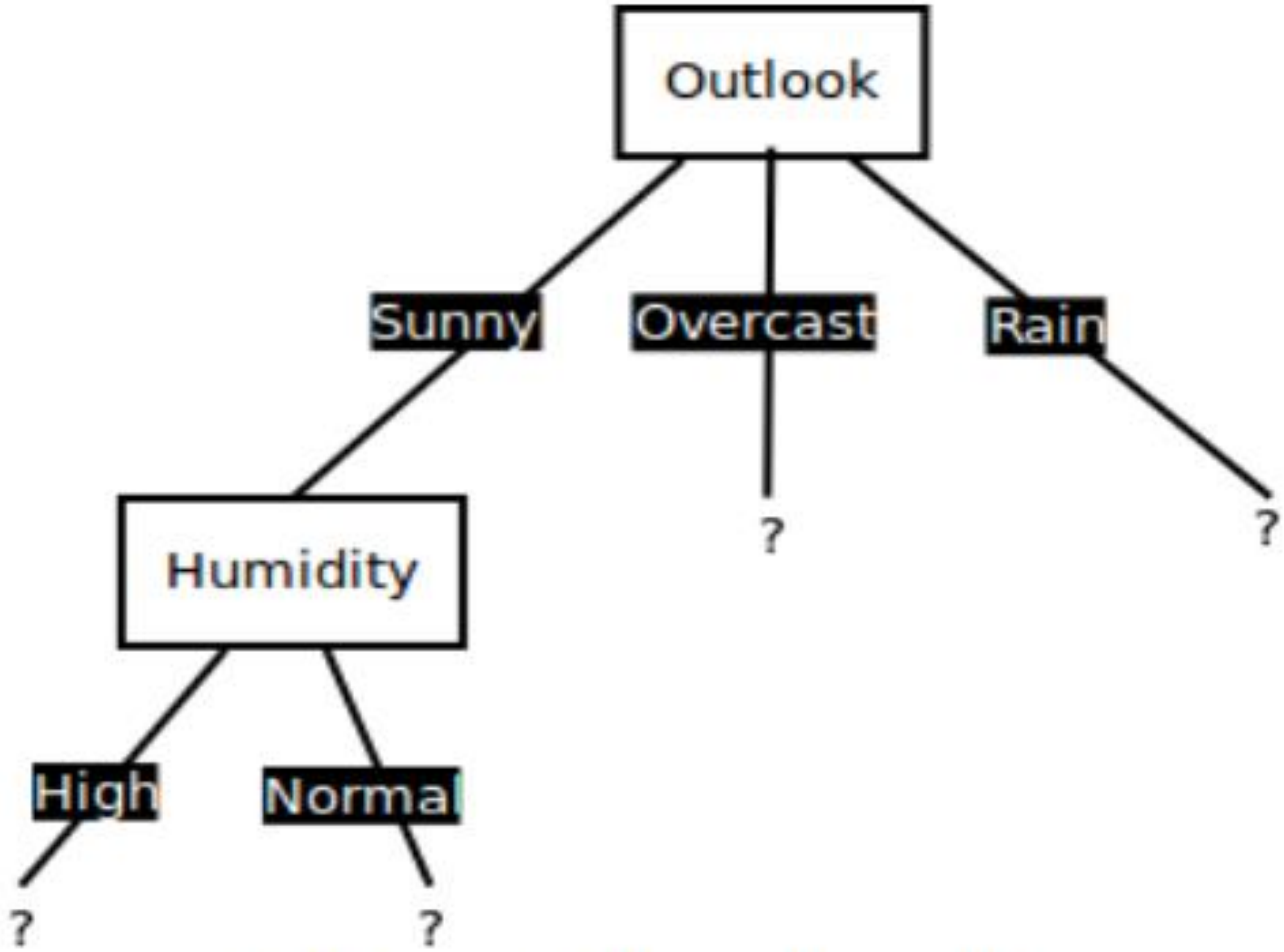
# ID3 Algorithm




Gain($S_{sunny}$, Humidity)=0.970-(3/5)0.0 – 2/5(0.0) = 0.970
Gain($S_{sunny}$, Temperature)=0.970-(2/5)0.0 –2/5(1.0)-(1/5)0.0 = 0.570
Gain($S_{sunny}$, Wind)=0.970 -(2/5)1.0 – 3/5(0.918) = 0.019

# Second iteration

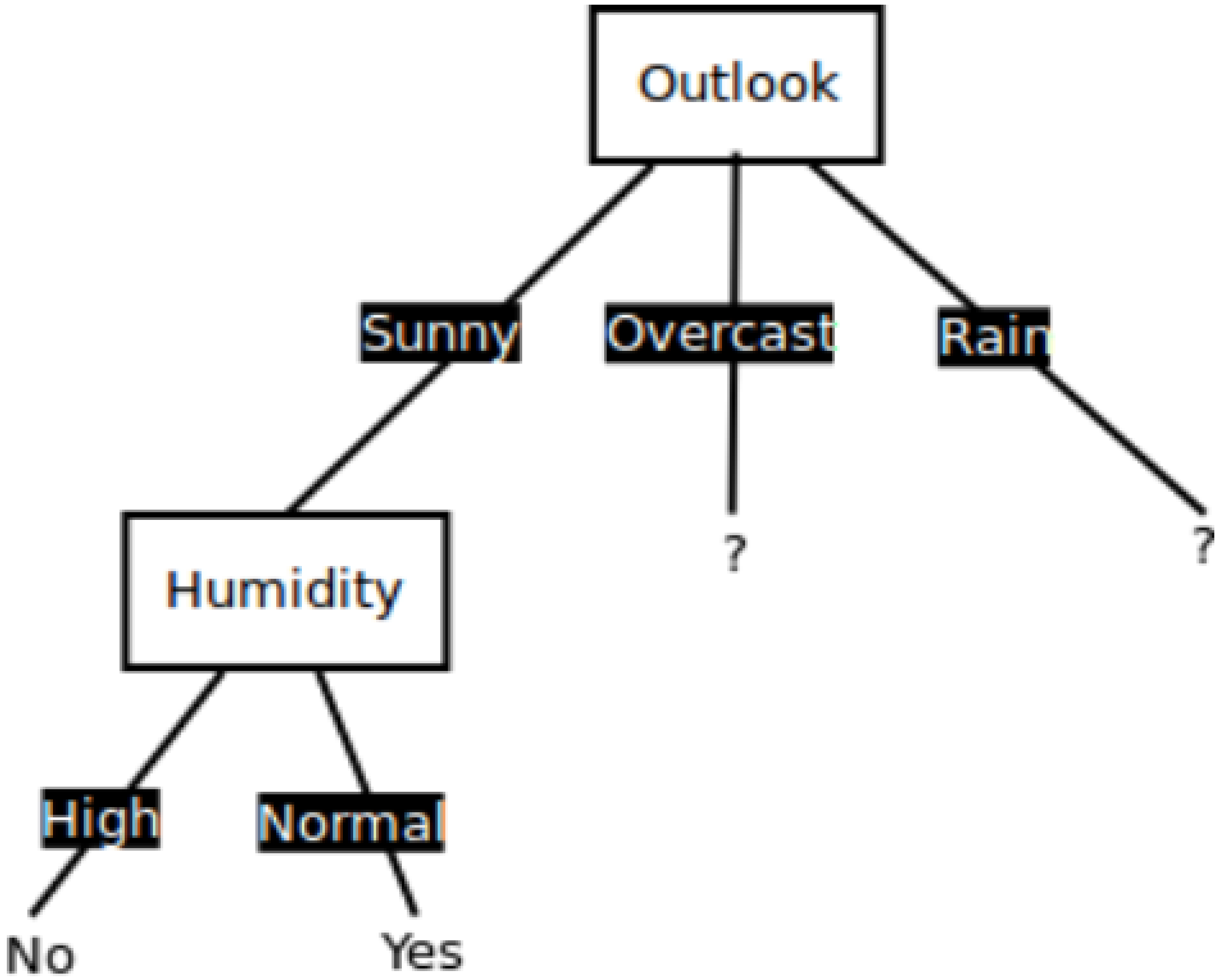
- So, based on the information gains calculated above,
- we choose attribute Humidity as attribute in branch Sunny.
- Our decision tree will look like an image below.




