ID3 Algorithm & Entropy

- There are many algorithms to construct Decision Trees, but one of the best is called as **ID3 Algorithm**.
- ID3 Stands for **Iterative Dichotomiser** 3 and it was developed by Quinlan, the selection of partitioning was made on the basis of the information gain/entropy reduction.
- **Shannon Entropy:** It measure the amount of uncertainty or randomness in data and is denoted by H(S) for a finite set S.

$$H(S) = \sum_{x \in X} p(x) \log_2 \left(\frac{1}{P(x)} \right)$$

- Intuitively, it tells us about the **predictability of a certain event**.
- Consider a bag that is filled with red balls, the entropy of such an event (drawing a red ball) can be predicted perfectly since the bag has all red balls, **i.e. event has no randomness and hence it's entropy is zero**.
- **Lower values of entropy imply less uncertainty** while higher values imply high uncertainty.

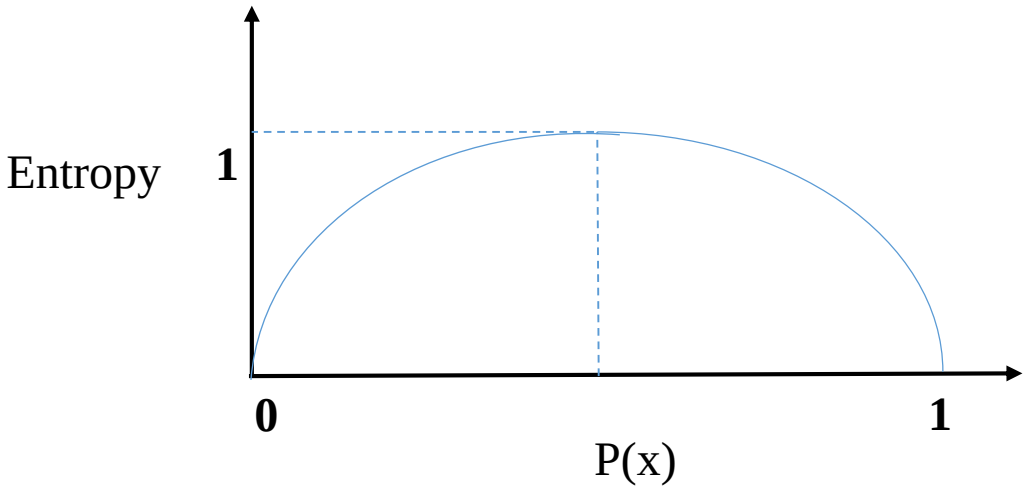
Shannon Entropy

- In general, Shannon entropy indicates the “amount of information in a variable” – intuitively it gives amount of amount of storage (i.e., number of bits) required to store the variable.
- The easiest way to measure the information is in bits or bytes. The basic unit of information is bit and it represents two possible states.
- If the **base** of the **logarithm** is e, the **entropy** is **measured** in nats.

- Think what is more intuitive to use to compute entropy: log base 2 or natural logarithm. 

ID3 Algorithm

- For predicting the class, **algorithm starts from the root node**.
- To get the root node, it **compares the values of root attribute** with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.
- At next node, **the algorithm again compares the attribute value** with the other sub-nodes and move further.
- The process **continues until it reaches the leaf node of the tree**.



Information Gain

- **<u>Information Gain:</u>** It measures the **relative change in entropy** with respect to the independent variables and it is also called as **Kullback-Leibler divergence denoted by IG(S, A) for a set S**.
- It indicates the effective change in entropy after deciding on a particular attribute A.

$$IG(\mathrm{S}, A) = H(S) - H(S, A)$$

$$IG(\mathrm{S}, A) = H(S) - \sum_{i=1}^{n} p(x) * H(x)$$

- The IG(S, A) is the information gain by applying feature A; H(S) is the Entropy of the entire set, and $H(S, A)$ calculates the Entropy after applying the feature A, where P(x) is the probability of event x.
- Choose the **DT that gives highest value of information gain.**

Measures of Impurity for Evaluating Splits in Decision Trees.

- An impurity is a heuristic for selecting the splitting criterion that "best" separates a given dataset *S* of class labeled training tuples into individual classes.

- If *S* were split into smaller partitions according to the outcome of the splitting criterion, ideally each partition would be pure.
- More the impurity (more the heterogeneity in the dataset), more the entropy, more the expected amount of information that would be needed to classify a new amount of information that would be needed to classify a new pattern.

Pruning the Tree

- Pruning is a process of deleting the unnecessary nodes from a tree to get the optimal decision tree.
- A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset.
- Two main groups:
  - Prepruning: Growing of the tree is stopped before it reaches the point where it perfectly classifies the training data.
  - Postpruning: The tree is allowed to grow to perfectly classify the training examples, and then post-prune is done.
- The second approach of post-pruning overfit trees has been found to be more successful in practice because it is not easy to precisely estimate when to stop growing the tree.
- **Pruning Techniques:** Cost Complexity Pruning & Reduced Error Pruning.

The ID3 Decision Tree

The ID3 Decision Tree

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Outlook	Temperature	Humidity	Windy	Play Tennis
Sunny	**Hot**	**Normal**	**True**	**?**

Day	Outlook	Temperature	Humidity	Wind	Play Golf
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	**Overcast**	**Hot**	**High**	**Weak**	**Yes**
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	**Overcast**	**Cool**	**Normal**	**Strong**	**Yes**
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	**Overcast**	**Mild**	**High**	**Strong**	**Yes**
D13	**Overcast**	**Hot**	**Normal**	**Weak**	**Yes**
D14	Rain	Mild	High	Strong	No

Example:

- Calculate H(S), the Entropy of the current state.
- In total there are 5 No's and 9 Yes's for total 14 outcomes.

