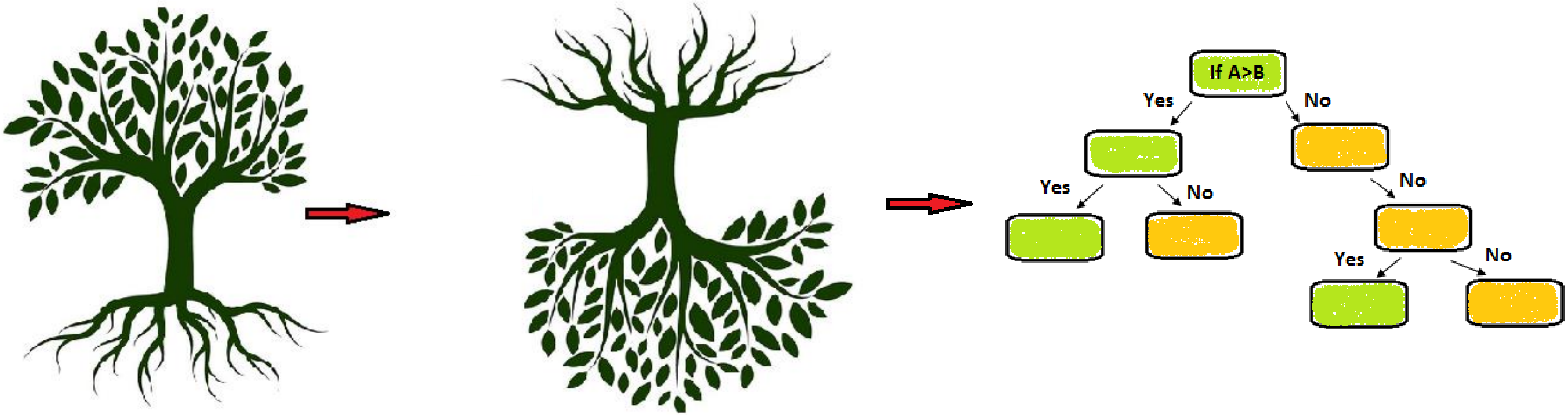
Decision Tree

Dr. Mrunal Rane

Introduction



Information Gain

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

Where:

- S is the set of samples at a particular node.
- c is the number of classes.
- p_i is the proportion of samples that belong to class i.

$$\text{Information Gain} = \text{Entropy(parent)} - \left(\frac{|S_{\text{left}}|}{|S|} \times \text{Entropy}(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} \times \text{Entropy}(S_{\text{right}})\right)$$

Where:

- S_{left} and S_{right} are the subsets of samples that go to the left and right child nodes respectively.
- $|S_{\text{left}}|$ and $|S_{\text{right}}|$ are the number of samples in the left and right subsets respectively.
- $|S|$ is the total number of samples at the parent node.

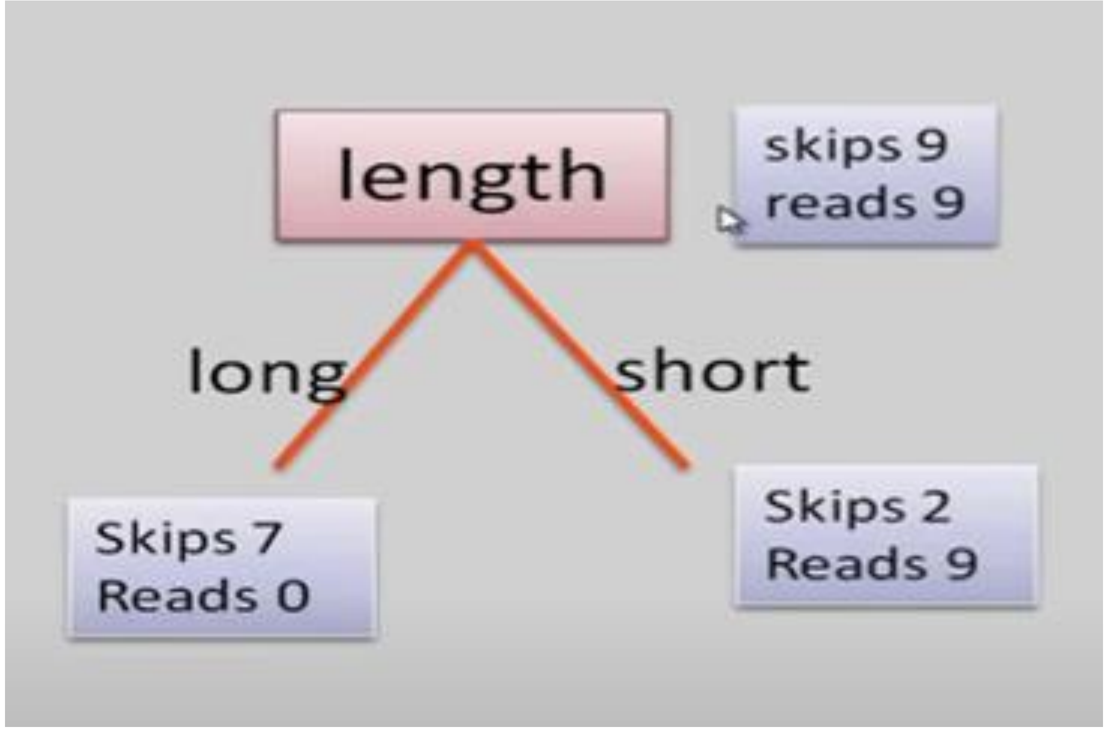
Gini Index

$$\text{Gini}(S) = 1 - \textstyle\sum_{i=1}^{c} p_i^2$$

$$\text{Information Gain} = \text{Gini}(\text{parent}) - \left(\frac{|S_{\text{left}}|}{|S|} \times \text{Gini}(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} \times \text{Gini}(S_{\text{right}}) \right)$$

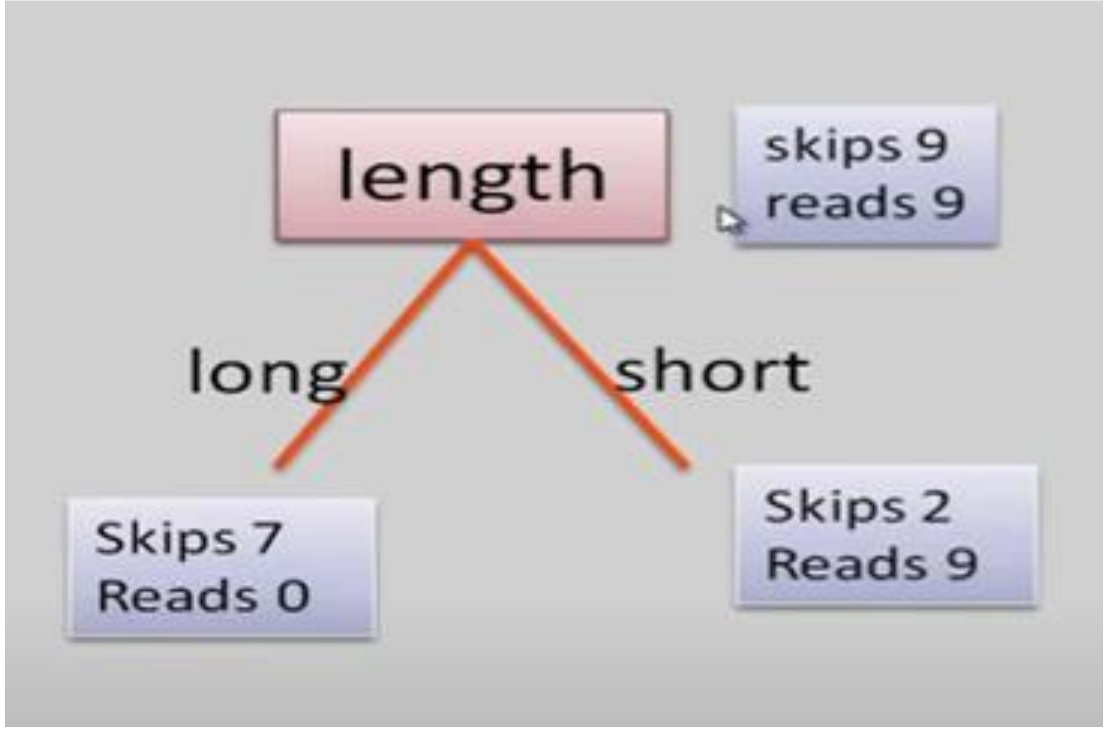
Example	*Author*	*Thread*	*Length*	*WhereRead*	*UserAction*
e_1	known	new	long	home	skips
e_2	unknown	new	short	work	reads
e_3	unknown	follow Up	long	work	skips
e_4	known	follow Up	long	home	skips
e_5	known	new	short	home	reads
e_6	known	follow Up	long	work	skips
e_7	unknown	follow Up	short	work	skips
e_8	unknown	new	short	work	reads
e_9	known	follow Up	long	home	skips
e_{10}	known	new	long	work	skips
e_{11}	unknown	follow Up	short	home	skips
e_{12}	known	new	long	work	skips
e_{13}	known	follow Up	short	home	reads
e_{14}	known	new	short	work	reads
e_{15}	known	new	short	home	reads
e_{16}	known	follow Up	short	work	reads
e_{17}	known	new	short	home	reads
e_{18}	unknown	new	short	work	reads
e_{19}	unknown	new	long	work	?
e_{20}	unknown	follow Up	long	home	?

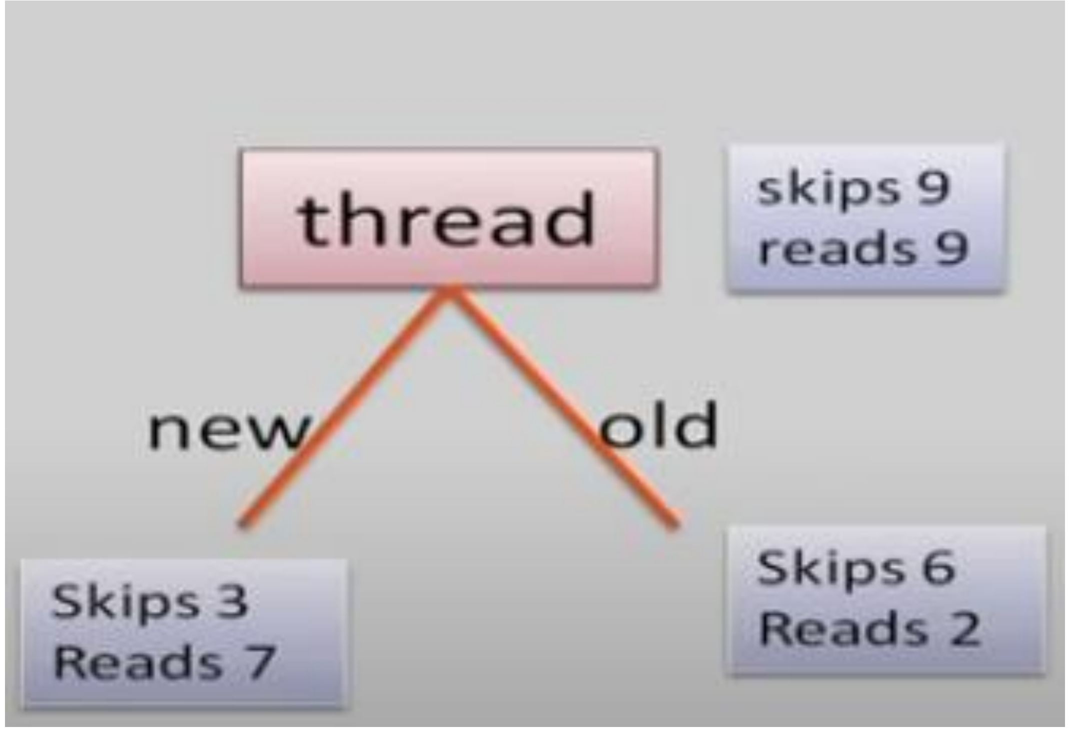
Possible splits



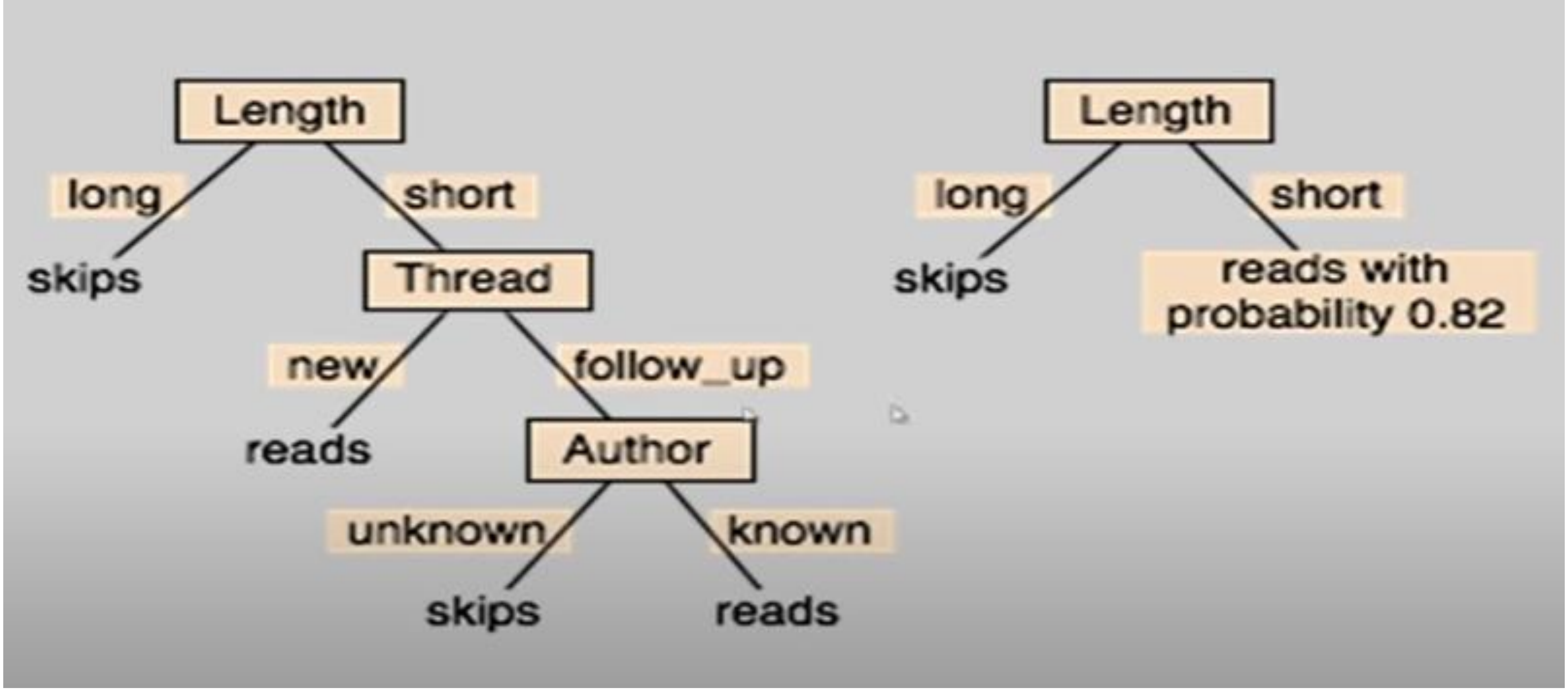
Example	Author	Thread	Length	WhereRead	UserAction
e_1	known	new	long	home	skips
e_2	unknown	new	short	work	reads
e_3	unknown	follow Up	long	work	skips
e_4	known	follow Up	long	home	skips
e_5	known	new	short	home	reads
e_6	known	follow Up	long	work	skips
e_7	unknown	follow Up	short	work	skips
e_8	unknown	new	short	work	reads
e_9	known	follow Up	long	home	skips
e_{10}	known	new	long	work	skips
e_{11}	unknown	follow Up	short	home	skips
e_{12}	known	new	long	work	skips
e_{13}	known	follow Up	short	home	reads
e_{14}	known	new	short	work	reads
e_{15}	known	new	short	home	reads
e_{16}	known	follow Up	short	work	reads
e_{17}	known	new	short	home	reads
e_{18}	unknown	new	short	work	reads
e_{19}	unknown	new	long	work	?
e_{20}	unknown	follow Up	long	home	?

Possible splits





Two examples of DT



ID	Fever	Cough	Breathing issues	Infected
1	NO	NO	NO	NO
2	YES	YES	YES	YES
3	YES	YES	NO	NO
4	YES	NO	YES	YES
5	YES	YES	YES	YES
6	NO	YES	NO	NO
7	YES	NO	YES	YES
8	YES	NO	YES	YES
9	NO	YES	YES	YES
10	YES	YES	NO	YES
11	NO	YES	NO	NO
12	NO	YES	YES	YES
13	NO	YES	YES	NO
14	YES	YES	NO	NO

$$\text{Entropy}(S) = -\sum p_i * \log_2(p_i)\ ;\ i = 1 \text{ to } n$$

where,

n is the total number of classes in the target column (in our case n = 2 i.e YES and NO)

p_i is the **probability of class 'i'** or the ratio of "*number of rows with class i in the target column*" to the "*total number of rows*" in the dataset.

Information Gain for a feature column A is calculated as:

$$IG(S, A) = \text{Entropy}(S) - \sum((|S_v| / |S|) * \text{Entropy}(S_v))$$

where S_v is the set of rows in S for which the feature column A has value v, $|S_v|$ is the number of rows in S_v and likewise $|S|$ is the number of rows in S.