Add new attribute Humidity

# Third iteration

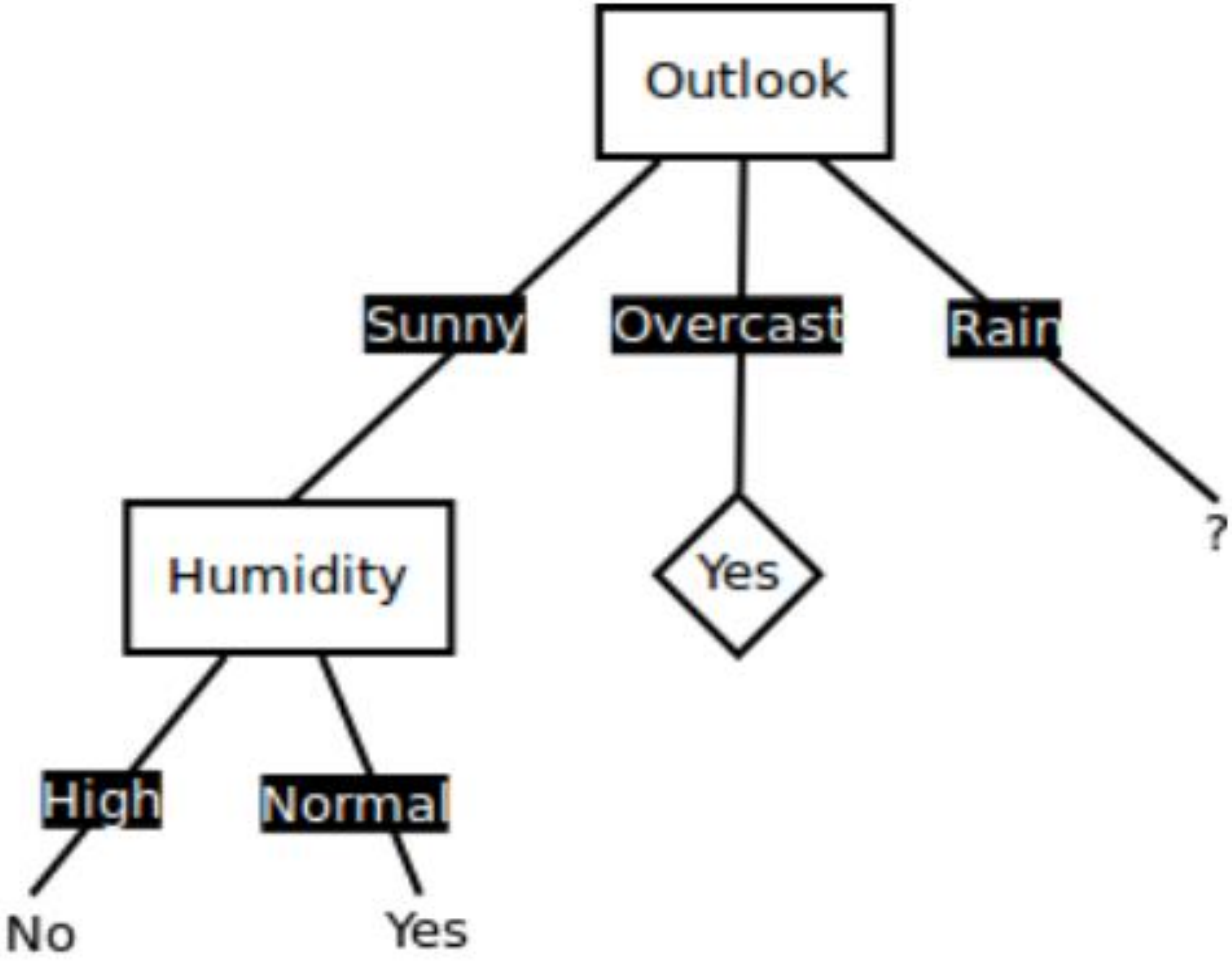
- Now, we remove Humidity from attribute list.
- Attributes left= [Wind, Temperature]
- Next node is an attribute Humidity which has two possible values {High, Normal}.
- A branch High dominated by single label which is No, caused this branch ended with a leaf contains label No. Same case with branch Normal which ended with a leaf contains label Yes.



Humidity's Leafs

# # Fourth iteration

All rows contain value Overcast are dominated by single label Yes, so branch of Overcast ended with a leaf contains label Yes.




Overcast's leaf

# # Fifth iteration

- We want examine which the best attribute for branch of Rain.