Entropy(S) = $\sum_{x \in X} p(x) \log_2 \left(\frac{1}{P(x)} \right)$

Gini Index = 1- [(p+)^2 + (p-)^2]
= 1- [(9/14)2 + (5/14)2]
= 0.4592

Entropy(S) = $-(\frac{9}{14}) \log_2 (\frac{9}{14}) - (\frac{5}{14}) \log_2 \left(\frac{5}{14}\right) = 0.940$

- Entropy is 0 means that all members belong to the same class, and if it 1 it indicates that half of them belong to class '0', and other half belong to class '1', which is a perfect random case.
- Here it's 0.94 means the distribution is fairly random.

- Choose the attribute that gives us highest possible Information Gain

$$IG(S, A) = H(S) - \sum_{i=1}^{n} p(x) * H(x)$$

Example:

- Lets start with “Wind”.
- In total there we have 8 places where wind is weak and 6 places where wind is strong for total 14 outcomes.

P(Sweak) = Number of weak/ Total = 8/14
P(Sstrong) = Number of strong/total = 6/14

$$\text{Entropy(Sweak)} = -(\tfrac{6}{8})\log_2(\tfrac{6}{8}) - (\tfrac{2}{8})\log_2\left(\tfrac{2}{8}\right) = 0.811$$

$$\text{Entropy(Sstrong)} = -(\tfrac{3}{6})\log_2(\tfrac{3}{6}) - (\tfrac{3}{6})\log_2\left(\tfrac{3}{6}\right) = 1$$

IG(S, Wind) = H(S) – P(Sweak)*H(Sweak) – P(Sstrong)*H(Sstrong)

$$= 0.940 - (\tfrac{8}{14})(0.811) - (\tfrac{6}{14})(1) = 0.048$$

- In similar way, we will calculate information gain for all other features

IG(S, Outlook) = 0.246
IG(S, Temperature) = 0.029
IG(S, Humidity) = 0.151
IG(S, Wind) = 0.048

- IG(S, Outlook) has the highest information gain of 0.246, **hence Outlook attribute is chosen as the root node**.

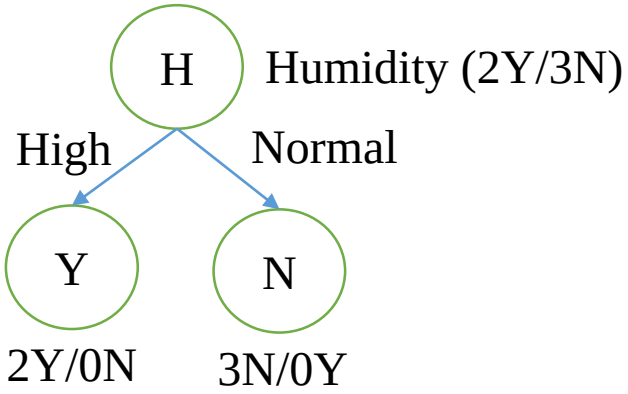
Example:

- There are three possible values of Outlook: Sunny, Overcast, and Rain.
- Overcast node already ended up having leaf node ‘Yes’, we have two subtrees to compute: Sunny and Rain.

$$H(Ssunny) = -(\frac{3}{5}) \log_2 (\frac{3}{5}) - (\frac{2}{5}) \log_2 \left(\frac{2}{5}\right) = 0.97$$

IG(Ssunny, Humidity) = 0.96
IG(Ssunny, Temperature) =0.57
IG(Ssunny, Wind) =0.019

- IG(Ssunny, Humidity) has the highest information gain of 0.96, **hence Humidity attribute is chosen.**
- Repeat the process



Detailed Calculations

Categorical values - high, normal

H(Sunny, Humidity=high) = - 0 - (3/3)*log(3/3) = 0
H(Sunny, Humidity=normal) = -(2/2)*log(2/2)-0 = 0
Average Entropy Information for Humidity –

I(Sunny, Humidity) = p(Sunny, high)*H(Sunny, Humidity=high) + p(Sunny, normal)*H(Sunny, Humidity=normal)
= (3/5)*0 + (2/5)*0 = 0 Information Gain = H(Sunny) - I(Sunny, Humidity) = 0.971 - 0 = 0.971

Categorical values - hot, mild, cool

H(Sunny, Temperature=hot) = -0-(2/2)*log(2/2) = 0
H(Sunny, Temperature=cool) = -(1)*log(1)- 0 = 0
H(Sunny, Temperature=mild) = -(1/2)*log(1/2)-(1/2)*log(1/2) = 1
Average Entropy Information for Temperature –