Steps for ID3

- Calculate the Information Gain of each feature.
- Considering that all rows don't belong to the same class, split the dataset S into subsets using the feature for which the Information Gain is maximum.
- Make a decision tree node using the feature with the maximum Information gain.
- If all rows belong to the same class, make the current node as a leaf node with the class as its label.
- Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

ID	Fever	Cough	Breathing issues	Infected
1	NO	NO	NO	NO
2	YES	YES	YES	YES
3	YES	YES	NO	NO
4	YES	NO	YES	YES
5	YES	YES	YES	YES
6	NO	YES	NO	NO
7	YES	NO	YES	YES
8	YES	NO	YES	YES
9	NO	YES	YES	YES
10	YES	YES	NO	YES
11	NO	YES	NO	NO
12	NO	YES	YES	YES
13	NO	YES	YES	NO
14	YES	YES	NO	NO

From the total of 14 rows in the dataset **S**, there are **8** rows with the target value **YES** and **6** rows with the target value **NO**. The entropy of **S** is calculated as:

```
Entropy(S) = –
```
$$Entropy(S) = - \sum p_i * \log_2(p_i) \; ; \; i = 1 \text{ to } n$$
```
(6/14) = 0.99
```

IG calculation for Fever:

Fever	Cough	Breathing issues	Infected
NO	NO	NO	NO
NO	YES	NO	NO
NO	YES	YES	YES
NO	YES	NO	NO
NO	YES	YES	YES
NO	YES	YES	NO

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	YES	NO	NO
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES
YES	YES	NO	YES
YES	YES	NO	NO

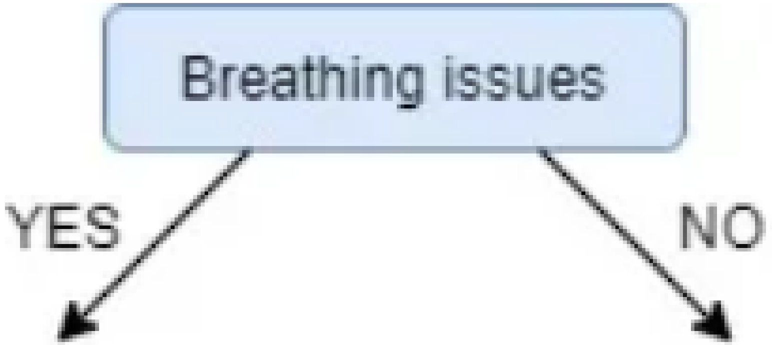
$$IG(S, A) = Entropy(S) - \sum((|S_v| \ / \ |S|) * Entropy(S_v))$$

- Calculate the IG for the features “Cough” and “Breathing issues”.

```
IG(S, Cough) = 0.04
IG(S, BreathingIssues) = 0.40
IG(S, Fever) = 0.13
```

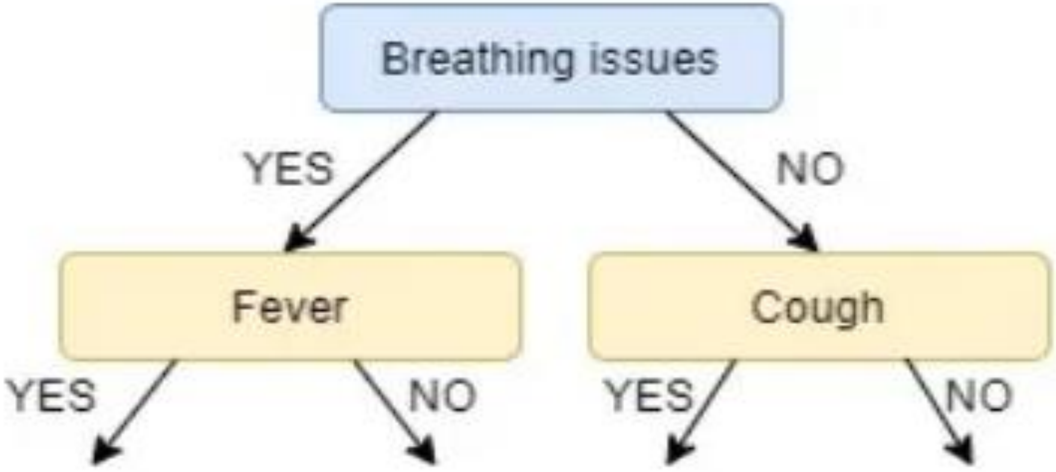
S_{BY}

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES
NO	YES	YES	YES
NO	YES	YES	YES
NO	YES	YES	NO



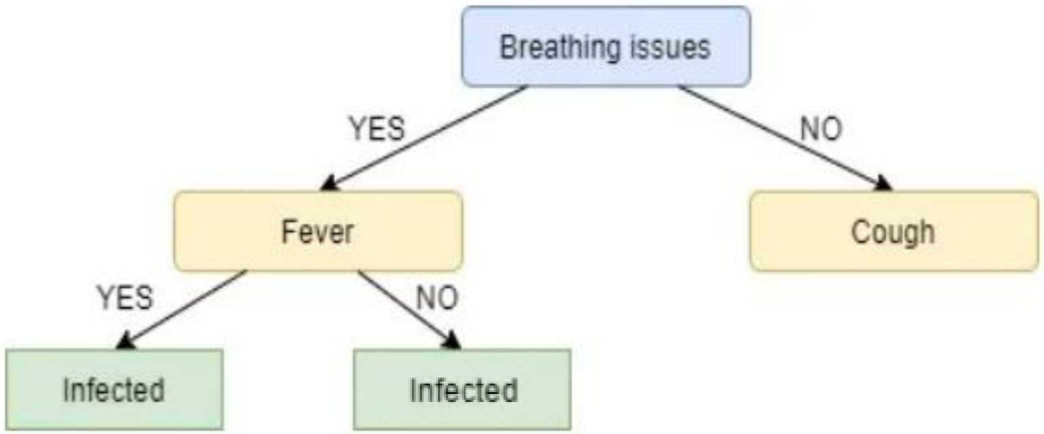
Fever or Cough?

```
IG(SBY, Fever) = 0.20
IG(SBY, Cough) = 0.09
```

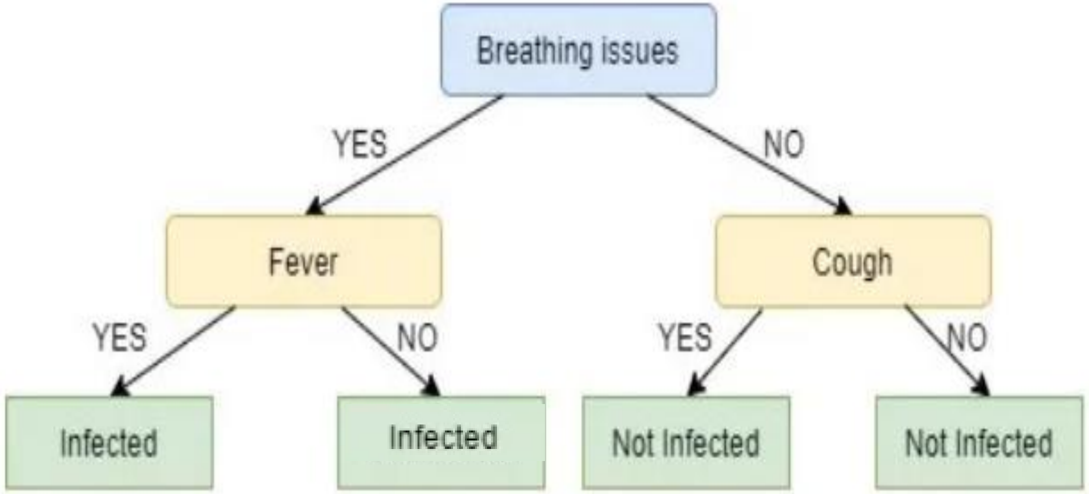


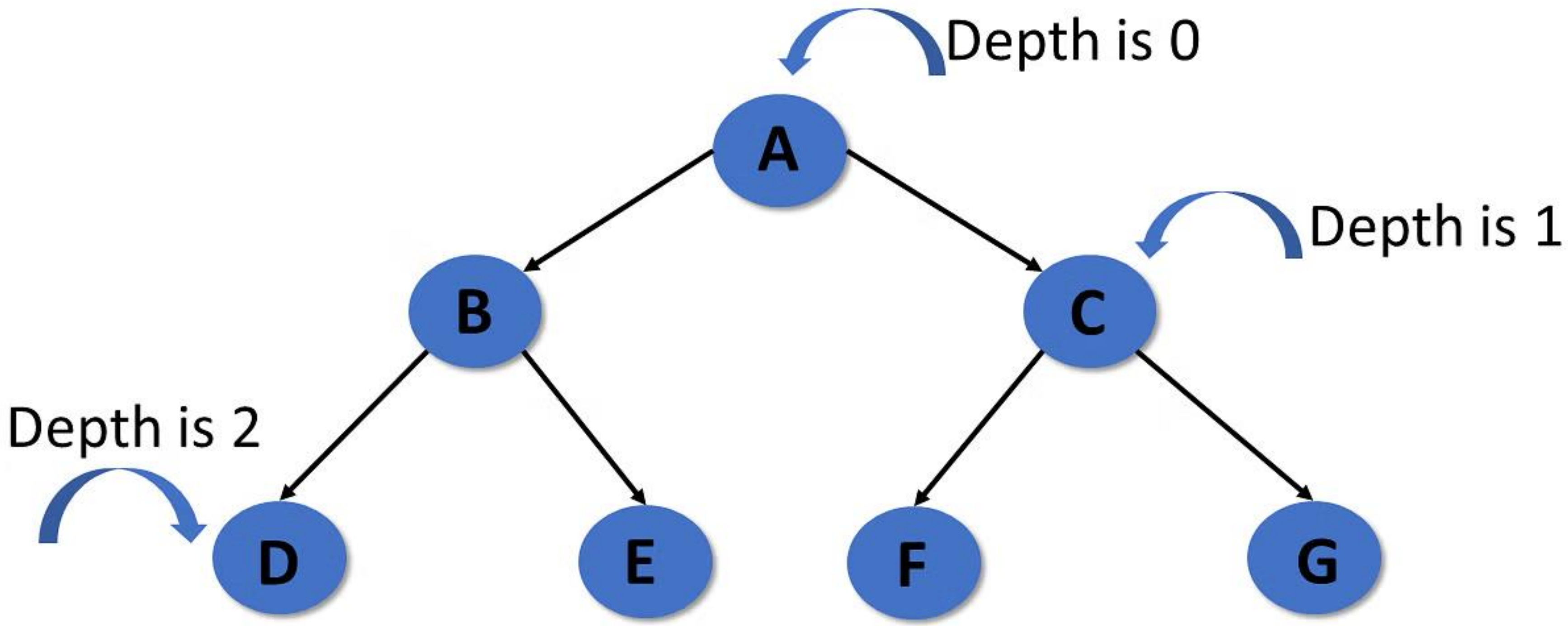
Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES

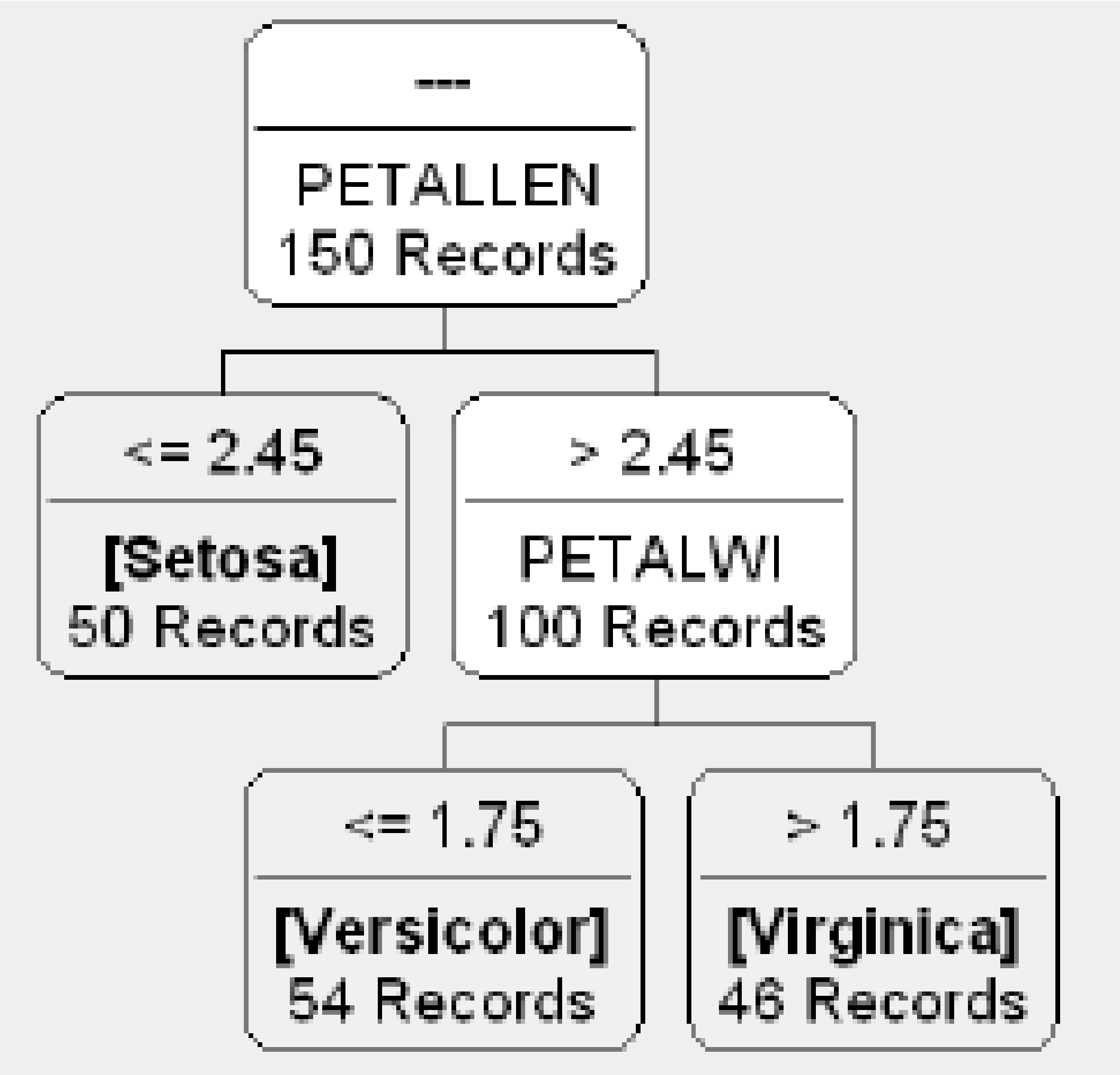
Fever	Cough	Breathing issues	Infected
NO	YES	YES	YES
NO	YES	YES	YES
NO	YES	YES	NO



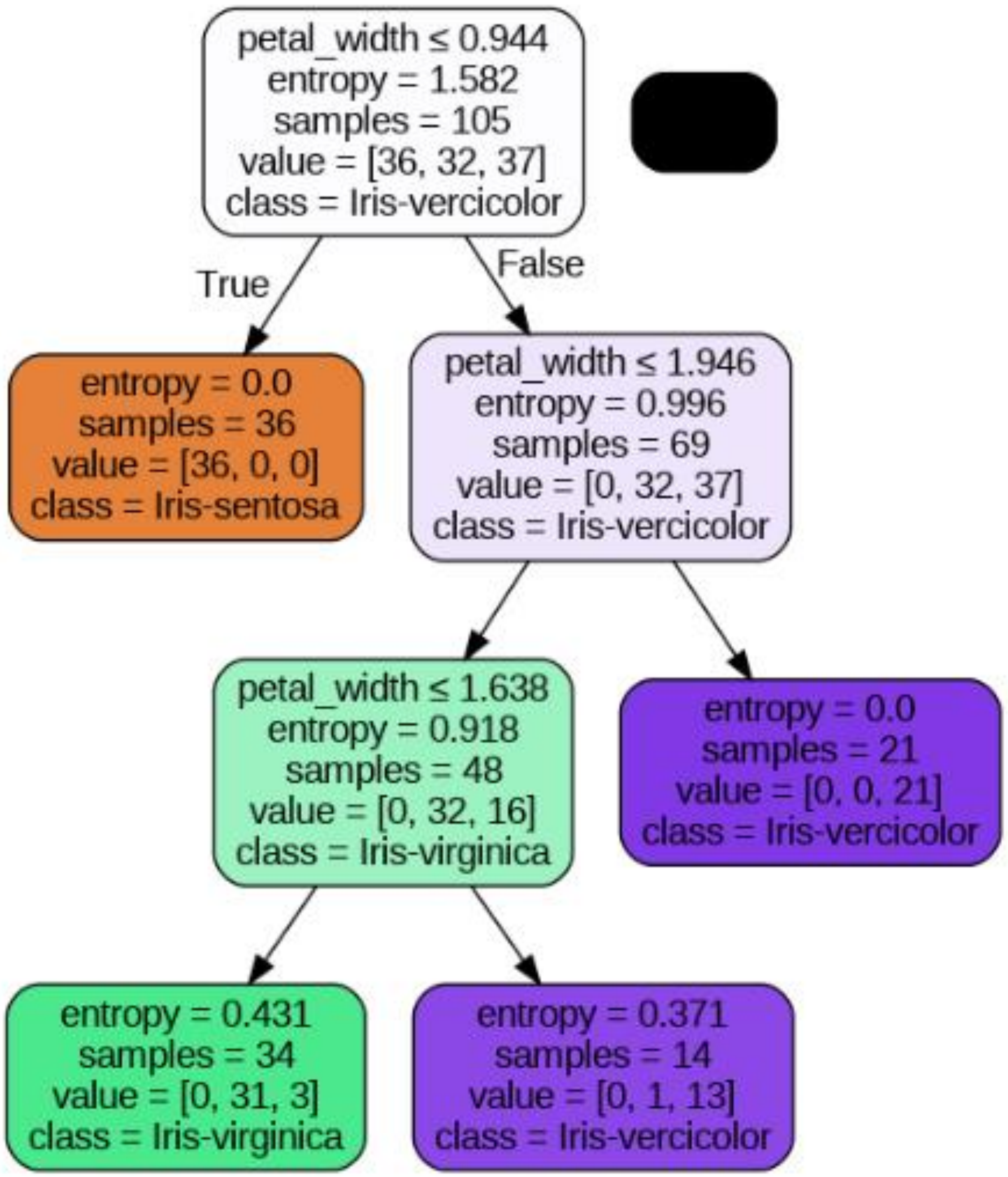
ID	Fever	Cough	Breathing issues	Infected
1	NO	NO	NO	NO
2	YES	YES	YES	YES
3	YES	YES	NO	NO
4	YES	NO	YES	YES
5	YES	YES	YES	YES
6	NO	YES	NO	NO
7	YES	NO	YES	YES
8	YES	NO	YES	YES
9	NO	YES	YES	YES
10	YES	YES	NO	YES
11	NO	YES	NO	NO
12	NO	YES	YES	YES
13	NO	YES	YES	NO
14	YES	YES	NO	NO