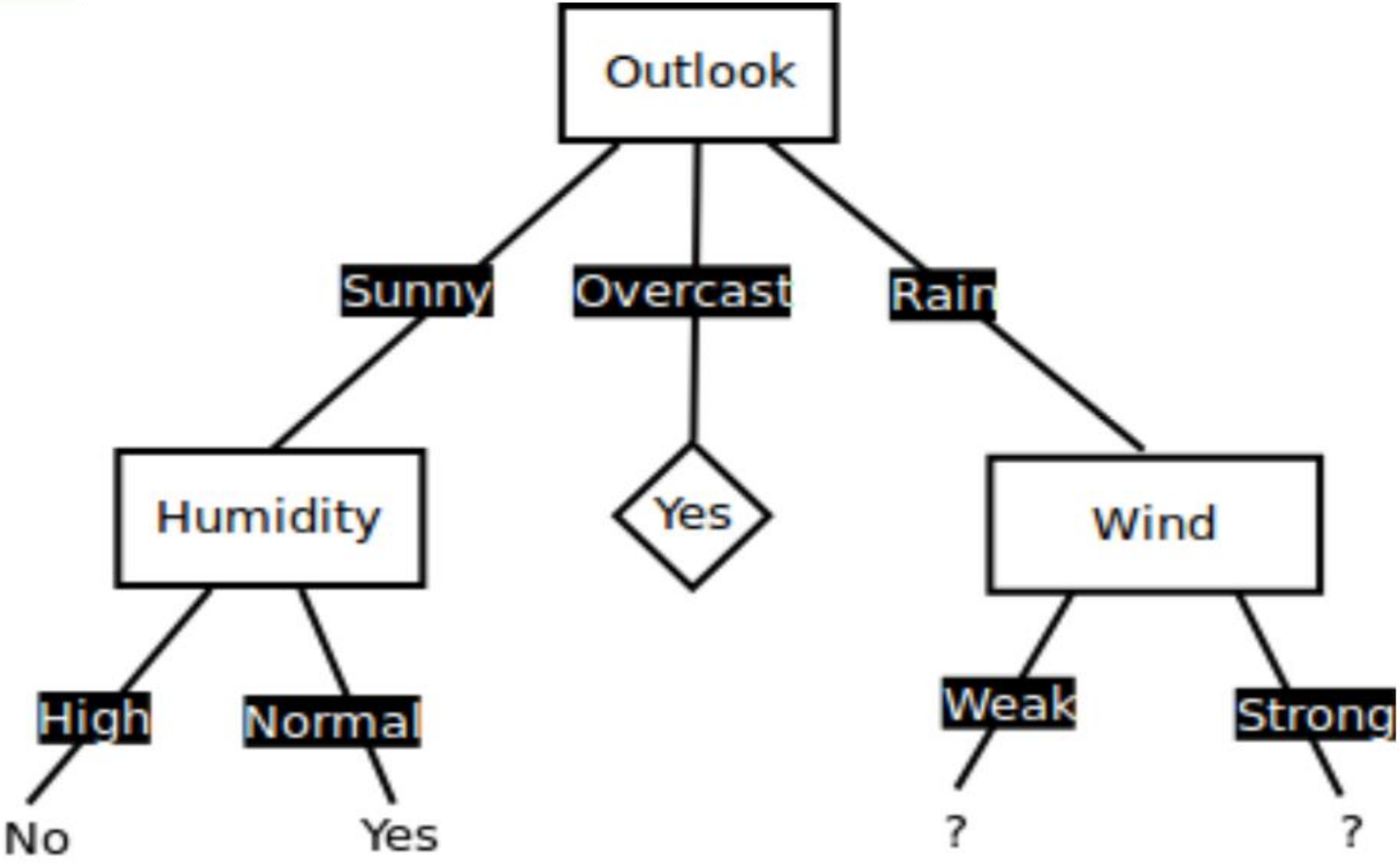
- Remember, that new S is rows containing values of Rain.

- Attributes left= [Wind , Temperature]

- $G(S_{rain}, Wind) = .970 - (2/5) * 0 - (3/5) * 0$
  $= .970$
- $G(S_{rain}, Temperature) = .970 - (0/5) * 0 - (3/5) * .918 - (2/5) * 1.0$
  $= .019$

# # Fifth iteration

- So, based on the information gains calculated above,
- we choose attribute Wind as attribute in branch of Rain.
- Our decision tree will look like an image below.