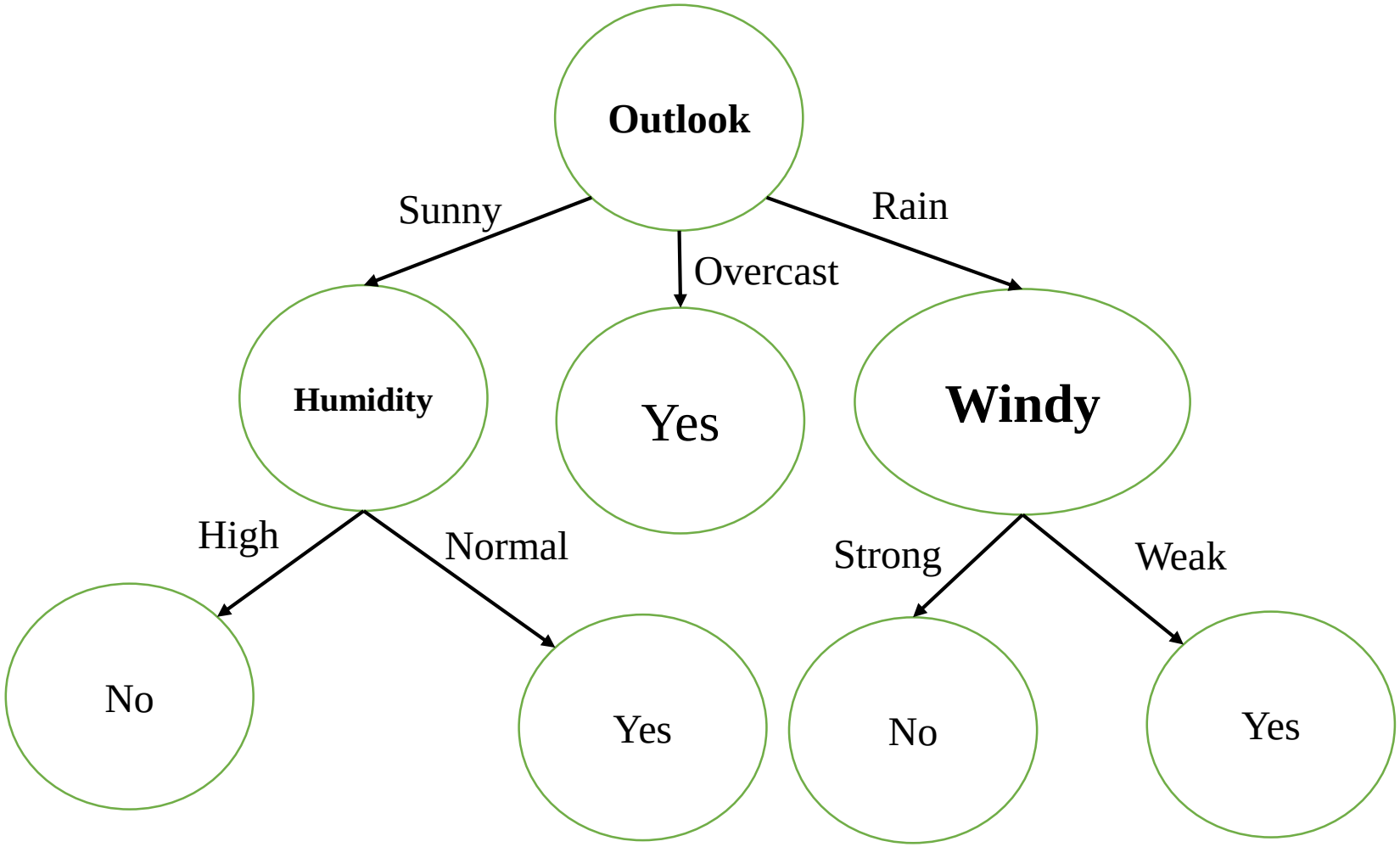
I(Sunny, Temperature) = p(Sunny, hot)*H(Sunny, Temperature=hot) + p(Sunny, mild)*H(Sunny, Temperature=mild) + p(Sunny, cool)*H(Sunny, Temperature=cool)
= (2/5)*0 + (1/5)*0 + (2/5)*1 = 0.4 Information Gain = H(Sunny) - I(Sunny, Temperature) = 0.971 - 0.4 = 0.571