Depth is 0
A
Depth is 1
B
C
Depth is 2
D
E
F
G



PETALLEN
150 Records
<= 2.45
[Setosa]
50 Records
> 2.45
PETALWI
100 Records
<= 1.75
[Versicolor]
54 Records
> 1.75
[Virginica]
46 Records


petal_width ≤ 0.944
entropy = 1.582
samples = 105
value = [36, 32, 37]
class = Iris-vercicolor
True
False
entropy = 0.0
samples = 36
value = [36, 0, 0]
class = Iris-sentosa
petal_width ≤ 1.946
entropy = 0.996
samples = 69
value = [0, 32, 37]
class = Iris-vercicolor
petal_width ≤ 1.638
entropy = 0.918
samples = 48
value = [0, 32, 16]
class = Iris-virginica
entropy = 0.0
samples = 21
value = [0, 0, 21]
class = Iris-vercicolor
entropy = 0.431
samples = 34
value = [0, 31, 3]
class = Iris-virginica
entropy = 0.371
samples = 14
value = [0, 1, 13]
class = Iris-vercicolor

Steps in building a Tree

- Step-1: Begin the tree with the root node, say S, which contains the complete dataset.
- Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Step-3: Divide the S into subsets that contains possible values for the best attributes.
- Step-4: Generate the decision tree node, which contains the best attribute.
- Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3.
- Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node

C4.5

- Handling both continuous and discrete (categorical)
- Handling training data with missing attribute values
- Pruning trees after creation (bottom-up)
- Adjusts the splitting criteria at each node to handle imbalanced datasets better.
- Less complex trees
- Avoids the problem of overfitting
- Gain ratio is used to determine the best attribute to split on at each node.
    - The gain ratio is calculated as the information gain divided by the split information.

Drawbacks of Information Gain as attribute selection measure

- The information gain measure is biased toward tests with many outcomes.

- That is, it prefers to select attributes having a large number of values.

- C4.5, a successor of ID3, uses an extension to information gain known as ***gain ratio***, which attempts to overcome this bias.

Gain Ratio

$$\text{Information\_Gain}(A) = \text{Entropy\_Info}(T) - \text{Entropy\_Info}(T, A)$$

$$\text{Split\_Info}(T, A) = -\sum_{i=1}^{v} \frac{|A_i|}{|T|} \times \log_2 \frac{|A_i|}{|T|}$$

$$\text{Gain\_Ratio}(A) = \frac{\text{Info\_Gain}(A)}{\text{Split\_Info}(T, A)}$$

Features	*ID3*	*C4.5*	*CART*
Type of data	Categorical	Continuous and Categorical	continuous anc nominal attributes data
Speed	Low	Faster than ID3	Average
Boosting	Not supported	Not supported	Supported
Pruning	No	Pre-pruning	Post pruning
Missing Values	Can't deal with	Can't deal with	Can deal with
Formula	Use information entropy and information Gain	Use split info and gain ratio	Use Gini diversity index

CART

- Classification and Regression Trees
- **Classification Trees:** The tree is used to determine which "class" the target variable is most likely to fall into when it is continuous.
- **Regression trees:** These are used to predict a continuous variable's value.
- Can handle continuous attributes
- Uses Gini Index

$$\text{Gini}(S) = 1 - \sum_{i=1}^{c} p_i^2$$

$$\text{Information Gain} = \text{Gini}(\text{parent}) - \left(\frac{|S_{\text{left}}|}{|S|} \times \text{Gini}(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} \times \text{Gini}(S_{\text{right}}) \right)$$

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Decision Tree Using Gini Index

- Compute the Gini Index for the overall collection of training examples.
- There are four possible output variables Cinema, Tennis, Stay In and Shopping.
- The data has 6 instances of Cinema, 2 instances of Tennis, 1 instance of Stay In and 1 of shopping.
- $\text{Gini}(S) = 1 - [(6/10)^2 + (2/10)^2 + (1/10)^2 + (1/10)^2]$
  $= 0.58$

Computation of Gini Index for Weather Attribute

- It has three possible values of Sunny (3 examples), Rainy (3 examples) and Windy (4 examples).

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

- Computation of Gini Index for Weather Attribute

- It has three possible values of Sunny (3 examples), Rainy (3 examples) and Windy (4 examples).

- Computation of Gini Index for Weather Attribute
- Weighted Average:

[Sunny] 0.444 * (3/10) + [Rainy] 0.444 * (3/10) + [Windy] 0.375 * (4/10) = 0.416

- Computation of Gini Index for Parents Attribute

- It has two possible values of Yes (5 examples) and No (5 examples).

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Computation of Gini Index for Parents Attribute

- It has two possible values of Yes (5 examples) and No (5 examples).
  - For Parents = Yes, there are 5 examples, all with "Cinema".

$$Gini(S) = 1 - [(5/5)^2] = 0$$

  - For Parents = No, there are 2 examples with "Tennis", 1 example with "Stay in", "Shopping" and "Cinema" each

$$Gini(S) = 1 - [(2/5)^2 + (1/5)^2 + (1/5)^2 + (1/5)^2] = 0.72$$

- Weighted Average: [Yes] 0 * (5/10) + [No] 0.72 * (5/10) = 0.36

- Computation of Gini Index for Money Attribute

- It has two possible values of Rich (7 examples) and Poor (3 examples).

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

- Computation of Gini Index for Money Attribute

- It has two possible values of Rich (7 examples) and Poor (3 examples).
  - For Money = Poor, there are 3 examples with "Cinema".

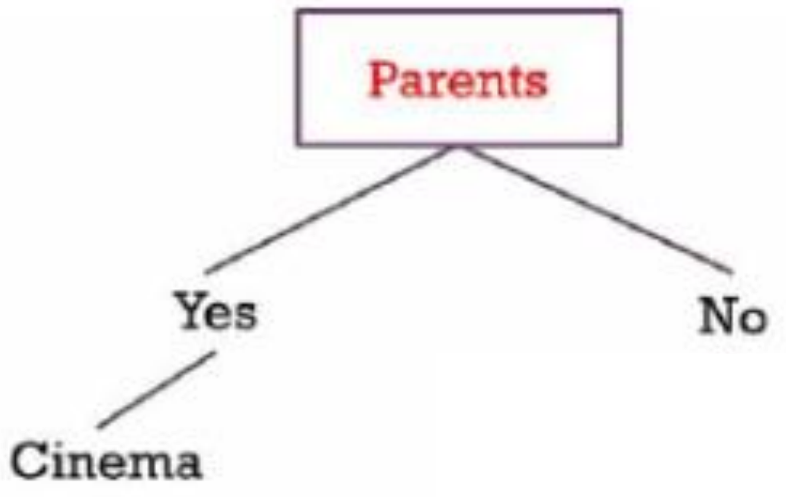
$$Gini(S) = 1 - [(3/3)^2] = 0$$

  - For Money = Rich, there are 2 examples with "Tennis", 3 examples with "Cinema" and 1 example with "Stay in", "Shopping" each

$$Gini(S) = 1 - [(2/7)^2 + (3/7)^2 + (1/7)^2 + (1/7)^2] = 0.694$$

- Weighted Average: [Poor] 0 * (3/10) + [Rich] 0.694 * (7/10) = 0.486

- For Weather
  - Gini Index: 0.416
- For Parents
  - Gini Index: 0.36
- For Money
  - Gini Index: 0.486



Weekend	Weather	Parents	Money	Decision
W1	Sunny	Yes	Rich	Cinema
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W6	Rainy	Yes	Poor	Cinema
W9	Windy	Yes	Rich	Cinema

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

- **Computation of Gini Index for Parents = No |Weather Attribute**

- It has three possible values of Sunny (2 examples), Windy (2 example) and Rainy (1 example).
  - For Parent= No |Weather = Sunny, there are 2 example with "Tennis", 1 example with "Stay In" and "Shopping".

$$Gini(S) = 1 - [(2/2)^2] = 0$$

  - For Parents = No | Weather = Rainy, there is 1 example with "Stay In".

$$Gini(S) = 1 - [(1/1)^2] = 0$$

- Computation of Gini Index for Parents = No | Weather Attribute
  - For Parents = No | Weather = Windy, there is 1 example with "Cinema" and 1 example with "Shopping".

  $\text{Gini}(S) = 1 - [(1/2)^2 + (1/2)^2] = 0.5$

- Weighted Average: [Sunny] 0 * (2/5) + [Rainy] 0 * (1/5) + [Windy] 0 .5* (2/5) = 0.2

- Computation of Gini Index for Parents = No | Money Attribute

- It has two possible values of Rich (4 examples) and Poor (1 example)

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

- Computation of Gini Index for Parents = No|Money Attribute

- It has two possible values of Rich (4 examples) and Poor (1 example)
  - For Parents = No| Money = Rich, there is 1 example with "Cinema" and "Shopping" each and 2 examples of "Tennis".

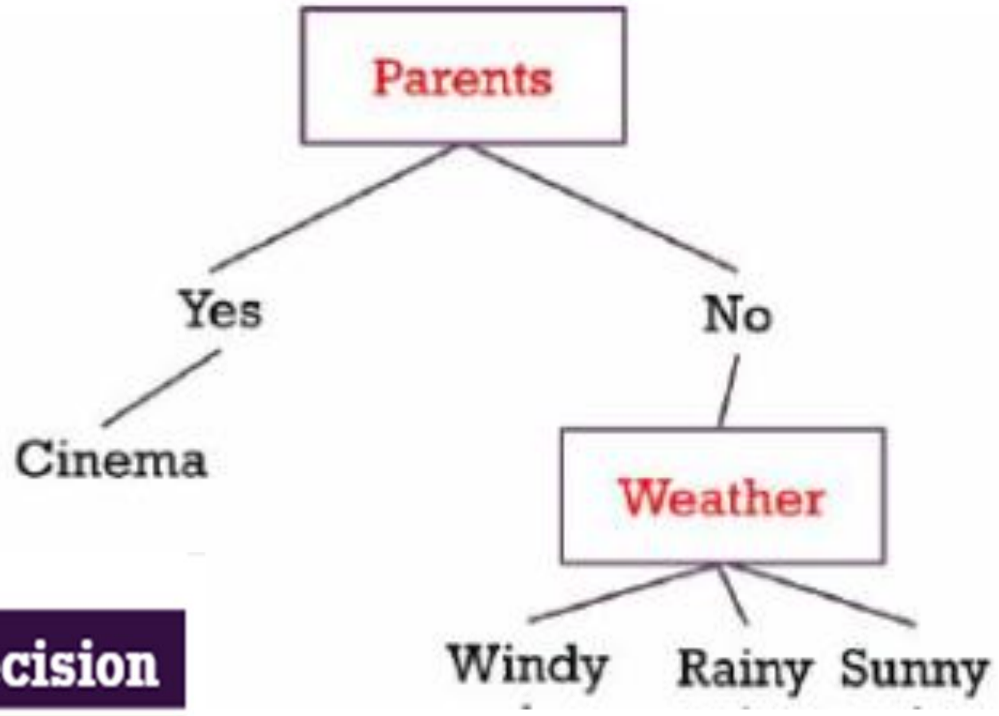
$$Gini(S) = 1 – [(1/4)^2 + (1/4)^2 + (2/4)^2] = 0.625$$

  - For Parents = No| Money = Poor, there is 1 example with "Cinema".

$$Gini(S) = 1 – [(1/1)^2] = 0$$

- Weighted Average: [Rich] 0.625 * (4/5) + [Poor] 0 * (1/5) = 0.5

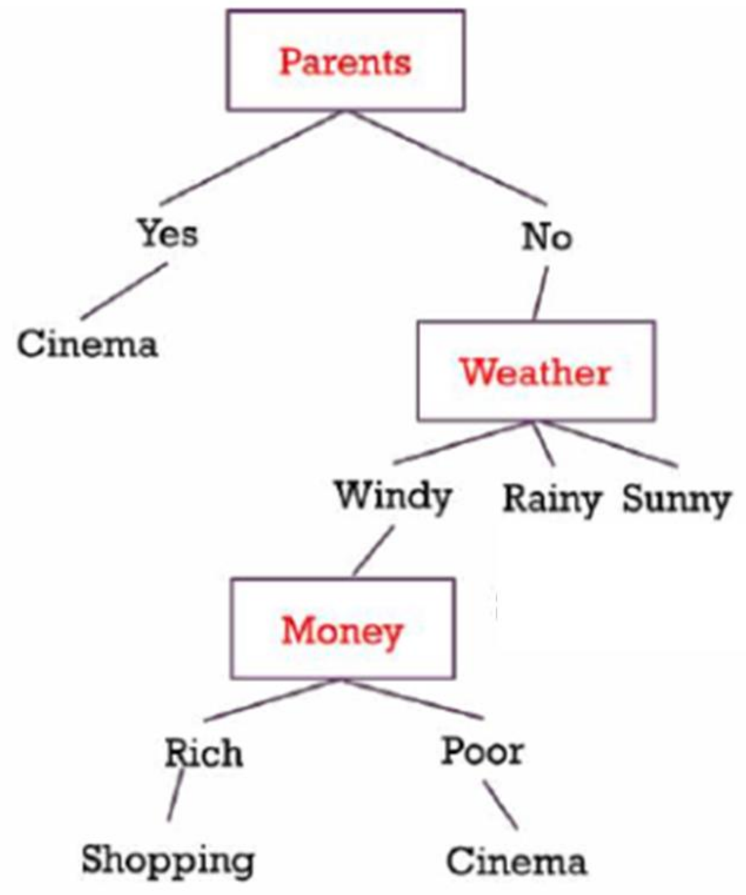
- For Parents = No | Weather
  - Gini Index: 0.2
- For Parents = No | Money
  - Gini Index: 0.5



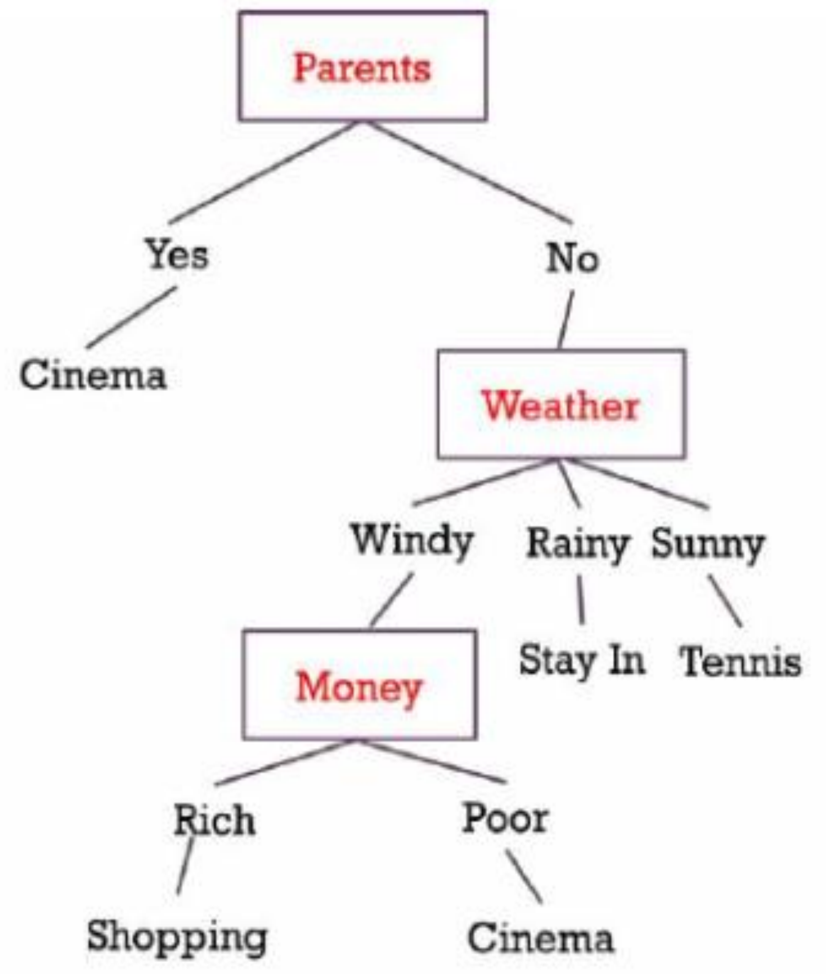
Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis