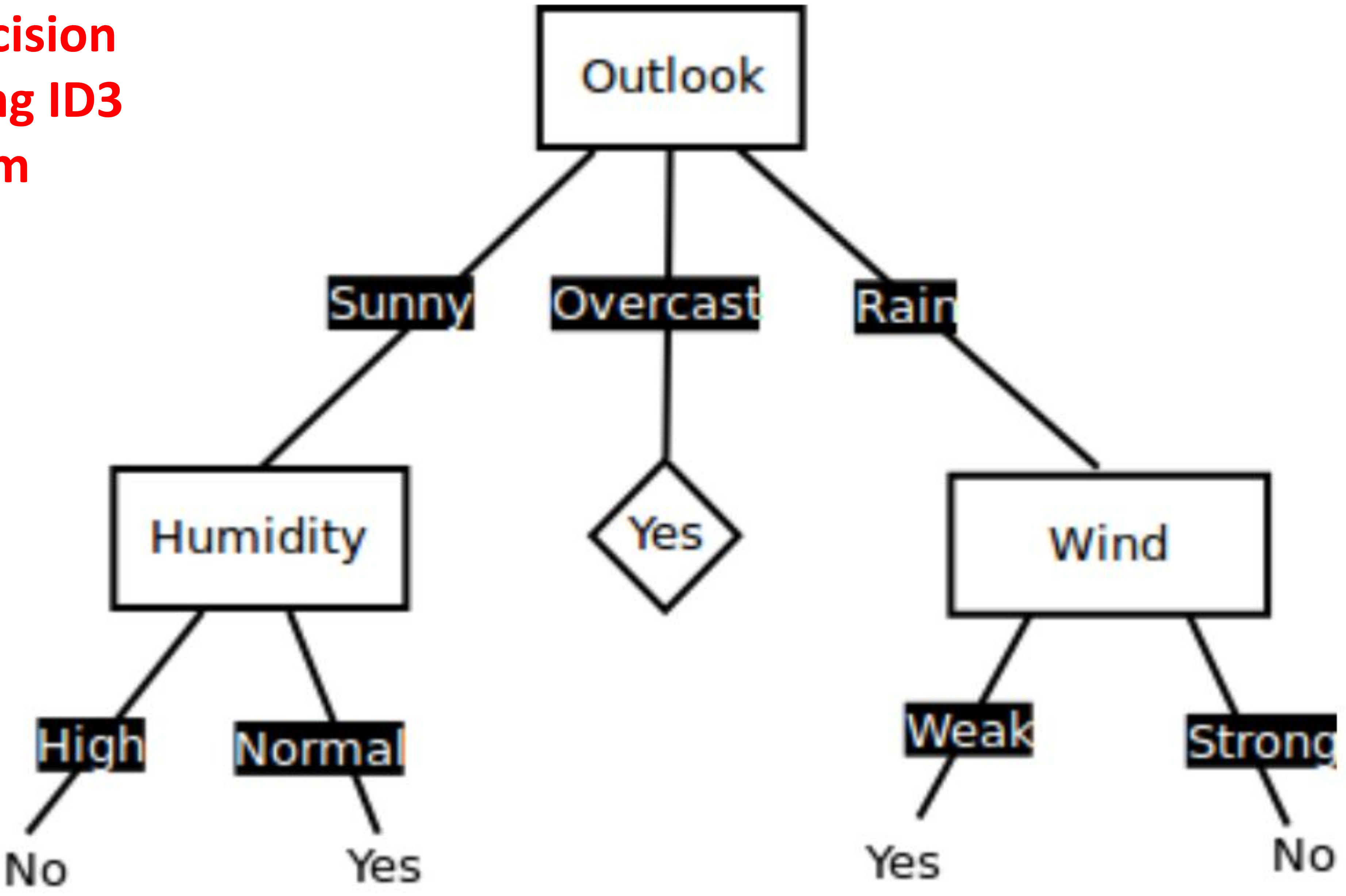



Add new attribute Wind

# # Sixth iteration

- For next iteration, we remove attribute Wind from attribute list.
- Next node is an attribute Wind which has two possible values {Weak, Strong}.
- A branch Strong dominated by single label which is No, caused this branch ended with a leaf contains label No.
- Same case with branch Weak which ended with a leaf contains label Yes.

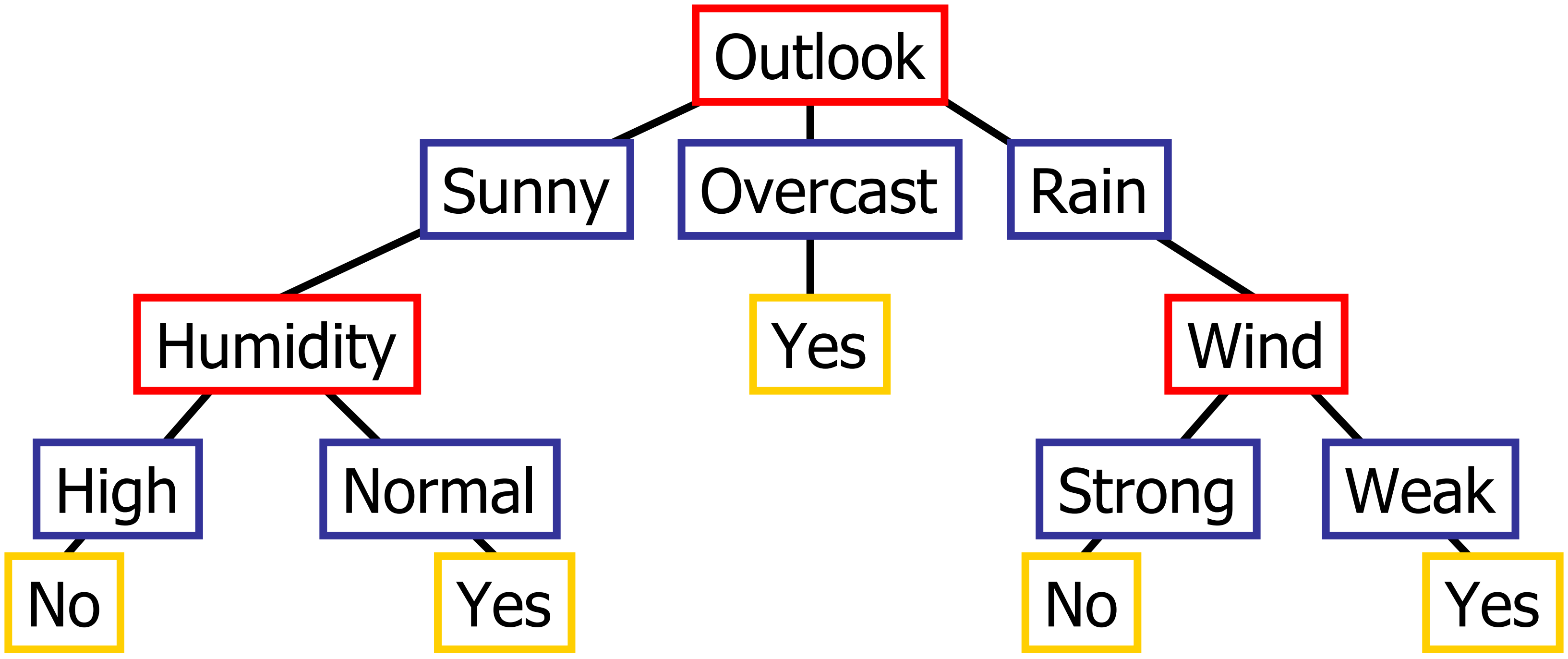
Final decision tree using ID3 algorithm



# Termination condition

- Since all branches in our decision tree ended with leaves.
- We stop here.
- We prune attribute Temperature from our decision tree.