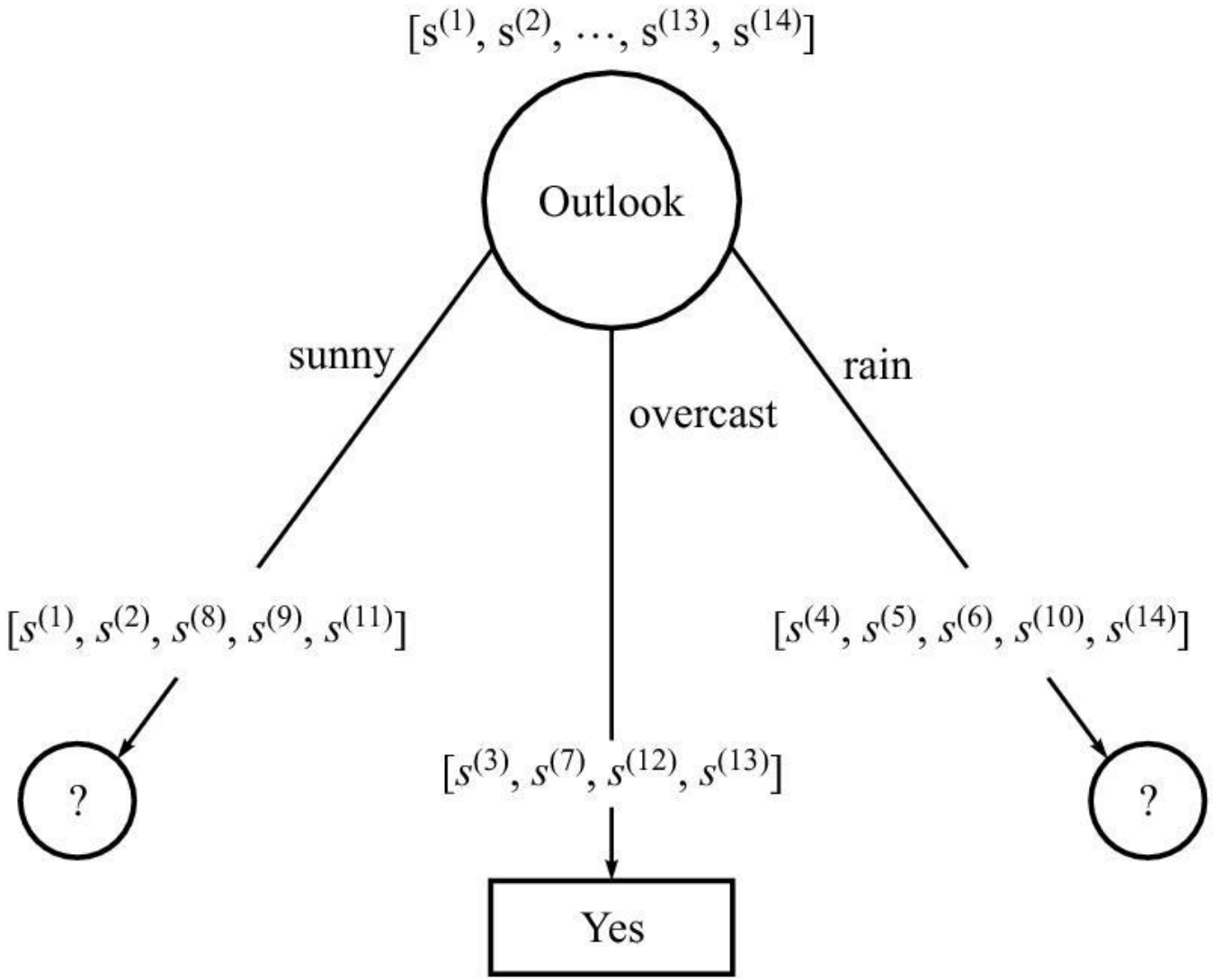
Categorical values - weak, strong

H(Sunny, Wind=weak) = -(1/3)*log(1/3)-(2/3)*log(2/3) = 0.918
H(Sunny, Wind=strong) = -(1/2)*log(1/2)-(1/2)*log(1/2) = 1
Average Entropy Information for Wind –

I(Sunny, Wind) = p(Sunny, weak)*H(Sunny, Wind=weak) + p(Sunny, strong)*H(Sunny, Wind=strong)
= (3/5)*0.918 + (2/5)*1 = 0.9508 Information Gain = H(Sunny) - I(Sunny, Wind) = 0.971 - 0.9508 = 0.0202

Example:


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Windy
High
Normal
Strong
Weak
No
Yes
No
Yes



Partially learned decision tree: the training examples are sorted to corresponding descendant nodes

The CART Decision Tree

Gini gain (S, outlook)= 0.4592 – 0.342 = 0.117
Gini gain (S, temperature) = 0.4592- 0.4405 = 0.0185
Gini gain (S, Humidity) = 0.0916
Gini gain (S, wind) = 0.0304

The CART Decision Tree

Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Outlook	Temperature	Humidity	Windy	Play Tennis
Sunny	**Hot**	**Normal**	**True**	**?**

		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

Attribute	Rule	Error	Total Error
Outlook	Sunny → **No**	2/5	4/14
	Overcast → **Yes**	0/4	
	Rainy → **Yes**	2/5	

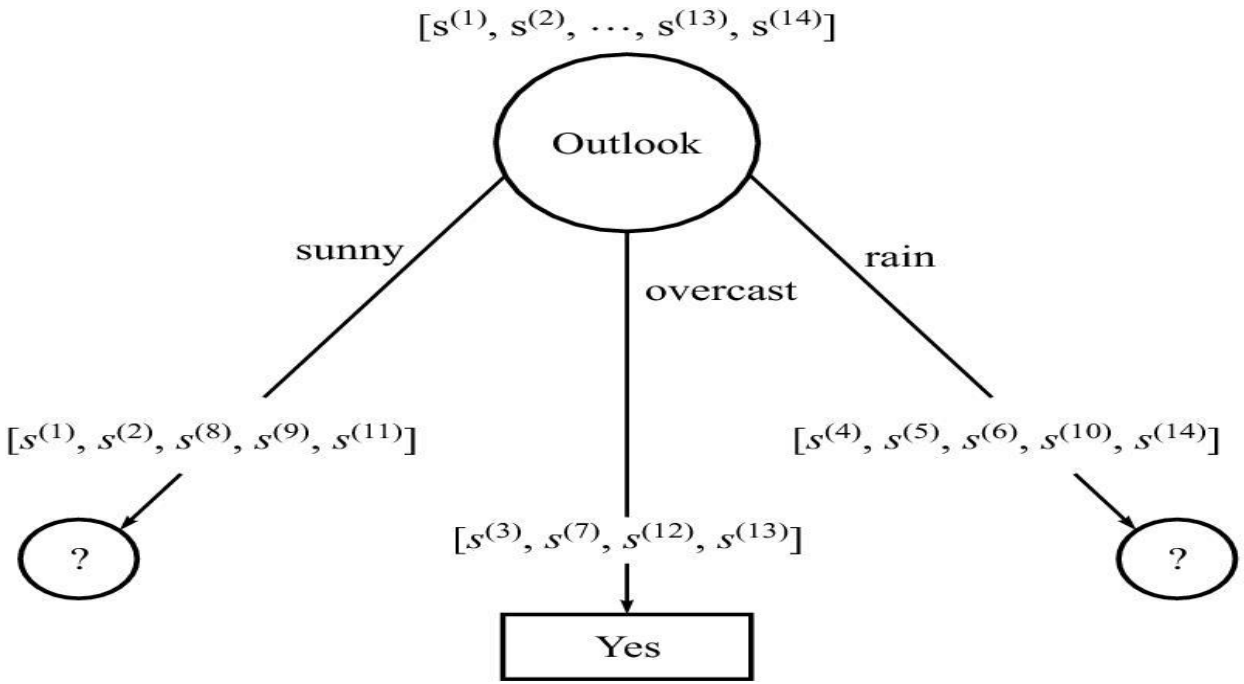
Attribute	Rule	Error	Total Error
Temp	Hot → **No / Yes**	2/4	5/14
	Mild → **Yes**	2/6	
	Cool → **Yes**	1/4	

Attribute	Rule	Error	Total Error
Humidity	High → **No**	3/7	4/14
	Normal → **Yes**	1/7	

Attribute	Rule	Error	Total Error
	True → **No / Yes**	3/6	5/14
Windy	False → **No**	2/8	

- Choose the one which has minimum error, in case there is a tie then choose the one that gives zero error for one of the category.