- For Parents = No | Weather
  - Gini Index: 0.2
- For Parents = No | Money
  - Gini Index: 0.5

Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis



Weekend	Weather	Parents	Money	Decision
W2	Sunny	No	Rich	Tennis
W5	Rainy	No	Rich	Stay In
W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W10	Sunny	No	Rich	Tennis



Outlook	Temperature	Humidity	Windy	Play
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

1. Calculate the Gini index for the original dataset:
   - There are 9 instances of 'Yes' and 5 instances of 'No'.
   - Gini index = 1 - (9/14)^2 - (5/14)^2 ≈ 0.459
2. Calculate the Gini index for each possible value of the 'Outlook' attribute and then calculate the weighted average:
   - For 'Sunny':
     - There are 5 instances of 'Sunny': 2 'Yes' and 3 'No'.
     - Gini index for 'Sunny' = 1 - (2/5)^2 - (3/5)^2 = 0.48
   - For 'Overcast':
     - There are 4 instances of 'Overcast': all 'Yes'.
     - Gini index for 'Overcast' = 1 - (4/4)^2 - (0/4)^2 = 0
   - For 'Rainy':
     - There are 5 instances of 'Rainy': 3 'Yes' and 2 'No'.
     - Gini index for 'Rainy' = 1 - (3/5)^2 - (2/5)^2 = 0.48
   - Weighted average = (5/14)*0.48 + (4/14)*0 + (5/14)*0.48 ≈ 0.171
3. Calculate the information gain:
   - Information Gain = Gini original - Weighted Gini = 0.459 - 0.171 ≈ 0.288

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

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

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

$$Gini(Outlook=Overcast) = 1 - (4/4)^2 - (0/4)^2 = 0$$

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

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

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

$$\text{Gini(Temp=Hot)} = 1 - (2/4)^2 - (2/4)^2 = 0.5$$

$$\text{Gini(Temp=Cool)} = 1 - (3/4)^2 - (1/4)^2 = 1 - 0.5625 - 0.0625 = 0.375$$

$$\text{Gini(Temp=Mild)} = 1 - (4/6)^2 - (2/6)^2 = 1 - 0.444 - 0.111 = 0.445$$

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

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

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

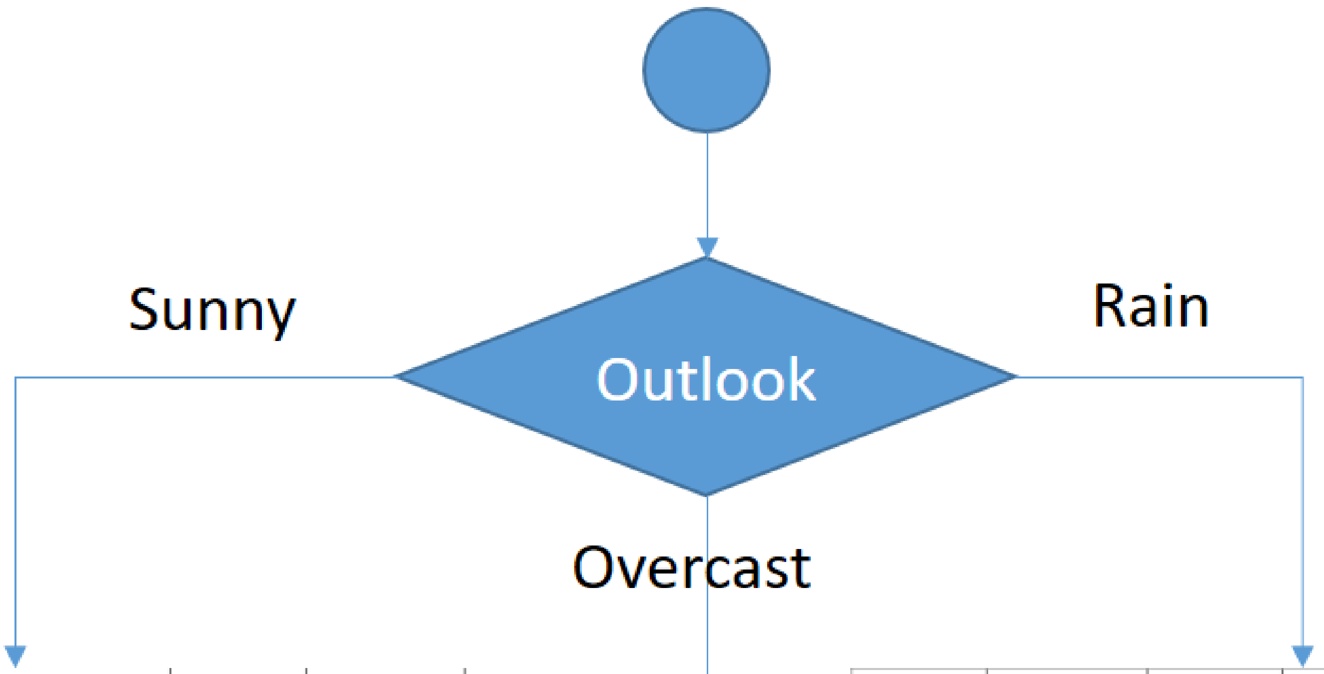
$$\text{Gini(Wind=Weak)} = 1 - (6/8)^2 - (2/8)^2 = 1 - 0.5625 - 0.062 = 0.375$$

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

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

Decision for root node

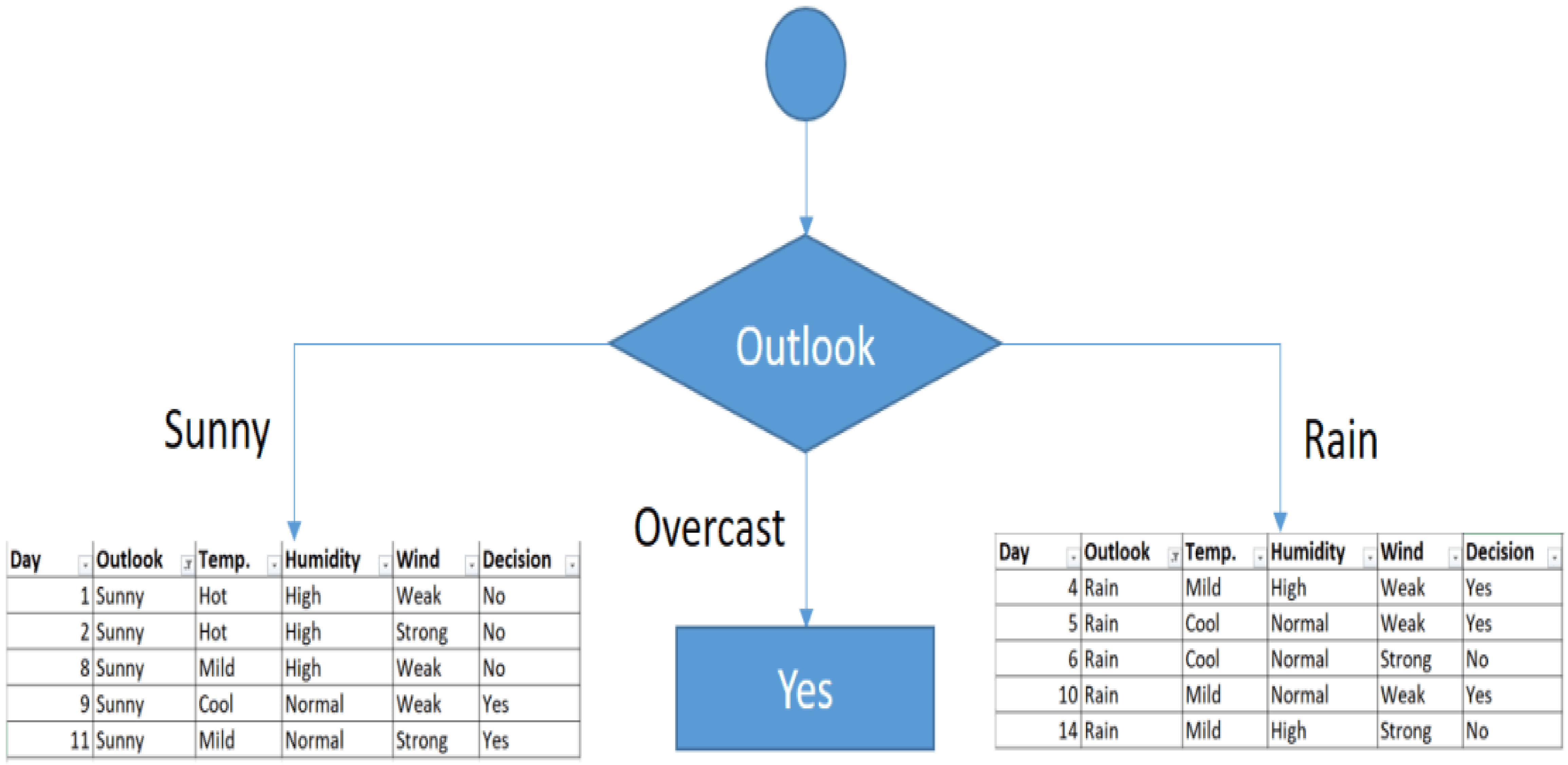
Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428



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



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
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

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

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

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

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

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

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) = $(3/5)\text{x}0 + (2/5)\text{x}0 = 0$

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

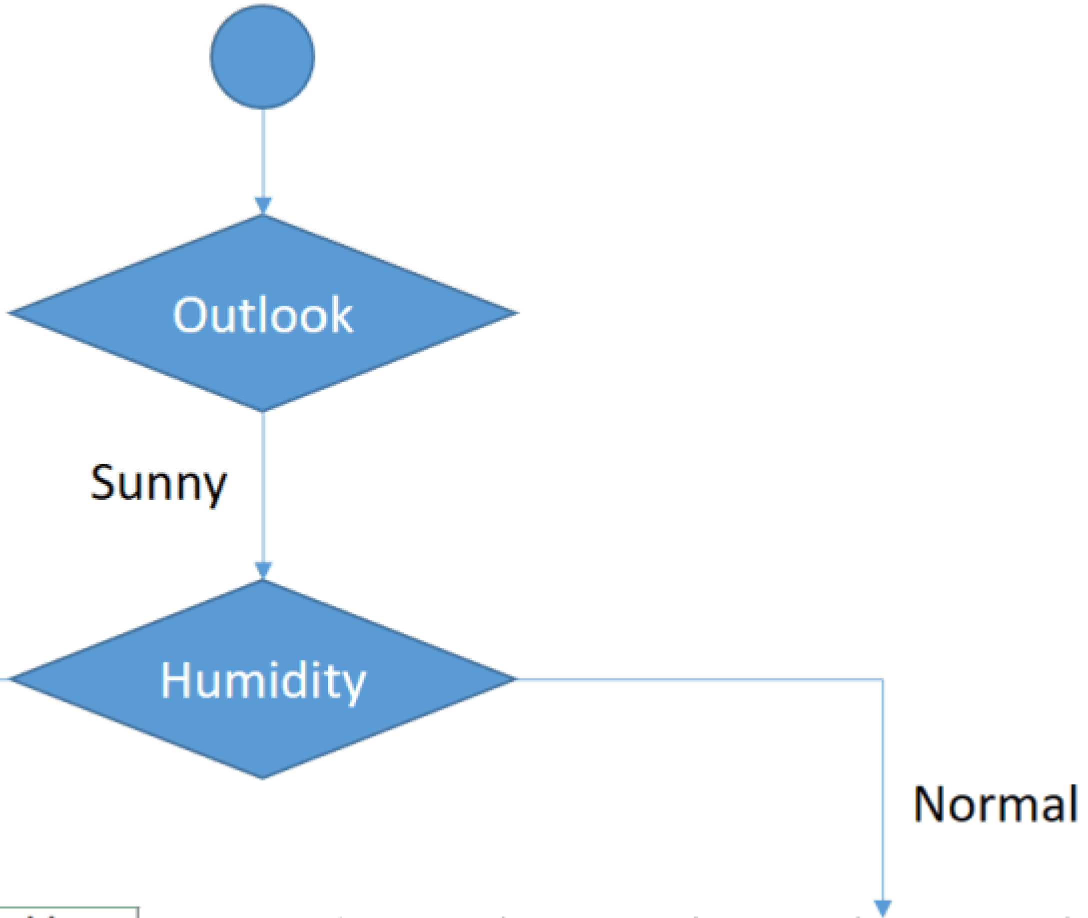
Gini(Outlook=Sunny and Wind=Weak) = $1 - (1/3)^2 - (2/3)^2 = 0.266$

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

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

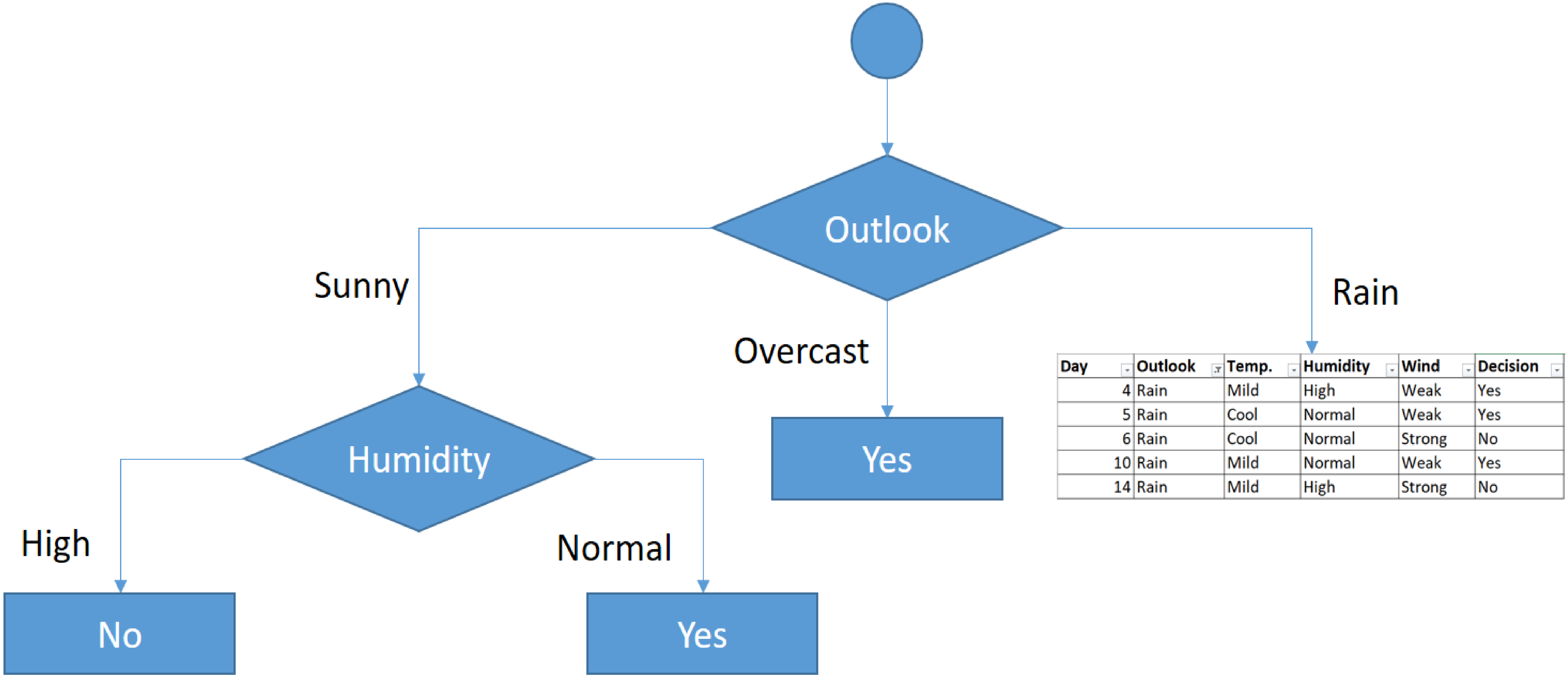
Decision on sunny - outlook

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
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
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

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

Gini of temprature for rain outlook

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

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

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

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

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) = $(2/5) \times 0.5 + (3/5) \times 0.444 = 0.466$