# Converting a Tree to Rules



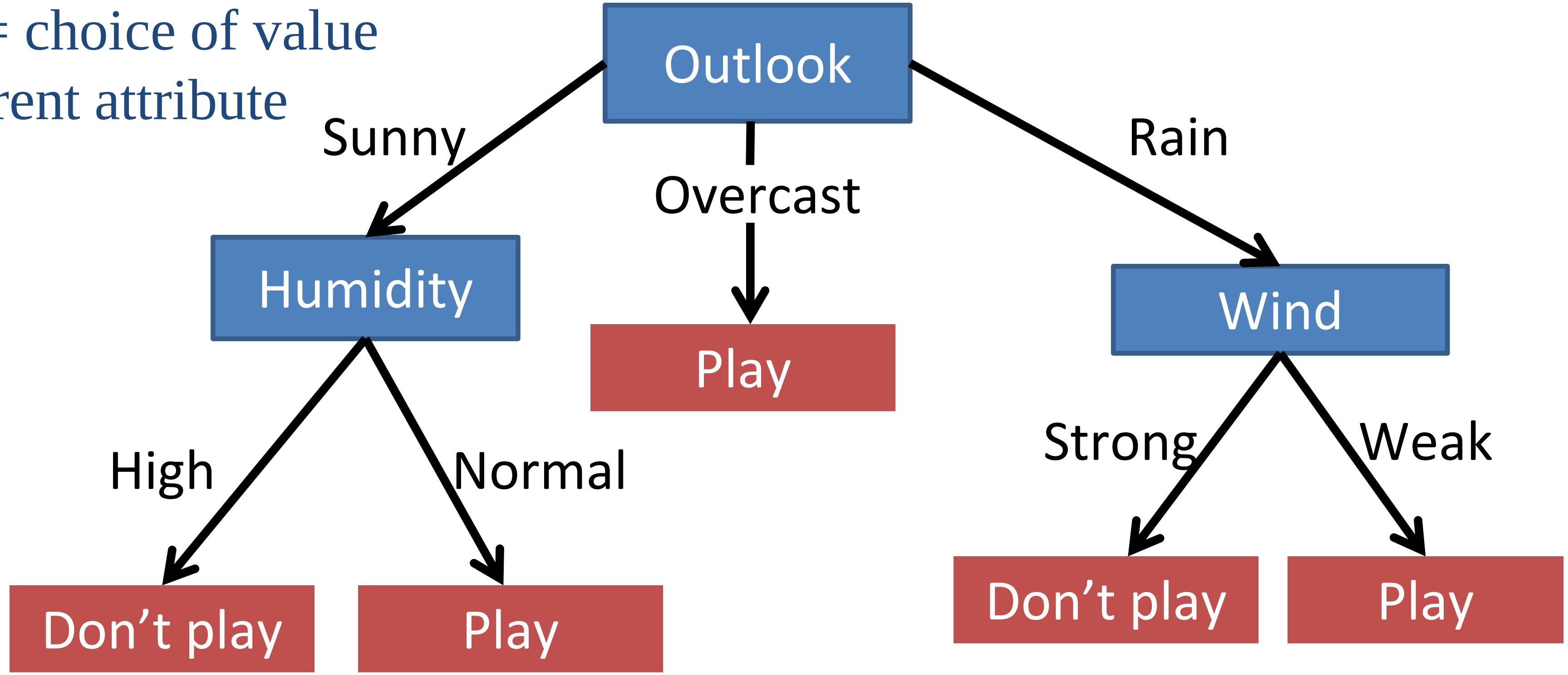

$R_1$: If (Outlook=Sunny) $\wedge$ (Humidity=High) Then PlayTennis=No
$R_2$: If (Outlook=Sunny) $\wedge$ (Humidity=Normal) Then PlayTennis=Yes
$R_3$: If (Outlook=Overcast) Then PlayTennis=Yes
$R_4$: If (Outlook=Rain) $\wedge$ (Wind=Strong) Then PlayTennis=No
$R_5$: If (Outlook=Rain) $\wedge$ (Wind=Weak) Then PlayTennis=Yes

# Decision Tree Representation

Good day for tennis?

Leaves = classification

Arcs = choice of value for parent attribute




Decision tree is equivalent to logic in disjunctive normal form

Play $\Leftrightarrow$ (Sunny $\wedge$ Normal) $\vee$ Overcast $\vee$ (Rain $\wedge$ Weak)

Construct a decision tree of the given training set using CART (Classification and Regression Tree) Algorithm

# Gini Index/Gini Impurity

$$\text{Gini Index}(\text{Attribute} = \text{Value}) = 1 - \sum_{i=1}^{N} (P_i)^2$$

$$\text{Ginni Index}(\text{Attribute}) = \sum_{v = \text{Values}} P_v * GI(v)$$

# Compute the Gini index of Outlook

Outlook

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

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

$$\text{Gini}(\text{Outlook}=\text{Overcast}) = 1 - (4/4)^2 - (0/4)^2 = 0$$

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

Now Calculate weighted Sum of Gini Indexes

$$\text{Gini}(\text{Outlook}) = (5/14) * 0.48 + \left(\frac{4}{14}\right) * 0 + \left(\frac{5}{14}\right) * 0.48$$

$$= 0.171 + 0 + 0.171 = 0.342$$

# Compute the Gini index of Temperature

Temperature

| Temperature | Yes | No | Number of Inst^n |
|---|---|---|---|
| Hot | 2 | 2 | 4 |
| Cool | 3 | 1 | 4 |
| Mild | 4 | 2 | 6 |

Gini (Temp = Hot) = $1-(2/4)^2-(2/4)^2 = 0.5$

Gini (Temp = Cool) = $1-(3/4)^2-(1/4)^2 = 0.375$

Gini (Temp = mild) = $1-(4/6)^2-(2/6)^2 = 0.445$

Calculate Weighted sum of Gini Indexes

Gini (Temperature) = $(4/14) * 0.5 + (4/14) * 0.375 + (6/14) * 0.445$

$= 0.142 + 0.107 + 0.190 = 0.439$

# Compute the Gini index of Wind and Humidity

Wind

| Wind | Yes | No | Number of Instances |
|---|---|---|---|
| weak | 6 | 2 | 8 |
| Strong | 3 | 3 | 6 |

$$Gini(wind = Weak) = 1-(6/8)^2-(2/8)^2 = 0.375$$

$$Gini(wind = Strong) = 1-(3/6)^2-(3/6)^2 = 0.5$$

$$Gini(wind) = \left(\frac{8}{14}\right) * 0.375 + \left(\frac{6}{14}\right) * 0.5 = 0.428$$