Therefore, outlook is selected as root node



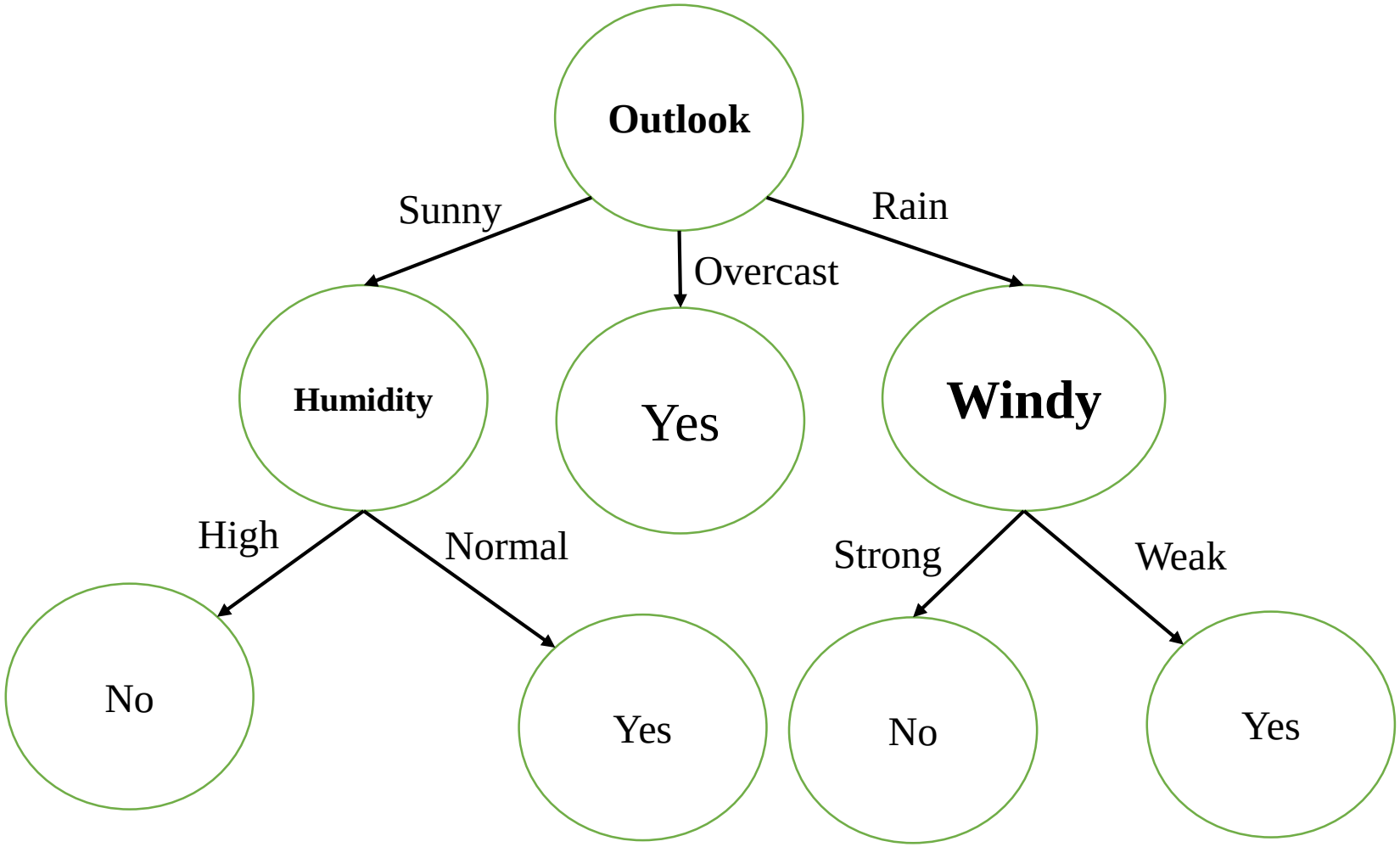
Attribute	Rule	Error	Total Error
Temp	Hot → **No**	0/2	1/5
	Mild → **No / Yes**	1/2	
	Cool → **Yes**	0/1	

Attribute	Rule	Error	Total Error
Humidity	High → **No**	0/3	0/5
	Normal → **Yes**	0/2	

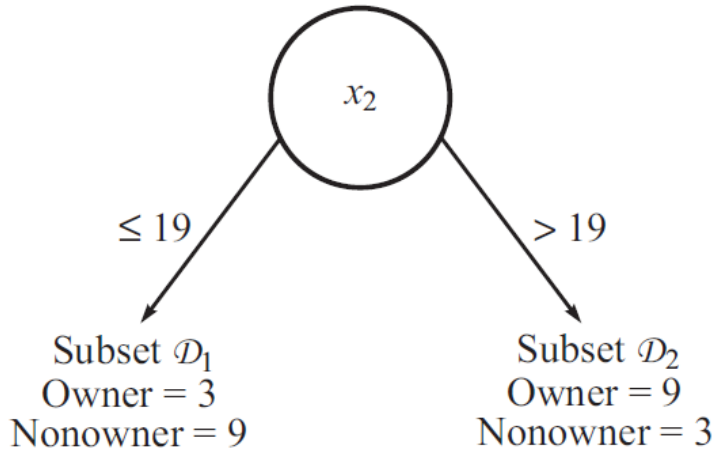
Attribute	Rule	Error	Total Error
Windy	True → **No / Yes**	1/3	2/5
	False → **No**	1/2	