Gini of wind for rain outlook

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

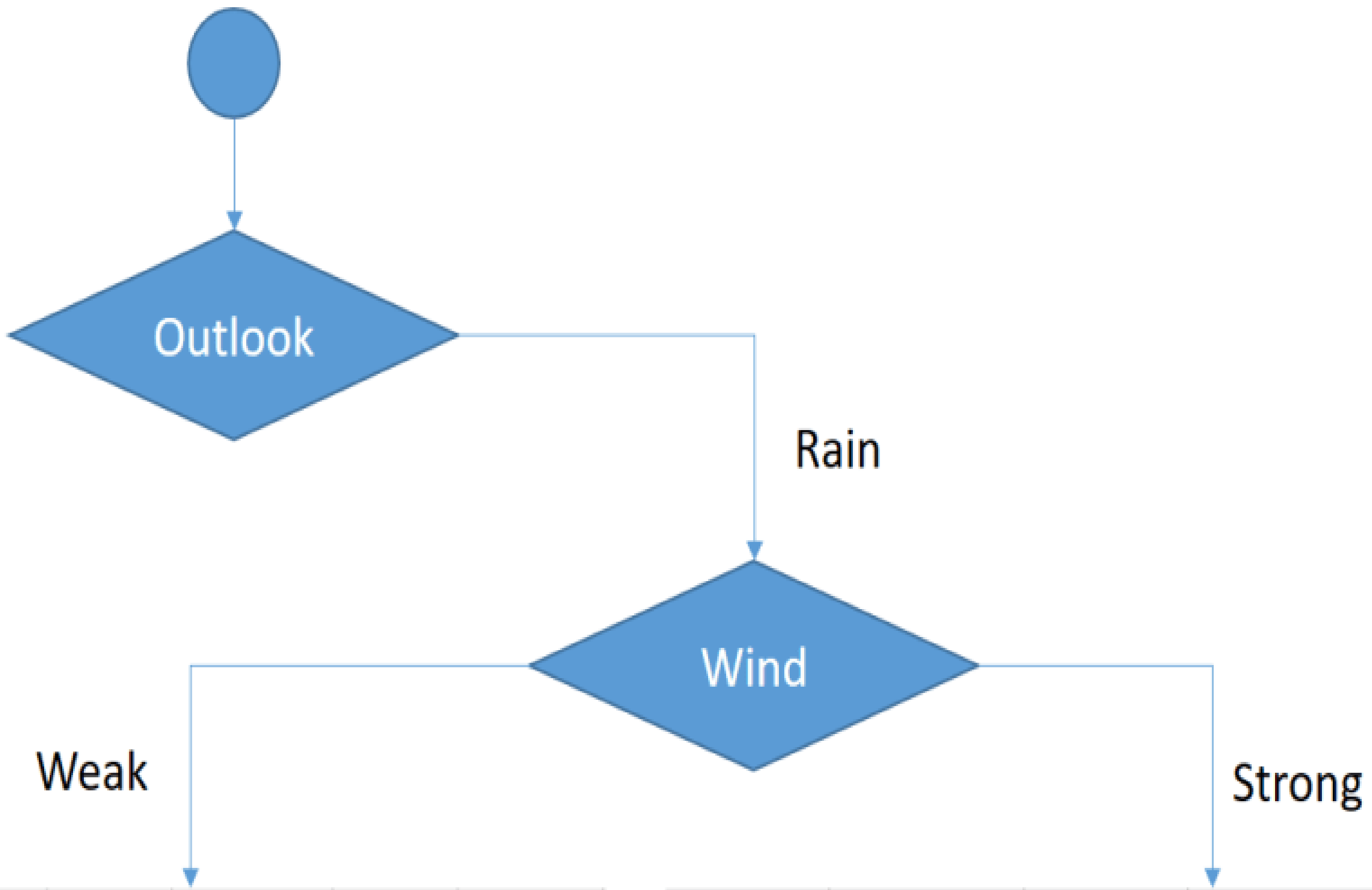
Gini(Outlook=Rain and Wind=Weak) = $1 - (3/3)^2 - (0/3)^2 = 0$

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

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

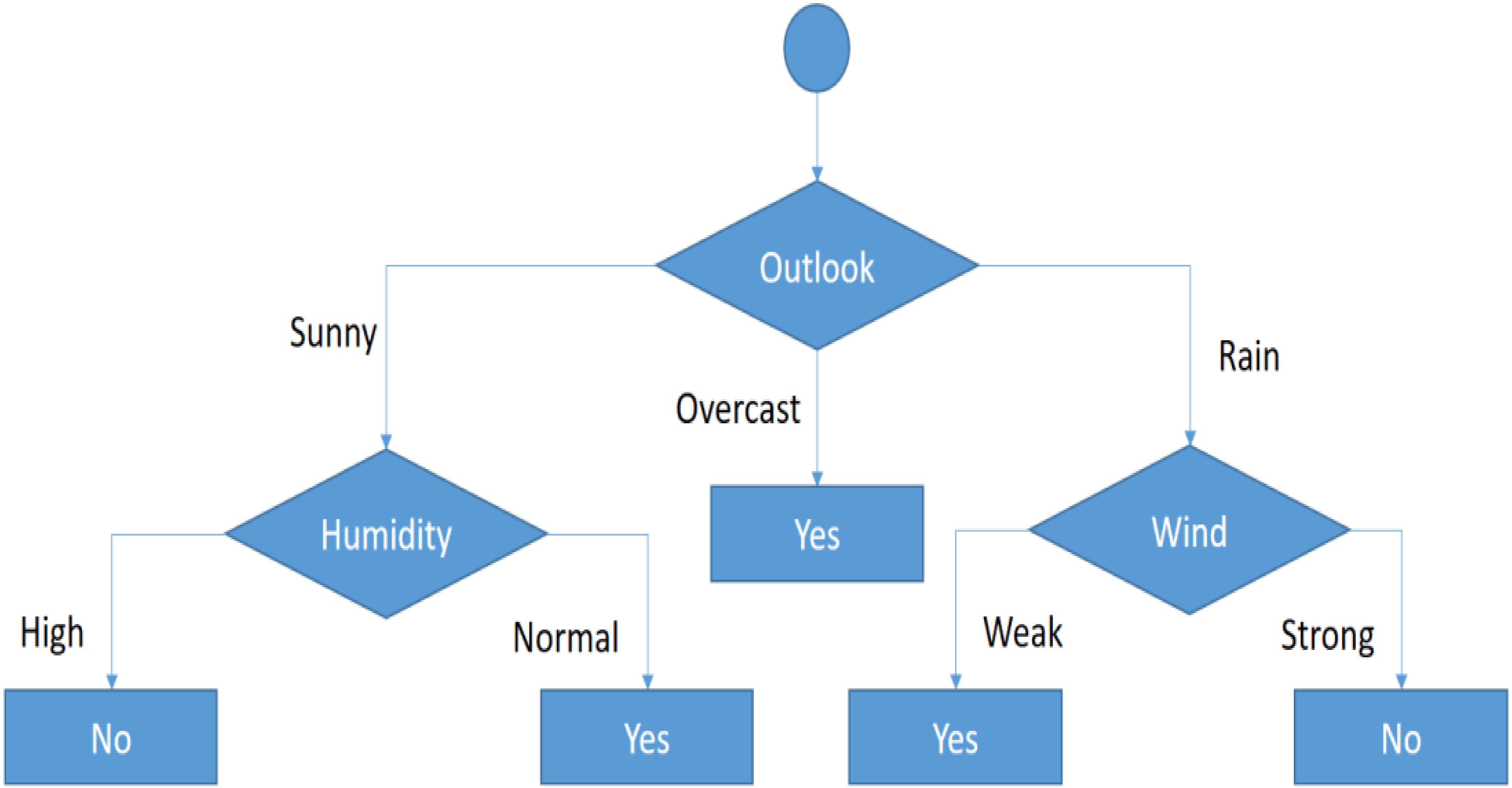
Decision for rain outlook

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


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

CART - Example

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

1. Calculate Gini for the dataset

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2$$

$$= 1 - 0.49 - 0.09$$

$$= 1 - 0.58$$

$$\text{Gini\_Index}(T) = 0.42$$

2. Calculate Gini Index for each attribute and each subset of the attribute

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥9	3	1
≥8	4	0
<8	0	2

$$\text{Gini\_Index}(T) = 1 - \sum_{i=1}^{m} P_i^2$$

$$\text{Gini\_Index}(T, A) = \frac{|S_1|}{|T|}\text{Gini}(S_1) + \frac{|S_2|}{|T|}\text{Gini}(S_2)$$

$$\begin{aligned}\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9, \geq 8\}) &= 1 - (7/8)^2 - (1/8)^2 \\ &= 1 - 0.7806 \\ &= 0.2194\end{aligned}$$

$$\begin{aligned}\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 9, \geq 8), <8\} &= (8/10) \times 0.2194 + (2/10) \times 0 \\ &= 0.17552 \\ &= 0\end{aligned}$$

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥ 9	3	1
≥ 8	4	0
<8	0	2

$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9, <8\}) = 1 - (3/6)^2 - (3/6)^2 = 1 - 0.5 = 0.5$$

$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 8\}) = 1 - (4/4)^2 - (0/4)^2 = 1 - 1 = 0$$

$$\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 9, <8), \geq 8\}) = (6/10) \times 0.5 + (4/10) \times 0 = 0.3$$

Categories of CGPA

CGPA	Job Offer = Yes	Job Offer = No
≥ 9	3	1
≥ 8	4	0
<8	0	2

$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 8, <8\}) = 1 - (4/6)^2 - (2/6)^2 = 1 - 0.555 = 0.445$$

$$\text{Gini\_Index}(T, \text{CGPA} \in \{\geq 9\}) = 1 - (3/4)^2 - (1/4)^2 = 1 - 0.625 = 0.375$$

$$\text{Gini\_Index}(T, \text{CGPA} \in \{(\geq 8, <8), \geq 9\}) = (6/10) \times 0.445 + (4/10) \times 0.375 = 0.417$$

Gini_Index of CGPA

Subsets		Gini_Index
$(\geq 9, \geq 8)$	<8	0.1755
$(\geq 9, <8)$	≥ 8	0.3
$(\geq 8, <8)$	≥ 9	0.417

3. Best splitting subset

4. Delta Gini for Best splitting subset

$$\Delta\text{Gini}(\text{CGPA}) = \text{Gini}(T) - \text{Gini}(T, \text{CGPA}) = 0.42 - 0.1755 = 0.2445$$

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Interactiveness

Interactiveness	Job Offer = Yes	Job Offer = No
Yes	5	1
No	2	2

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes}\}) = 1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2$$
$$= 1 - 0.72$$
$$= 0.28$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{No}\}) = 1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2$$
$$= 1 - 0.5$$
$$= 0.5$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes, No}\}) = \frac{6}{10}(0.28) + \frac{4}{10}(0.5)$$
$$= 0.168 + 0.2$$
$$= 0.368$$

$$\Delta\text{Gini}(\text{Interactiveness}) = \text{Gini}(T) - \text{Gini}(T, \text{Interactiveness})$$
$$= 0.42 - 0.368$$
$$= 0.052$$

CGPA	Interactive	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\} = 1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2$$
$$= 0.7544$$
$$= 0.2456$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Average}\}) = 1 - \left(\frac{1}{3}\right)^2 - \left(\frac{2}{3}\right)^2$$
$$= 1 - 0.555 = 0.445$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Good}\}, \text{Average})$$
$$= \left(\frac{7}{10}\right)^2 \times 0.2456 + \left(\frac{3}{10}\right) \times 0.445$$
$$= 0.3054$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2$$
$$= 1 - 0.52$$
$$= 0.48$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good}\}) = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2$$
$$= 1 - 0.68$$
$$= 0.32$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good, Average}\}, \text{Good})$$
$$= \left(\frac{5}{10}\right) \times 0.48 + \left(\frac{5}{10}\right) \times 0.32 = 0.40$$

Categories for Practical Knowledge

Practical Knowledge	Job Offer = Yes	Job Offer = No
Very Good	2	0
Good	4	1
Average	1	2

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good, Average}\}, \text{Very Good})$$
$$= \left(\frac{8}{10}\right) \times 0.4688 + \left(\frac{2}{10}\right) \times 0 = 0.3750$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Good, Average}\}) = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2$$
$$= 1 - 0.5312 = 0.4688$$

$$\text{Gini\_Index}(T, \text{Practical Knowledge} \in \{\text{Very Good}\}) = 1 - \left(\frac{2}{2}\right)^2 - \left(\frac{0}{2}\right)^2$$
$$= 1 - 1 = 0$$

Gini_Index for Practical Knowledge

Subsets		Gini_Index
(Very Good, Good)	Average	0.3054
(Very Good, Average)	Good	0.40
(Good, Average)	Very Good	0.3750

$$\Delta\text{Gini(Practical Knowledge)} = \text{Gini}(T) - \text{Gini}(T, \text{Practical Knowledge})$$
$$= 0.42 - 0.3054 = 0.1146$$

Categories for Communication Skills

Communication Skills	Job Offer = Yes	Job Offer = No
Good	4	1
Moderate	3	0
Poor	0	2

Gini-Index for Subsets of Communication Skills

Subsets		Gini_Index
(Good, Moderate)	Poor	0.1755
(Good, Poor)	Moderate	0.3429
(Moderate, Poor)	Good	0.40

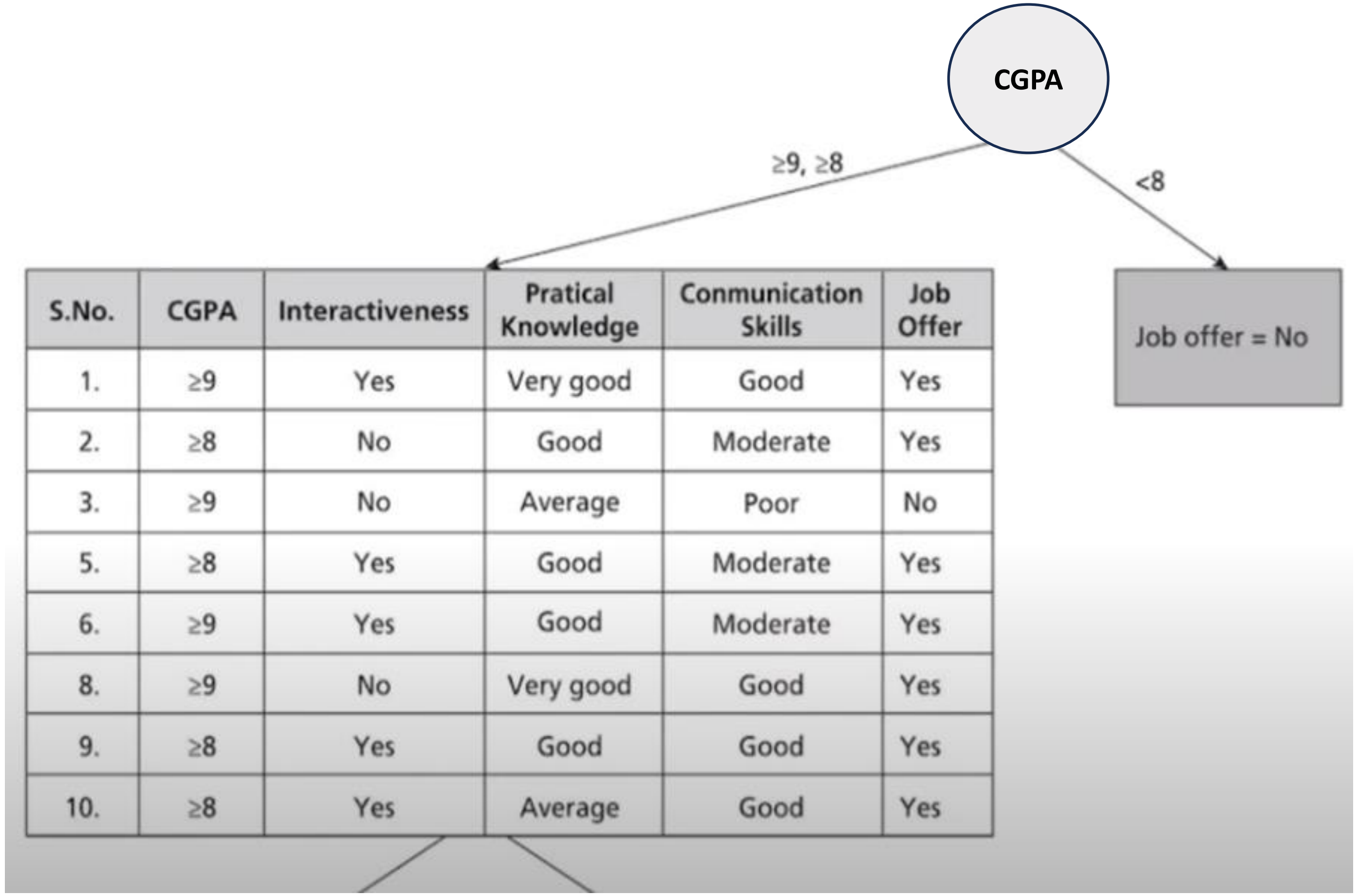
$$
\begin{aligned}
\Delta\text{Gini(Communication Skills)} &= \text{Gini}(T) - \text{Gini}(T, \text{Communication Skills}) \\
&= 0.42 - 0.1755 \\
&= 0.2445
\end{aligned}
$$

Gini_Index and ΔGini for all Attributes

Attribute	Gini_Index	ΔGini
CGPA	0.1755	0.2445
Interactiveness	0.368	0.052
Practical knowledge	0.3054	0.1146
Communication Skills	0.1755	0.2445

Subset **CGPA = {(>=9, >=8), <8}** is the best splitting subset

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
<8	No	Average	Good	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
<8	Yes	Good	Poor	No
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes



S.No.	CGPA	Interactiveness	Pratical Knowledge	Conmunication Skills	Job Offer
1.	≥9	Yes	Very good	Good	Yes
2.	≥8	No	Good	Moderate	Yes
3.	≥9	No	Average	Poor	No
5.	≥8	Yes	Good	Moderate	Yes
6.	≥9	Yes	Good	Moderate	Yes
8.	≥9	No	Very good	Good	Yes
9.	≥8	Yes	Good	Good	Yes
10.	≥8	Yes	Average	Good	Yes

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes

$$\text{Gini\_Index}(T) = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2$$
$$= 1 - 0.766 - 0.0156$$
$$= 1 - 0.58$$
$$\text{Gini\_Index}(T) = 0.2184$$