Humidity

| Humidity | Yes | No | Number of Instances |
|---|---|---|---|
| High | 3 | 4 | 7 |
| Normal | 6 | 1 | 7 |

$$Gini(Humidity = High) = 1-(3/7)^2-(4/7)^2 = 0.489$$

$$Gini(Humidity = Normal) = 1-(6/7)^2-(1/7)^2 = 0.244$$

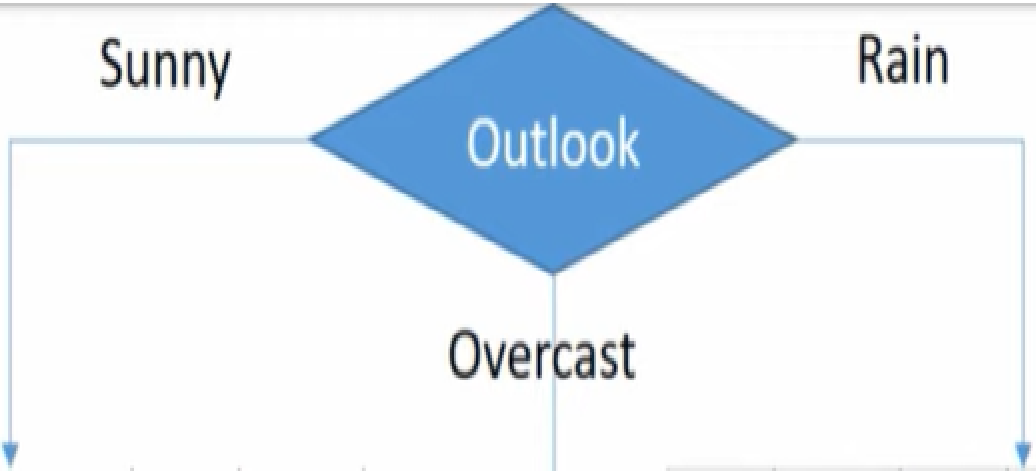
$$Gini(Humidity) = \left(\frac{7}{14}\right) * 0.489 + \left(\frac{7}{14}\right) * 0.244$$

$$= 0.367$$

- Attribute with lowest value of Gini Index is considered as Root node
- Here, Outlook has lowest Gini Index.

Time to decide

| Feature | Gini Index |
|---|---|
| Outlook | 0.342 |
| Temperature | 0.439 |
| Humidity | 0.367 |
| Wind | 0.428 |




| 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 |
|---|---|---|---|---|---|
| 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 |
|---|---|---|---|---|---|
| 3 | Overcast | Hot | High | Weak | Yes |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |

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 |

## 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 and Temp.=Hot) = $1 - (0/2)^2 - (2/2)^2 = 0$

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

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

Gini(Outlook=Sunny and Temp.) = $(2/5)*0 + (1/5)*0 + (2/5)*0.5 = 0.2$

**Gini of humidity for sunny outlook**

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

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

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

Gini(Outlook=Sunny and Humidity) = $(3/5)*0 + (2/5)*0 = 0$

**Gini of wind for sunny outlook**

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

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

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

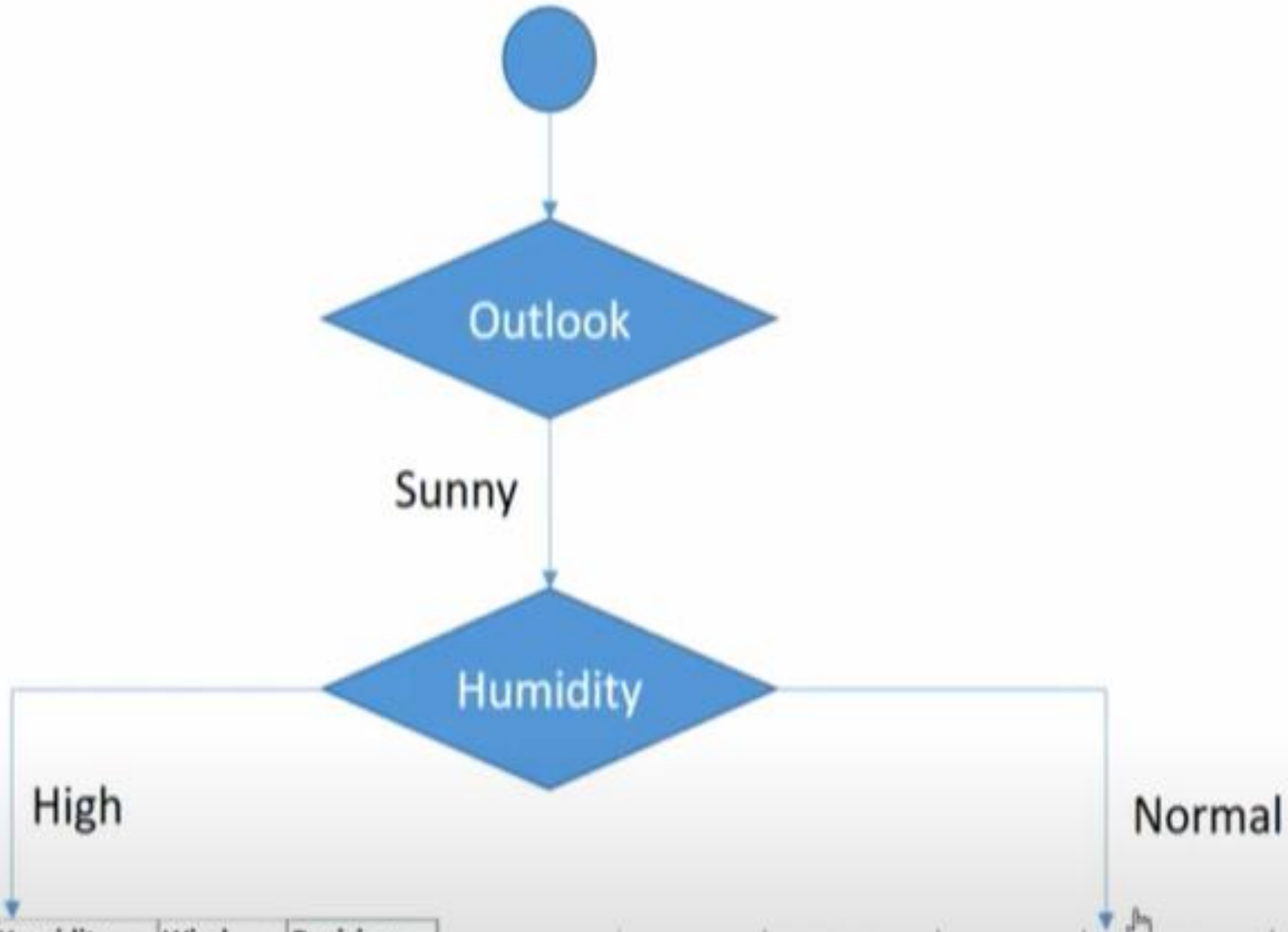
Gini(Outlook=Sunny and Wind) = $(3/5)*0.266 + (2/5)*0.2 = 0.466$