Select Humidity

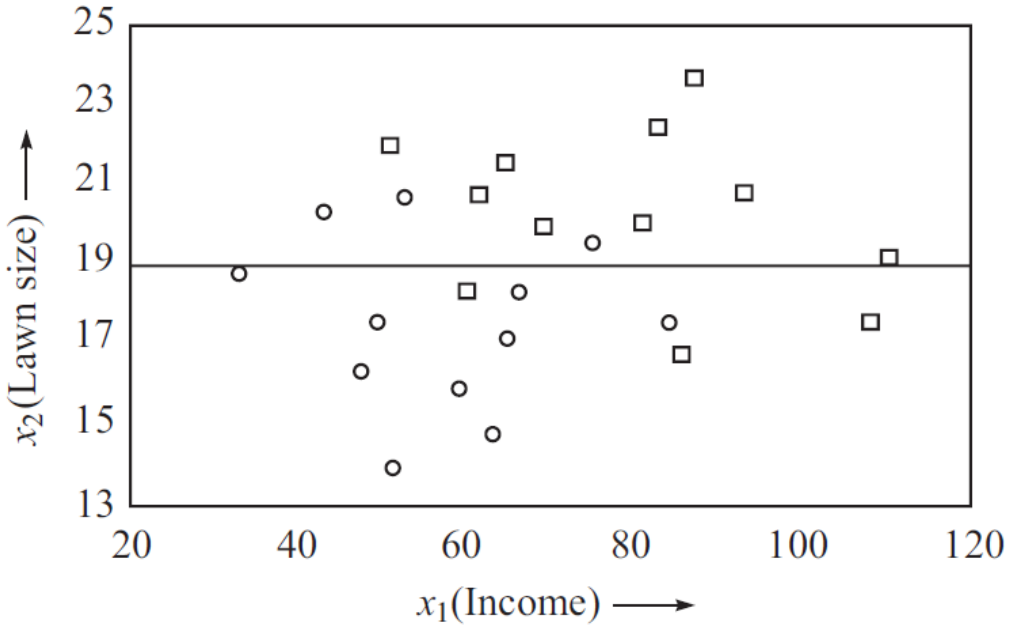
Example:


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Windy
High
Normal
Strong
Weak
No
Yes
No
Yes

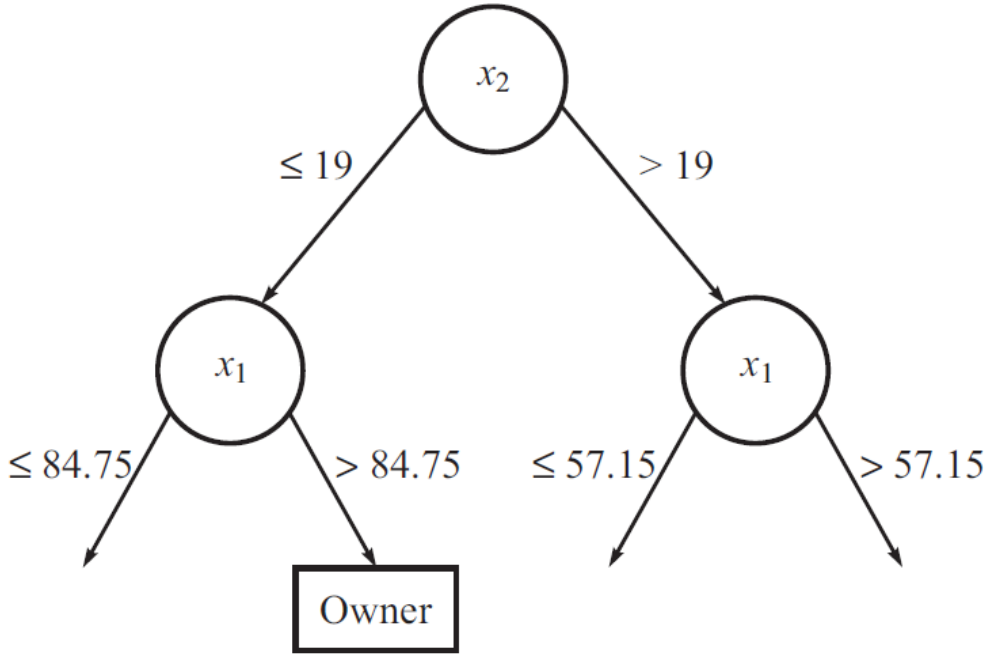
Household $s^{(i)}$	Income ($ thousands) x_1	Lawn Size (thousands ft^2) x_2	Ownership of a lawn tractor y
1	60	18.4	Owner
2	75	19.6	Nonowner
3	85.5	16.8	Owner
4	52.8	20.8	Nonowner
5	64.8	21.6	Owner
6	64.8	17.2	Nonowner
7	61.5	20.8	Owner
8	43.2	20.4	Nonowner
9	87	23.6	Owner
10	84	17.6	Nonowner
11	110.1	19.2	Owner
12	49.2	17.6	Nonowner
13	108	17.6	Owner
14	59.2	16	Nonowner
15	82.8	22.4	Owner
16	66	18.4	Nonowner
17	69	20	Owner
18	47.4	16.4	Nonowner
19	93	20.8	Owner
20	33	18.8	Nonowner
21	51	22	Owner
21	51	14	Nonowner
23	81	20	Owner
24	63	14.8	Nonowner
Random Sample of Households in a city with respect to ownership of a lawn tractor			