$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{No}\}) = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2$$
$$= 1 - 0.44 - 0.111 = 0.449$$
$$\text{Gini\_Index}(T, \text{Interactiveness} \in \{\text{Yes, No}\}) = \left(\frac{7}{8}\right) \times 0 + \left(\frac{1}{8}\right) \times 0.449$$
$$= 0.056$$

$$\Delta\text{Gini(Interactiveness)} =$$
$$\text{Gini}(T) - \text{Gini}(T, \text{Interactiveness})$$
$$= 0.2184 - 0.056 = 0.1624$$

Gini_Index for Subsets of Practical Knowledge

Subsets		Gini_Index
(Very Good, Good)	Average	0.125
(Very Good, Average)	Good	0.1875
(Good, Average)	Very Good	0.2085

$$\begin{aligned}\Delta \text{Gini(Practical Knowledge)} &= \text{Gini}(T) - \text{Gini}(T, \text{Practical Knowledge}) \\ &= 0.2184 - 0.125 \\ &= 0.0934\end{aligned}$$

Gini_Index for Subsets of Communication Skills

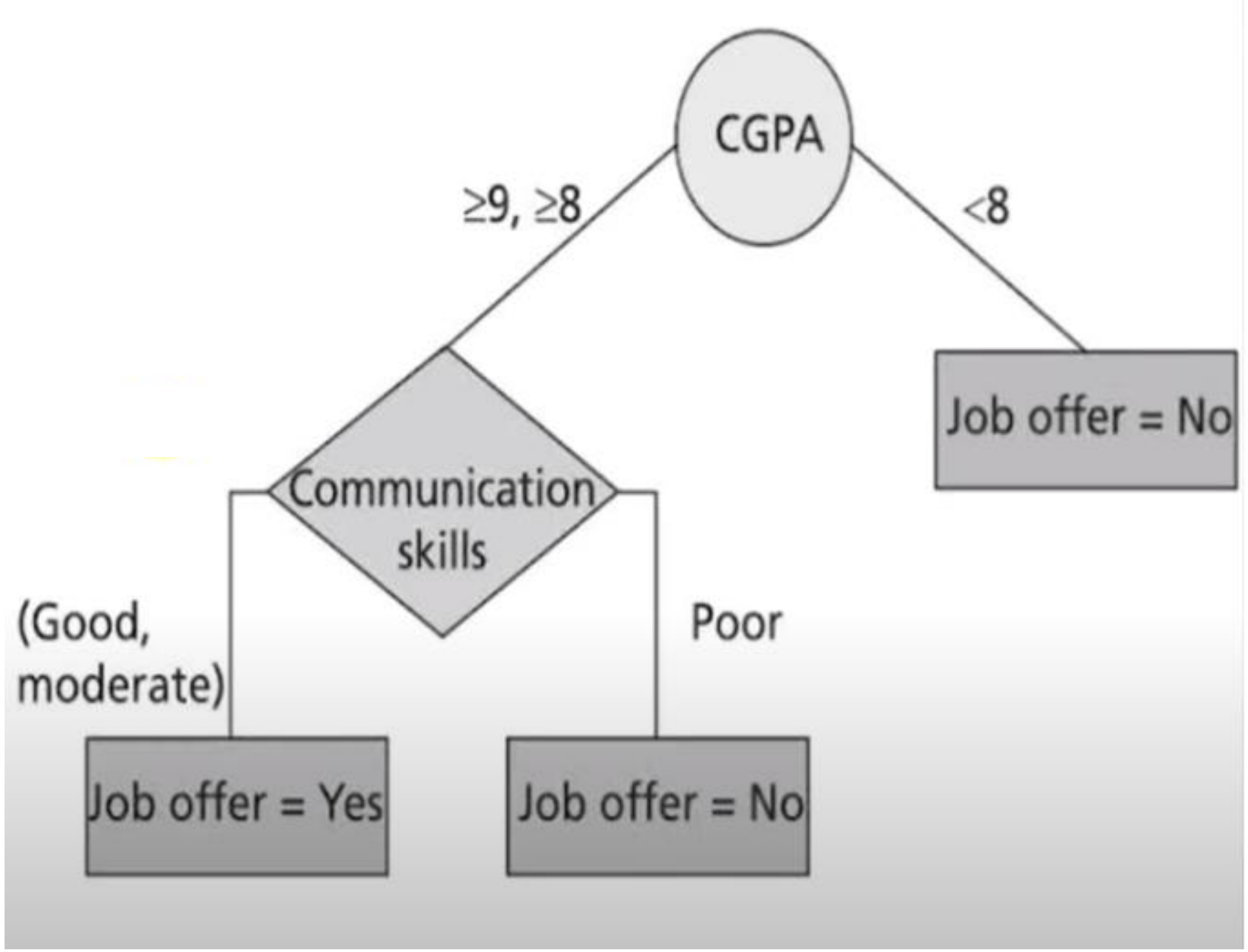
Subsets		Gini_Index
(Good, Moderate)	Poor	0
(Good, Poor)	Moderate	0.2
(Moderate, Poor)	Good	0.1875

$$\begin{aligned}\Delta \text{Gini(Communication Skills)} &= \text{Gini}(T) - \text{Gini}(T, \text{Communication Skills}) \\ &= 0.2184 - 0 = 0.2184\end{aligned}$$

Gini_Index and ΔGini Values for All Attributes

Attribute	Gini_Index	ΔGini
Interactiveness	0.056	0.1624
Practical knowledge	0.125	0.0934
Communication Skills	0	0.2184

CGPA	Inter active	Practical Knowledge	Comm Skills	Job Offer
>=9	Yes	Very good	Good	Yes
>=8	No	Good	Moderate	Yes
>=9	No	Average	Poor	No
>=8	Yes	Good	Moderate	Yes
>=9	Yes	Good	Moderate	Yes
>=9	No	Very good	Good	Yes
>=8	Yes	Good	Good	Yes
>=8	Yes	Average	Good	Yes



CART

age	income	student	credit_rating	buys_computer
youth	high	no	fair	no
youth	high	no	excellent	no
middle_aged	high	no	fair	yes
senior	medium	no	fair	yes
senior	low	yes	fair	yes
senior	low	yes	excellent	no
middle_aged	low	yes	excellent	yes
youth	medium	no	fair	no
youth	low	yes	fair	yes
senior	medium	yes	fair	yes
youth	medium	yes	excellent	yes
middle_aged	medium	no	excellent	yes
middle_aged	high	yes	fair	yes
senior	medium	no	excellent	no

Class **P** = 9: buys_computer = "**yes**"

Class **N** = 5: buys_computer = "**no**"

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2,$$

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

$$\Delta gini(A) = gini(D) - gini_A(D)$$

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = \mathbf{0.459}$$

CART- Regression Trees

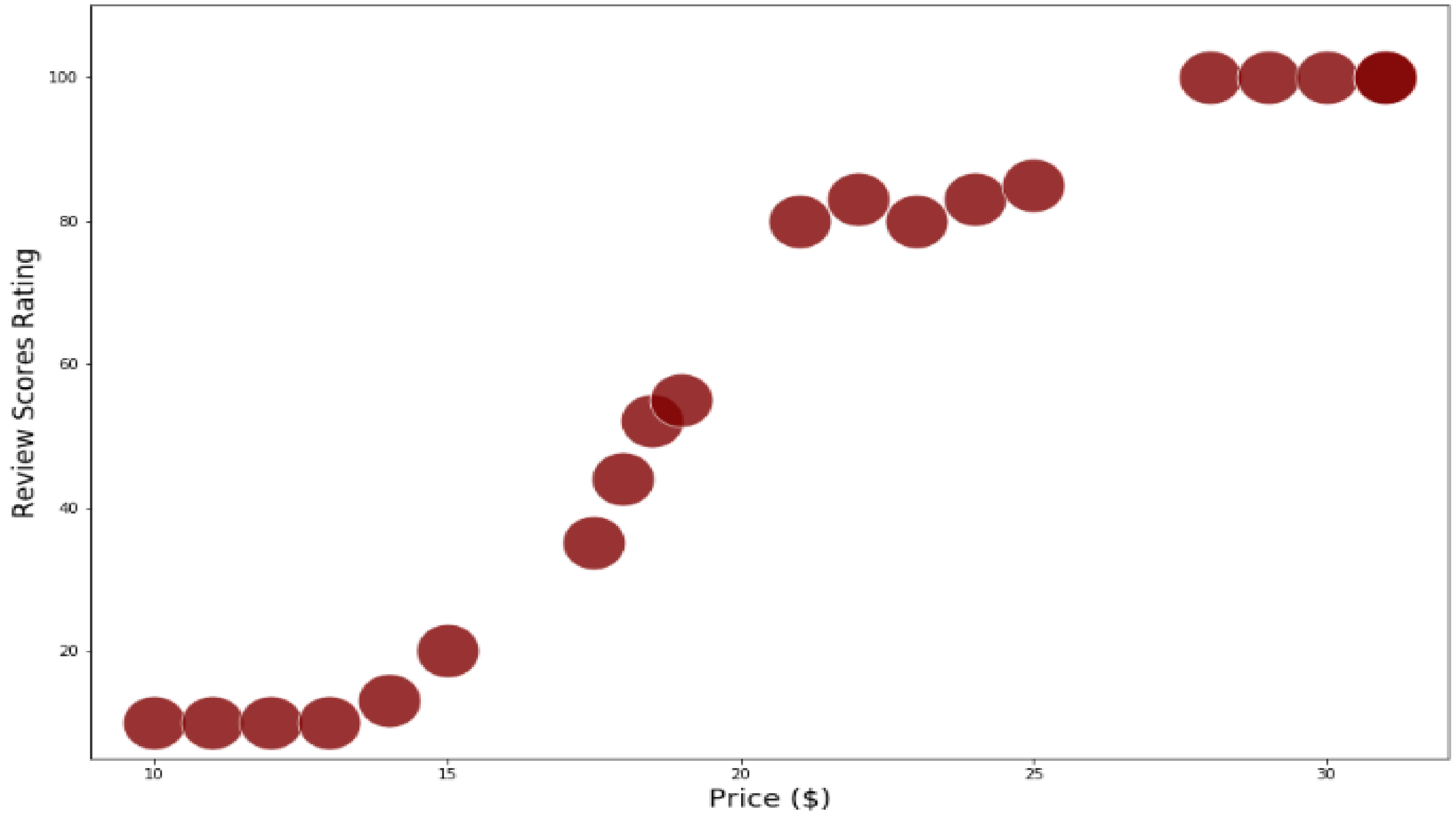
- CART in classification cases uses **Gini Impurity**
- CART in regression cases uses **least squares**
- Splits are chosen to minimize the **residual sum of squares** between the observation and the mean in each node.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2$$

CART- Regression Trees

	Price ($)	Review Scores Rating
0	10.0	10
1	11.0	10
2	12.0	10
3	13.0	10
4	14.0	13
5	15.0	20
6	17.5	35
7	18.0	44
8	18.5	52
9	19.0	55
10	21.0	80
11	22.0	83
12	23.0	80
13	24.0	83
14	25.0	85
15	28.0	100
16	29.0	100
17	30.0	100
18	31.0	100
19	31.0	100



R1

	Price ($)	Review Scores Rating
0	10.0	10
1	11.0	10
2	12.0	10
3	13.0	10
4	14.0	13
5	15.0	20
6	17.5	35
7	18.0	44
8	18.5	52
9	19.0	55
10	21.0	80
11	22.0	83
12	23.0	80
13	24.0	83
14	25.0	85
15	28.0	100
16	29.0	100
17	30.0	100
18	31.0	100
19	31.0	100

R2

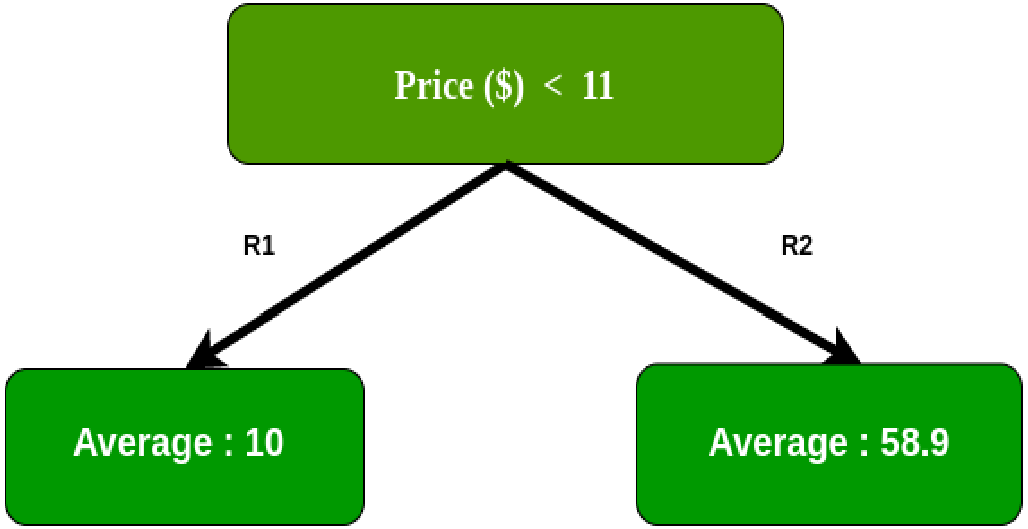
CART- Regression Trees

first, calculate **RSS** by splitting into two regions, **Start within index 0**

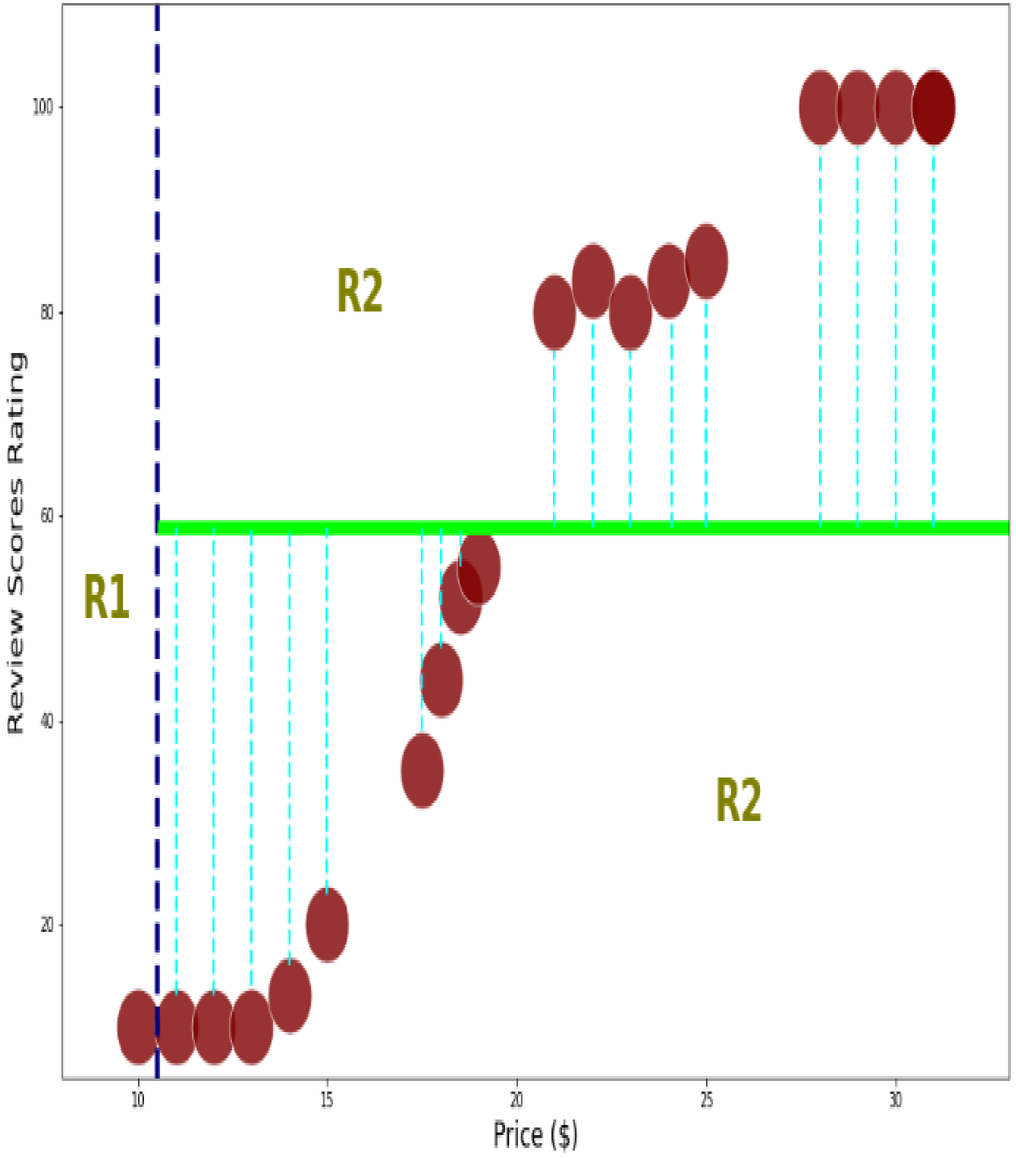
R1

	Price ($)	Review Scores Rating
0	10.0	10
1	11.0	10
2	12.0	10
3	13.0	10
4	14.0	13
5	15.0	20
6	17.5	35
7	18.0	44
8	18.5	52
9	19.0	55
10	21.0	80
11	22.0	83
12	23.0	80
13	24.0	83
14	25.0	85
15	28.0	100
16	29.0	100
17	30.0	100
18	31.0	100
19	31.0	100

R2



CART- Regression Trees



$$\varepsilon = \sum_{i=1}^{n} (actual\ value - average\ value\ in\ each\ region)^2$$