**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 |

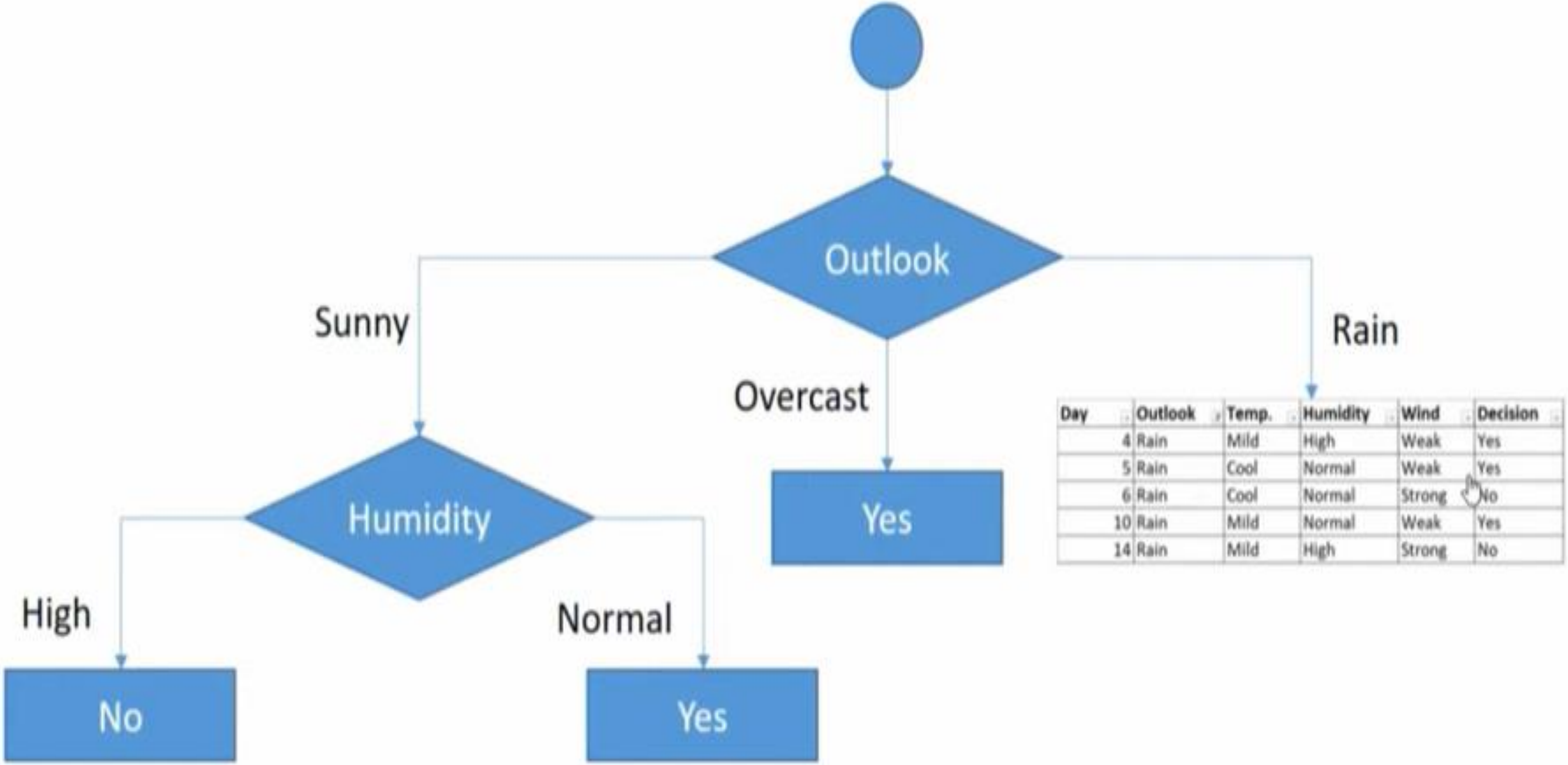
We'll put humidity check at the extension of sunny outlook.




| 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 |

Sub datasets for high and normal humidity




| 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 |

Decisions for high and normal humidity

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 |

We'll calculate Gini index scores for temperature, humidity and wind features when outlook is rain.