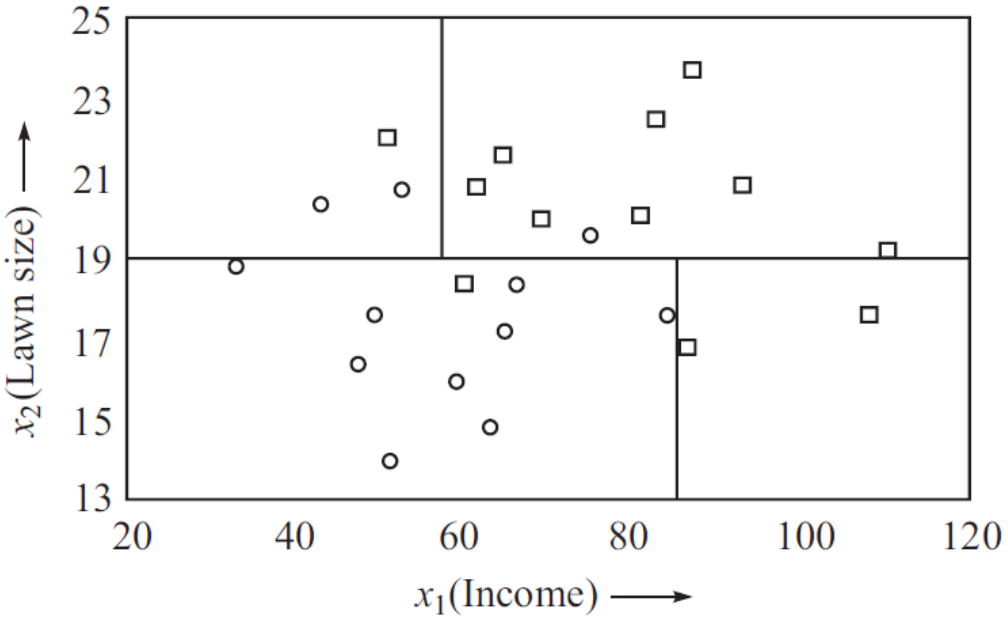
Tree stumps after first split



Scatter plot after first split



Tree stumps after first three splits



Scatter plot after first three splits

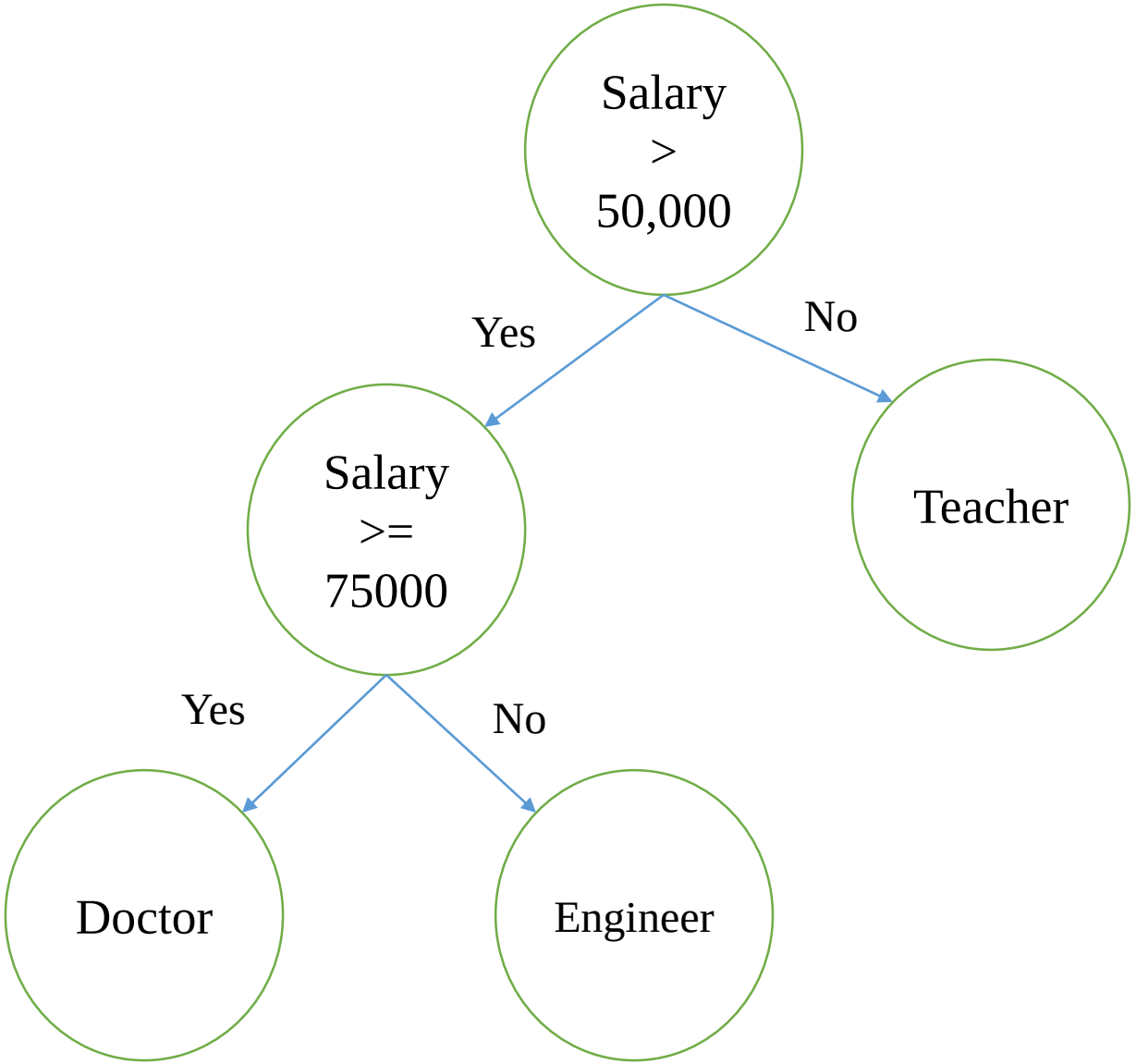
- If the partitioning is continued till all the branches hit leaf nodes, each rectangle will have data points from just one of the two classes