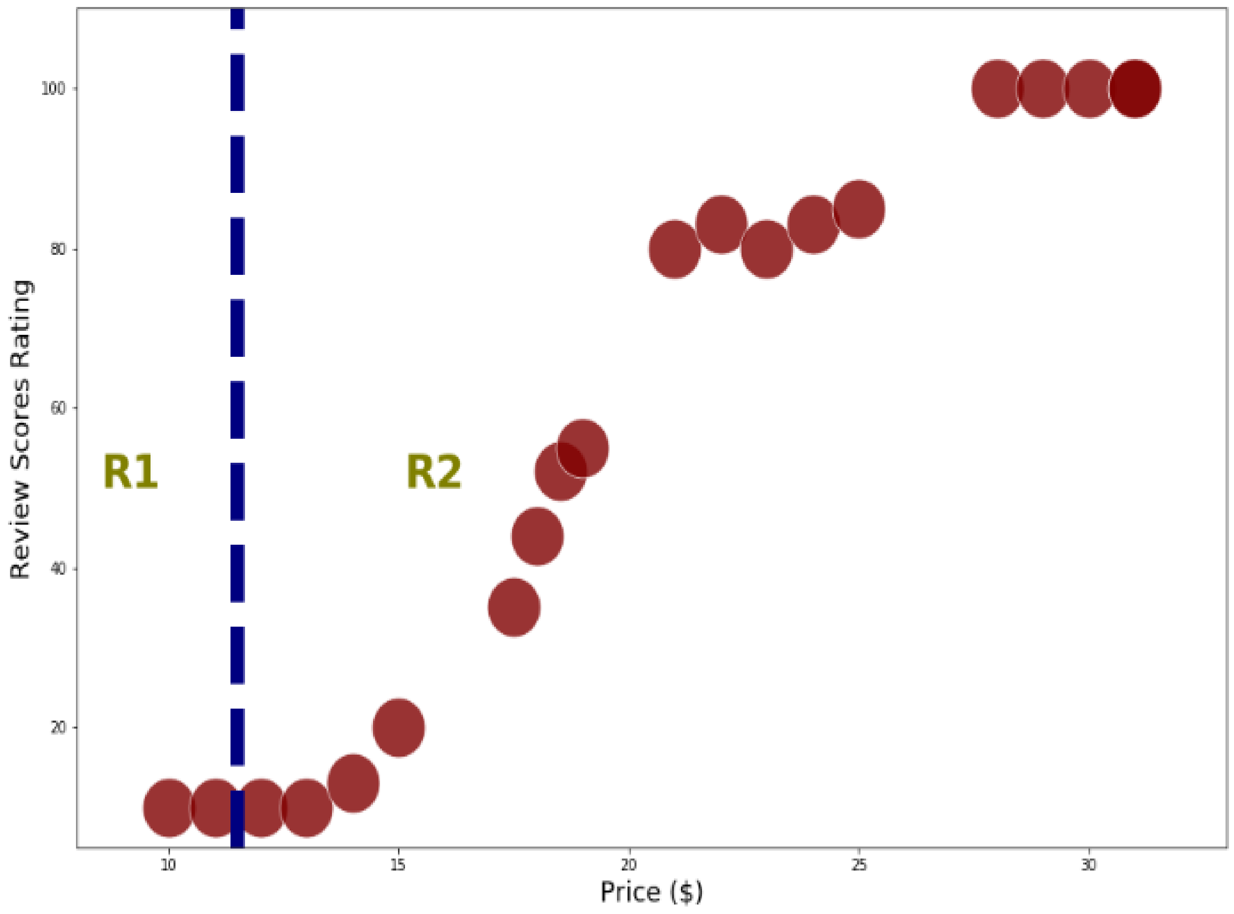
$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2$$

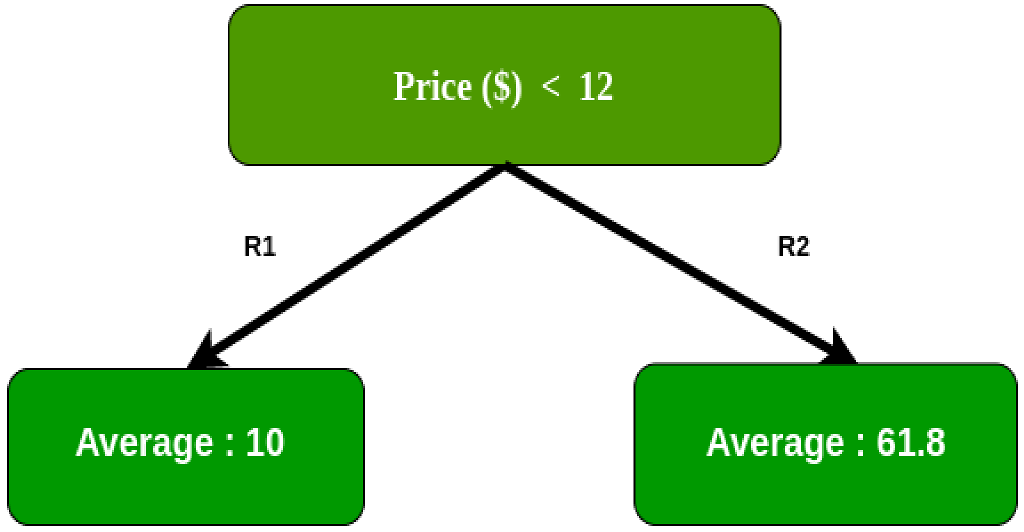
$$RSS = (10 - 10)^2 + (10 - 58.9)^2 + (10 - 58.9)^2 \ldots (10 - 58.9)^2 = 21139.78$$

CART- Regression Trees

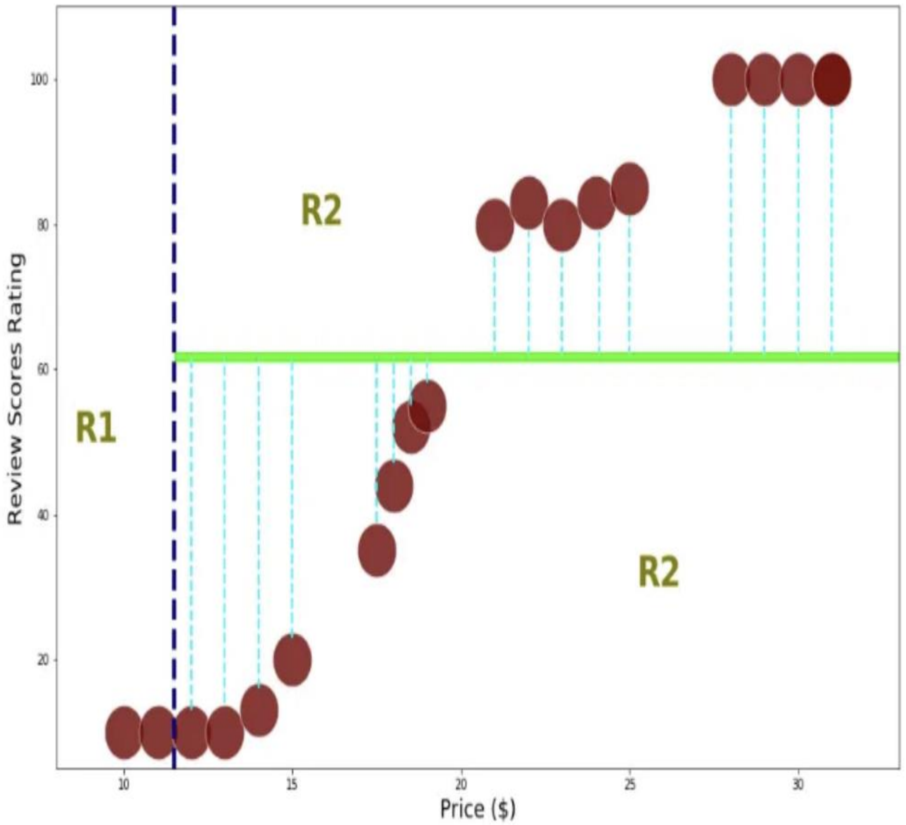
Start within index 1

calculate **RSS** by split into two regions within index 1





CART- Regression Trees



$$\varepsilon = \sum_{i=1}^{n} (actual\ value - average\ value\ in\ each\ region)^2$$

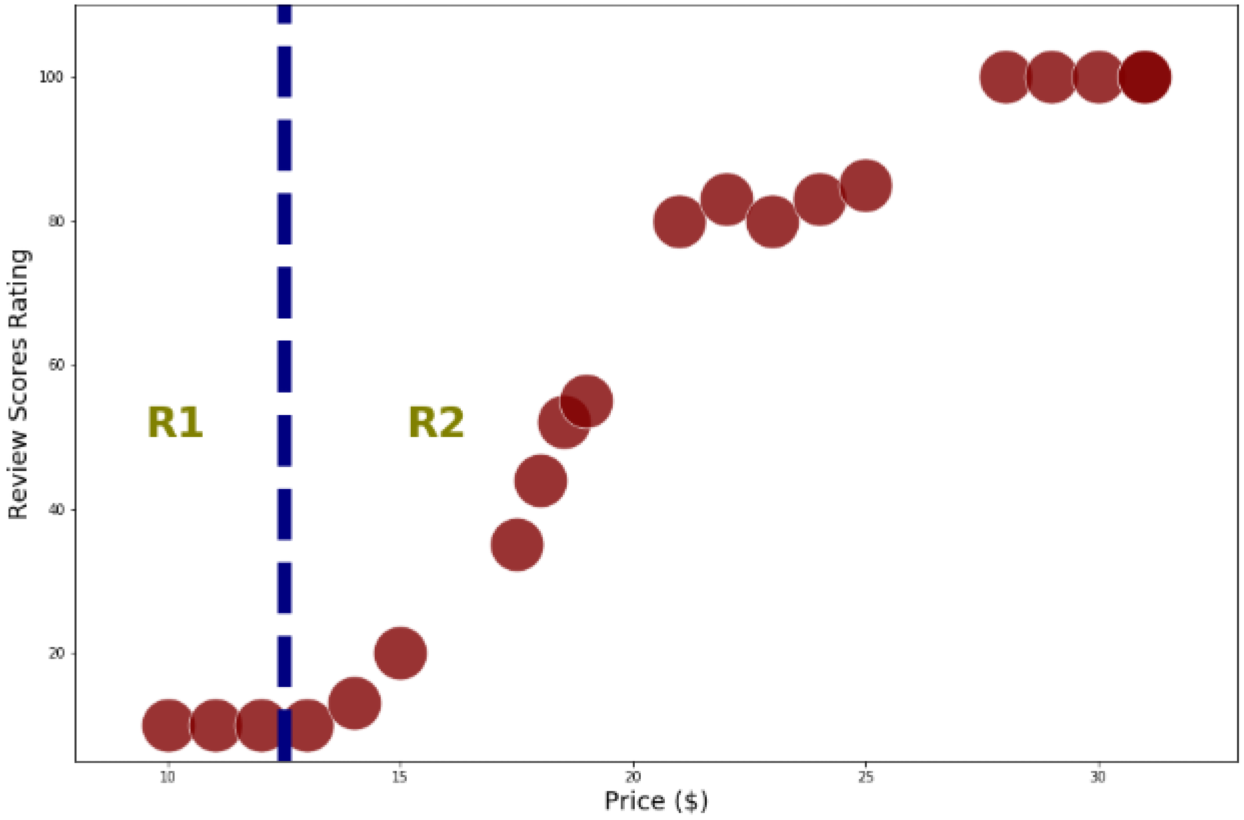
$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2$$

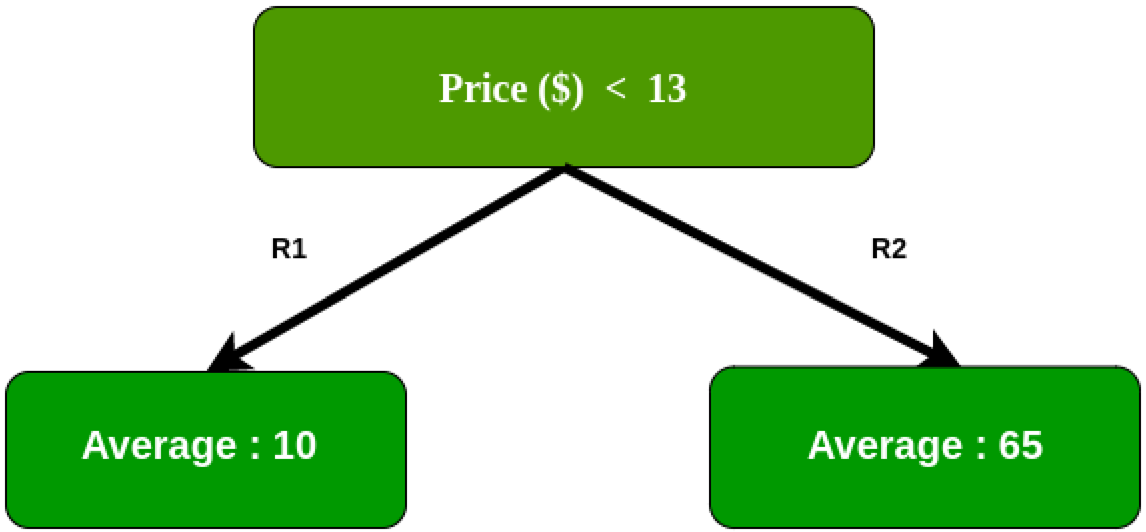
$$RSS = (10 - 10)^2 + (10 - 10)^2 + (10 - 58.9)^2 \ldots (10 - 58.9)^2 = 18609.08$$

CART- Regression Trees

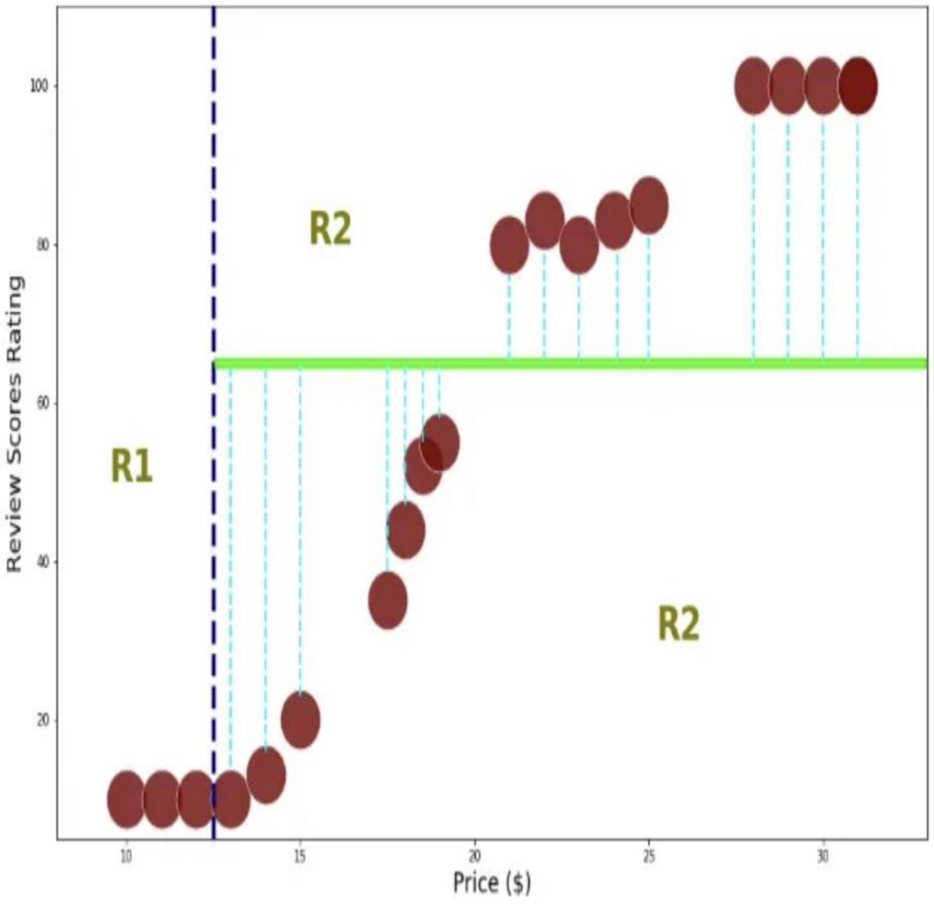
Start within index 2

calculate **RSS** by split into two regions within index 2





CART- Regression Trees



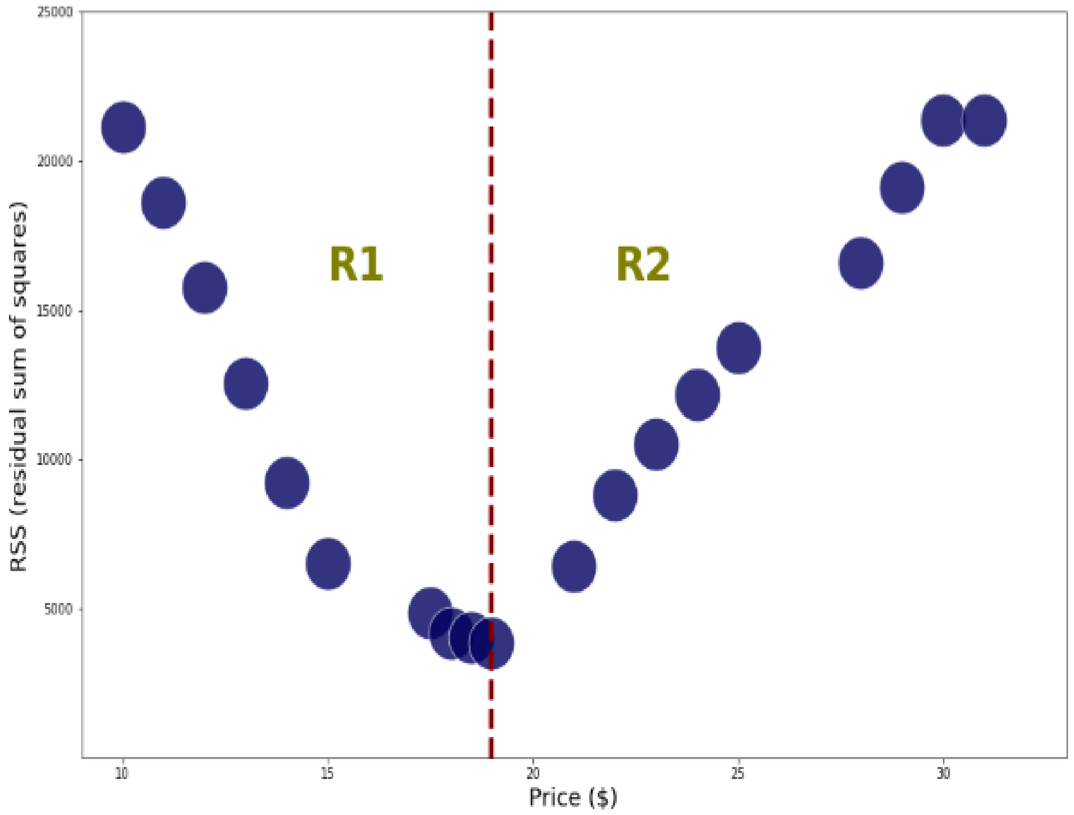
$$\varepsilon = \sum_{i=1}^{n} (actual\ value - average\ value\ in\ each\ region)^2$$