## Gini of temperature for rain outlook

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

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

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

Gini(Outlook=Rain and Temp.) = $(2/5)*0.5 + (3/5)*0.444 = 0.466$

**Gini of humidity for rain outlook**

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

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

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

Gini(Outlook=Rain and Humidity) = $(2/5)*0.5 + (3/5)*0.444 = 0.466$

## Gini of wind for rain outlook

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

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

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

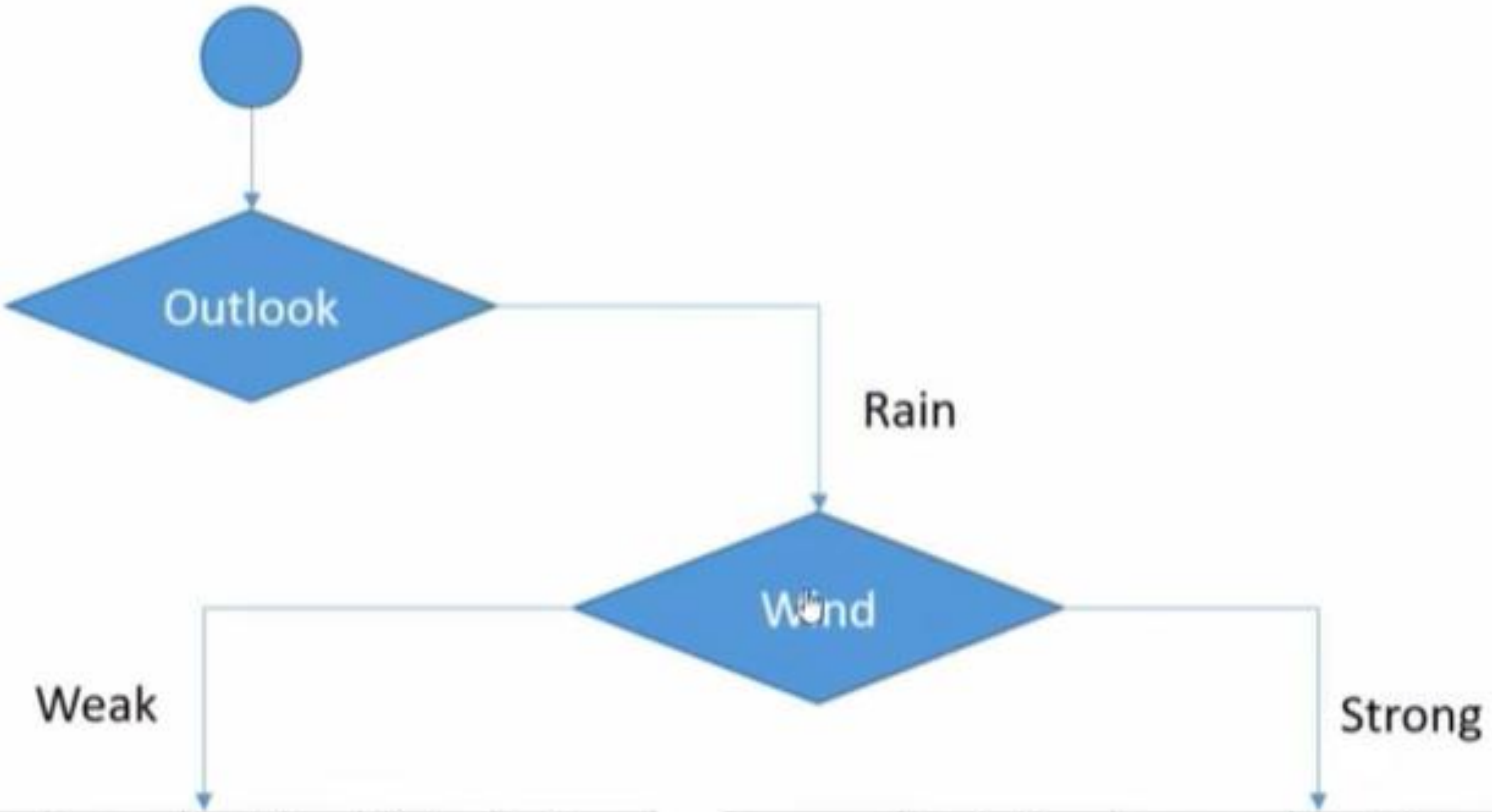
Gini(Outlook=Rain and Wind) = $(3/5)*0 + (2/5)*0 = 0$

**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 |

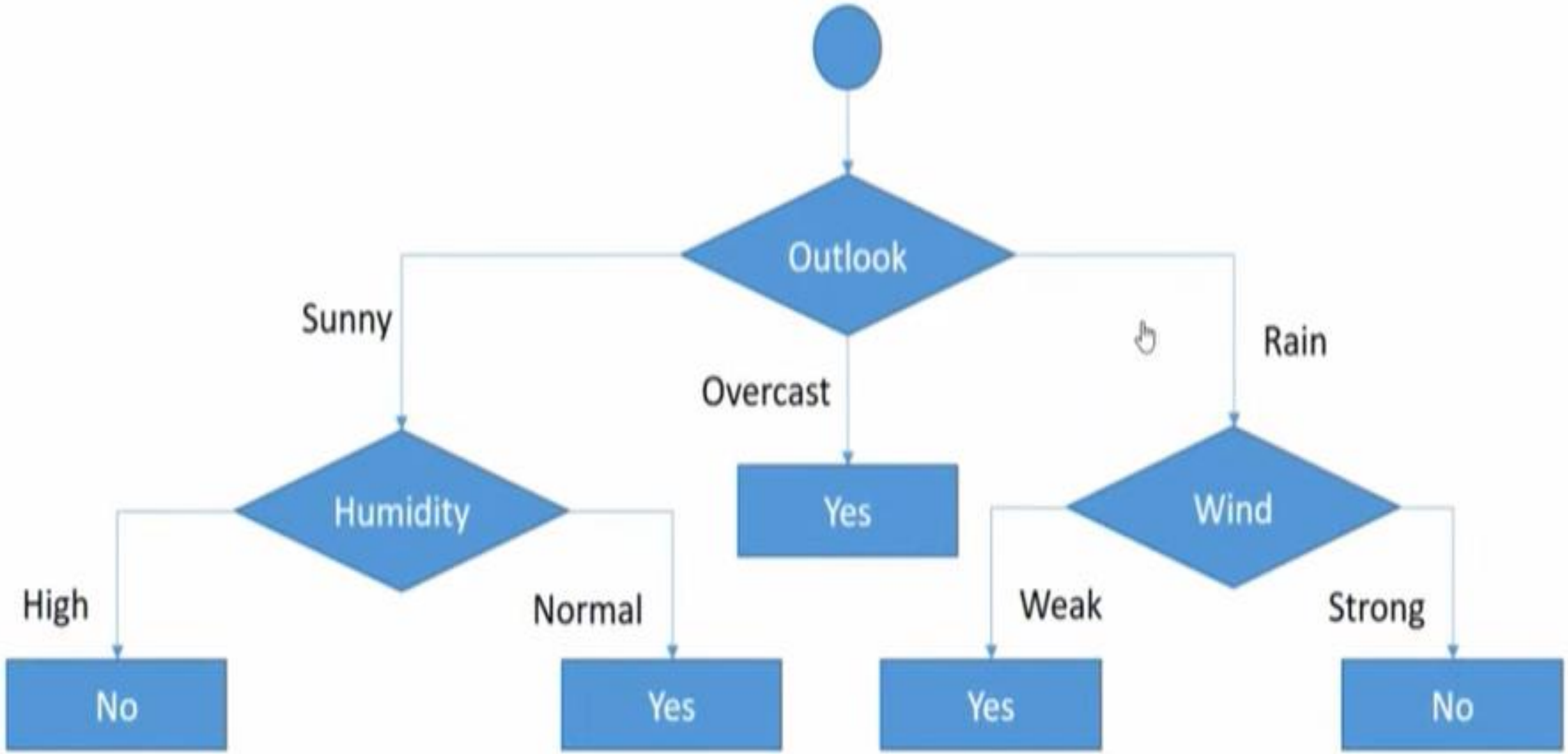
Put the wind feature for rain outlook branch and monitor the new sub data sets.




| 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 |

Sub data sets for weak and strong wind and rain outlook




Final form of the decision tree built by CART algorithm.