Impact of Outlier

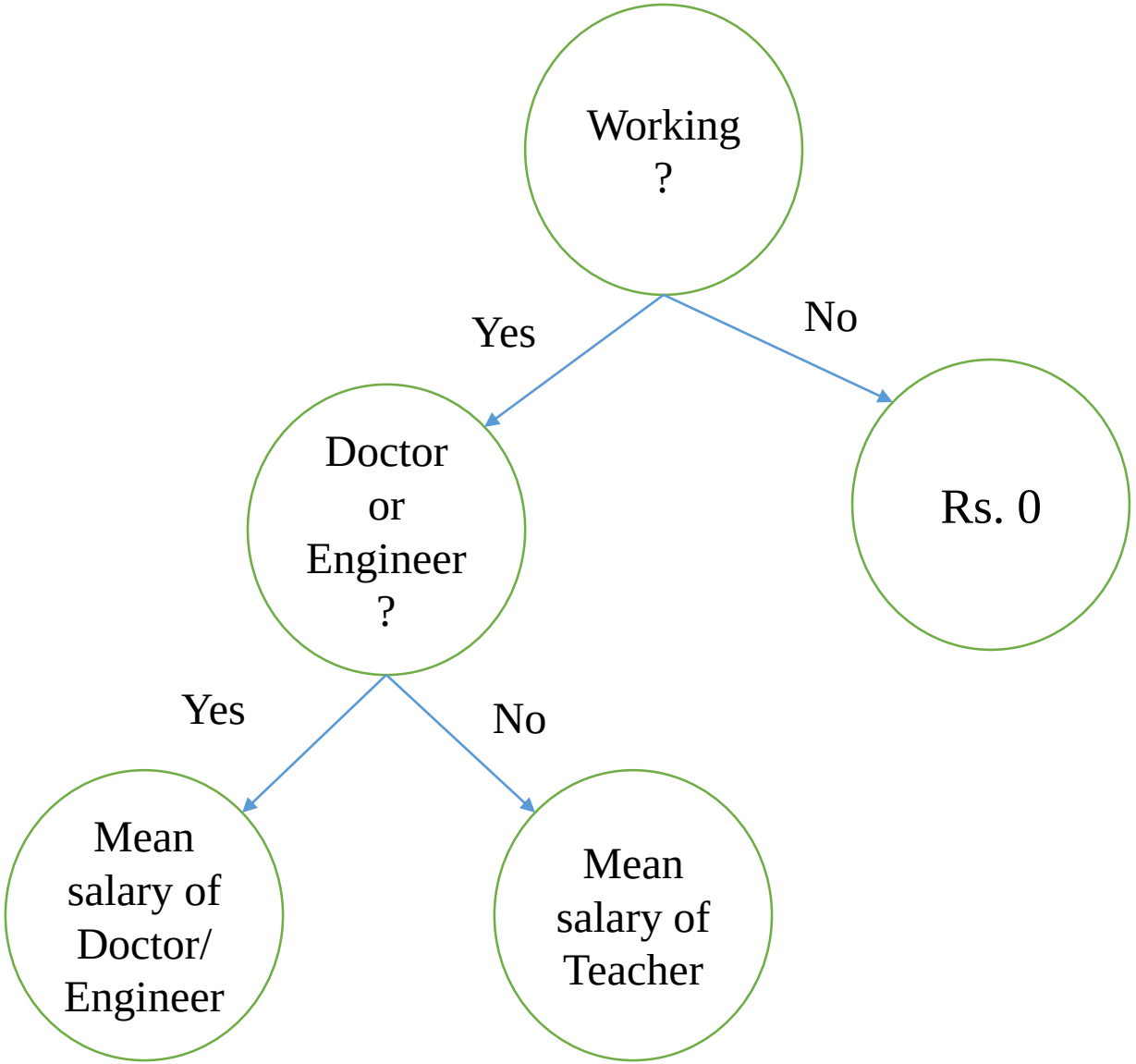
- Decision tree works on an "**if-then**" concept that makes the model to ask specific questions to the data. If the condition is satisfied then it gives a defined output.
- Outliers will have a considerable effect only if it is present in a continuous (numerical) column.
- Impact of outliers in predictor variables (continuous variable)
- Impact of outliers in target variables (continuous variable)
- For categorical variables, if there are few rare occurrences of a particular class then we can consider them as anomalies or class imbalance and treat them accordingly.

Example

Person Id	Salary	Occupation
1.	Rs. 80,000	Doctor
2.	Rs. 55,000	Engineer
3.	Rs. 18,000	Teacher
4.	Rs. 25,000	Teacher
5.	**Rs. 10**	**Teacher**
6.	**Rs. 1,20,000**	**Doctor**
7.	Rs. 65,000	Engineer
8.	Rs. 75,000	Doctor
9.	Rs. 72,000	Engineer



- **This decision tree can predict the occupation of person 5 and person 6 without any error despite them being a part of outliers. Therefore, we can say that** an outlier in predictor variables cannot affect the predictive ability of the model in most of the time.



- Salary of engineers and doctors is not correct; thus, if the outliers are present in target variables then there might be some impact (but not necessarily).