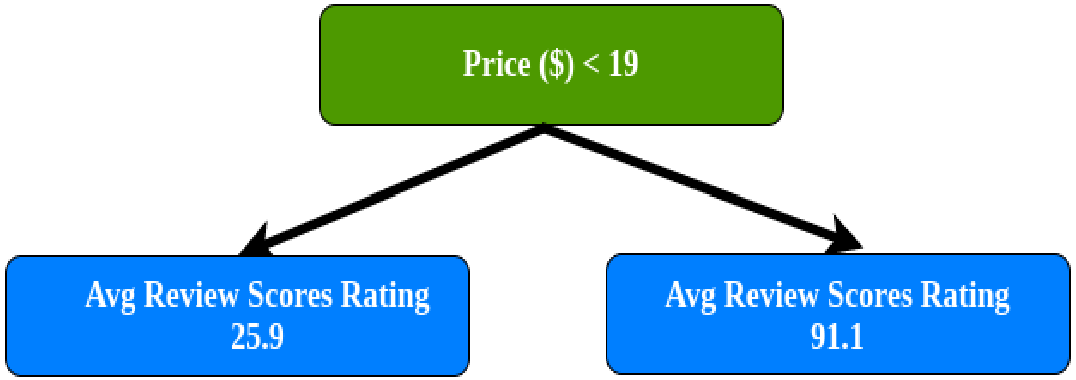
$$RSS = \varepsilon_1^2 + \varepsilon_2^2 + .. + \varepsilon_n^2$$

$$RSS = (10 - 10)^2 + (10 - 10)^2 + (10 - 10)^2 \ldots (10 - 58.9)^2 = 15762.0$$

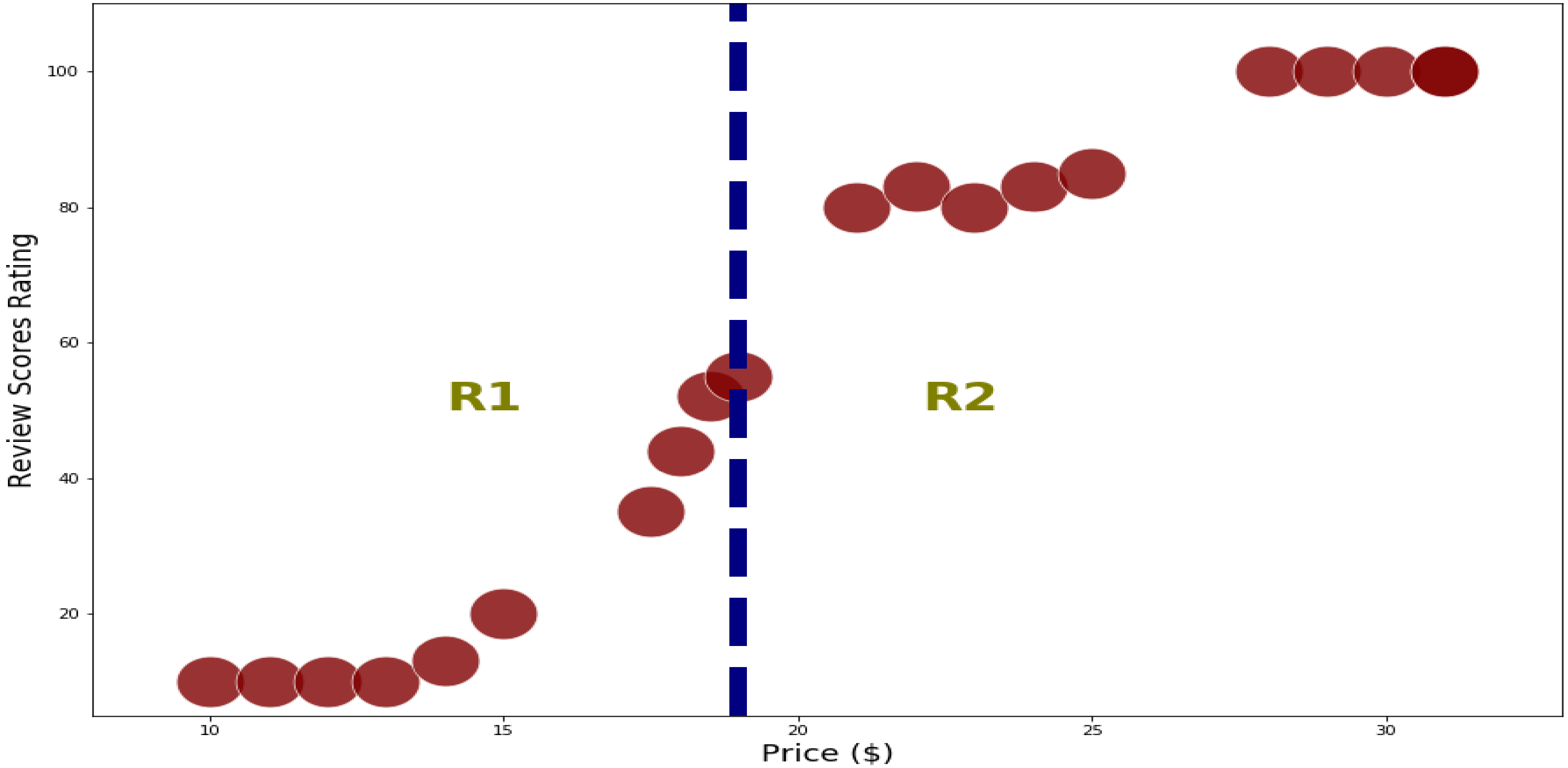
CART- Regression Trees

This process continues until the calculation of RSS in the last index. Price with threshold 19 has a smallest RSS, in R1 there are 10 data within price < 19, so we'll split the data in R1. In order to avoid overfitting, we define the minimum data for each region >= 6. If the region has less than 6 data, the split process in that region stops.



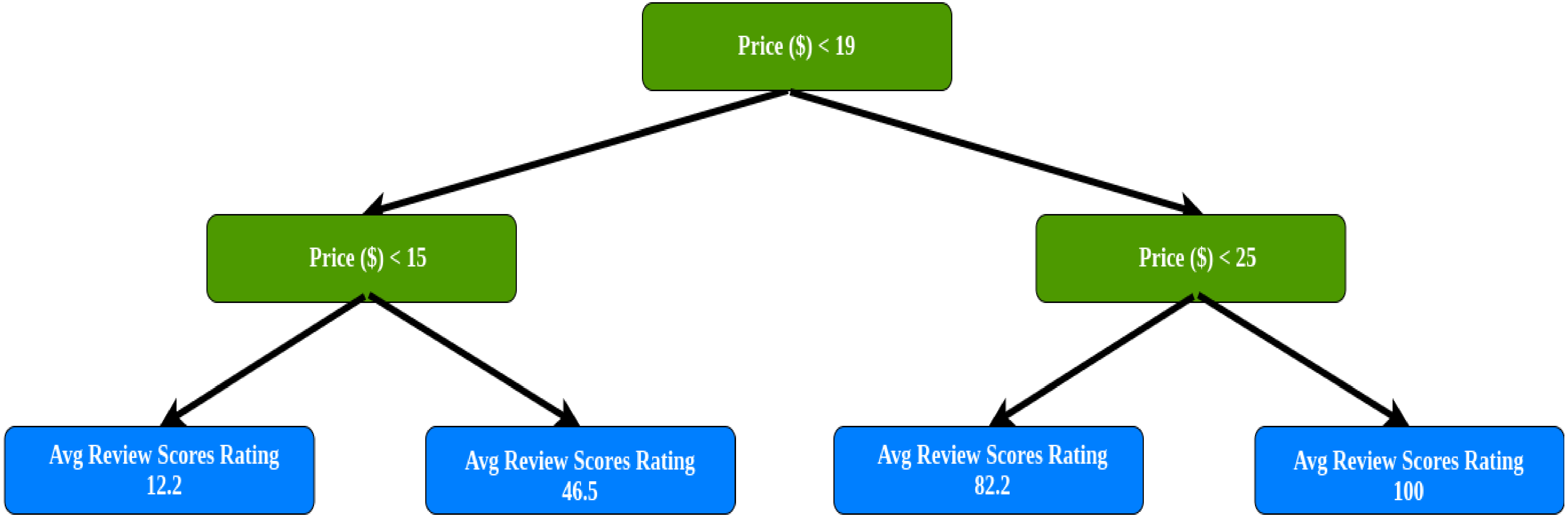


CART- Regression Trees



CART- Regression Trees

Repeat the process till entire tree construction



Regression Trees using ID3

Day	Outlook	Temp	Humidity	Wind	PlayTennis
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

Regression Trees

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
3	Overcast	Hot	High	Weak	46
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
7	Overcast	Cool	Normal	Strong	43
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
10	Rain	Mild	Normal	Weak	46
11	Sunny	Mild	Normal	Strong	48
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44
14	Rain	Mild	High	Strong	30

Regression Trees using ID3

The ID3 algorithm can be used to construct a decision tree for regression by replacing **Information Gain** with ***Standard Deviation Reduction***.

Steps-

1) Calculate Standard Deviation(SD) for class variable
2) Calculate SD for each attribute
3) Calculate weighted SD for each attribute
4) Calculate reduction in SD for each attribute by subtracting its weighted SD from SD of class variable
5) Select the attribute with highest SD reduction
6) Split the dataset based on this attribute
7) Repeat the process for each subset dataset of selected attribute

Regression Trees

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

Standard deviation

Golf players = {25, 30, 46, 45, 52, 23, 43, 35, 38, 46, 48, 52, 44, 30}

Average of golf players = (25 + 30 + 46 + 45 + 52 + 23 + 43 + 35 + 38 + 46 + 48 + 52 + 44 + 30)/14 = 39.78

Standard deviation of golf players = $\sqrt{[((25 - 39.78)^2 + (30 - 39.78)^2 + (46 - 39.78)^2 + \ldots + (30 - 39.78)^2)/14]} = 9.32$

Calculating SD for each Attribute

Outlook can be sunny, overcast and rain

Sunny outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Golf players for sunny outlook = {25, 30, 35, 38, 48}

Average of golf players for sunny outlook = (25+30+35+38+48)/5 = 35.2

Standard deviation of golf players for sunny outlook = $\sqrt{(((25 - 35.2)^2 + (30 - 35.2)^2 + \ldots)/5)}$ = 7.78

Calculating SD for each Attribute

Outlook can be sunny, overcast and rain

Overcast outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

Golf players for overcast outlook = {46, 43, 52, 44}

Average of golf players for overcast outlook = (46 + 43 + 52 + 44)/4 = 46.25

Standard deviation of golf players for overcast outlook = $\sqrt{(((46-46.25)^2+(43-46.25)^2+\ldots)} = 3.49$

Calculating SD for each Attribute

Outlook can be sunny, overcast and rain

Rainy outlook

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

Golf players for overcast outlook = {45, 52, 23, 46, 30}

Average of golf players for overcast outlook = (45+52+23+46+30)/5 = 39.2

Standard deviation of golf players for rainy outlook = $\sqrt{((45 - 39.2)^2+(52 - 39.2)^2+\ldots)/5}$=10.87

Calculating SD for each Attribute

Summarizing standard deviations for the outlook feature

Outlook	Stdev of Golf Players	Instances
Overcast	3.49	4
Rain	10.87	5
Sunny	7.78	5

Weighted standard deviation for outlook = (4/14)x3.49 + (5/14)x10.87 + (5/14)x7.78 = 7.66

You might remember that we have calculated the global standard deviation of golf players 9.32 in previous steps. Standard deviation reduction is difference of the global standard deviation and standard deviation for current feature. In this way, maximized standard deviation reduction will be the decision node.

Standard deviation reduction for outlook = 9.32 – 7.66 = 1.66

Calculating SD for remaining Attributes

Summarizing standard deviations for temperature feature

Temperature	Stdev of Golf Players	Instances
Hot	8.95	4
Cool	10.51	4
Mild	7.65	6

Weighted standard deviation for temperature = (4/14)x8.95 + (4/14)x10.51 + (6/14)x7.65 = 8.84

Standard deviation reduction for temperature = 9.32 – 8.84 = 0.47

Calculating SD for remaining Attributes

Summarizing standard deviations for humidity feature

Humidity	Stdev of Golf Player	Instances
High	9.36	7
Normal	8.73	7

Weighted standard deviation for humidity = (7/14)x9.36 + (7/14)x8.73 = 9.04

Standard deviation reduction for humidity = 9.32 – 9.04 = 0.27

Calculating SD for remaining Attributes

Summarizing standard deviations for wind feature

Wind	Stdev of Golf Player	Instances
Strong	10.59	6
Weak	7.87	8

Weighted standard deviation for wind = (6/14)x10.59 + (8/14)x7.87 = 9.03

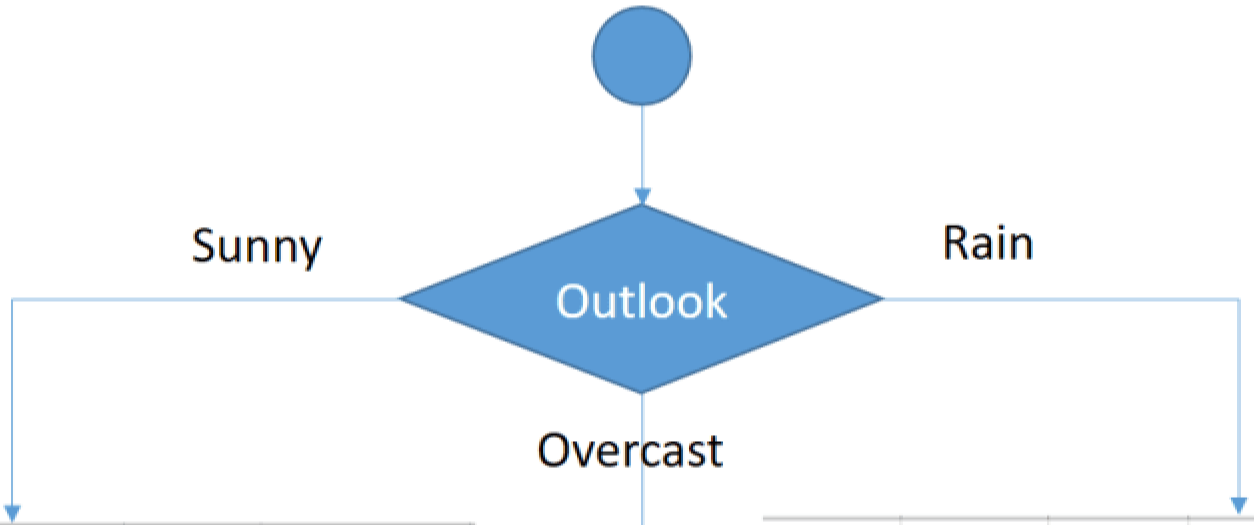
Standard deviation reduction for wind = 9.32 – 9.03 = 0.29

Calculating SD for remaining Attributes

Feature	Standard Deviation Reduction
Outlook	1.66
Temperature	0.47
Humidity	0.27
Wind	0.29

Select the attribute with highest standard deviation reduction

Regression Trees



Day	Outlook	Temp.	Humidity	Wind	Golf Players
1	Sunny	Hot	High	Weak	25
2	Sunny	Hot	High	Strong	30
8	Sunny	Mild	High	Weak	35
9	Sunny	Cool	Normal	Weak	38
11	Sunny	Mild	Normal	Strong	48

Day	Outlook	Temp.	Humidity	Wind	Golf Players
4	Rain	Mild	High	Weak	45
5	Rain	Cool	Normal	Weak	52
6	Rain	Cool	Normal	Strong	23
10	Rain	Mild	Normal	Weak	46
14	Rain	Mild	High	Strong	30

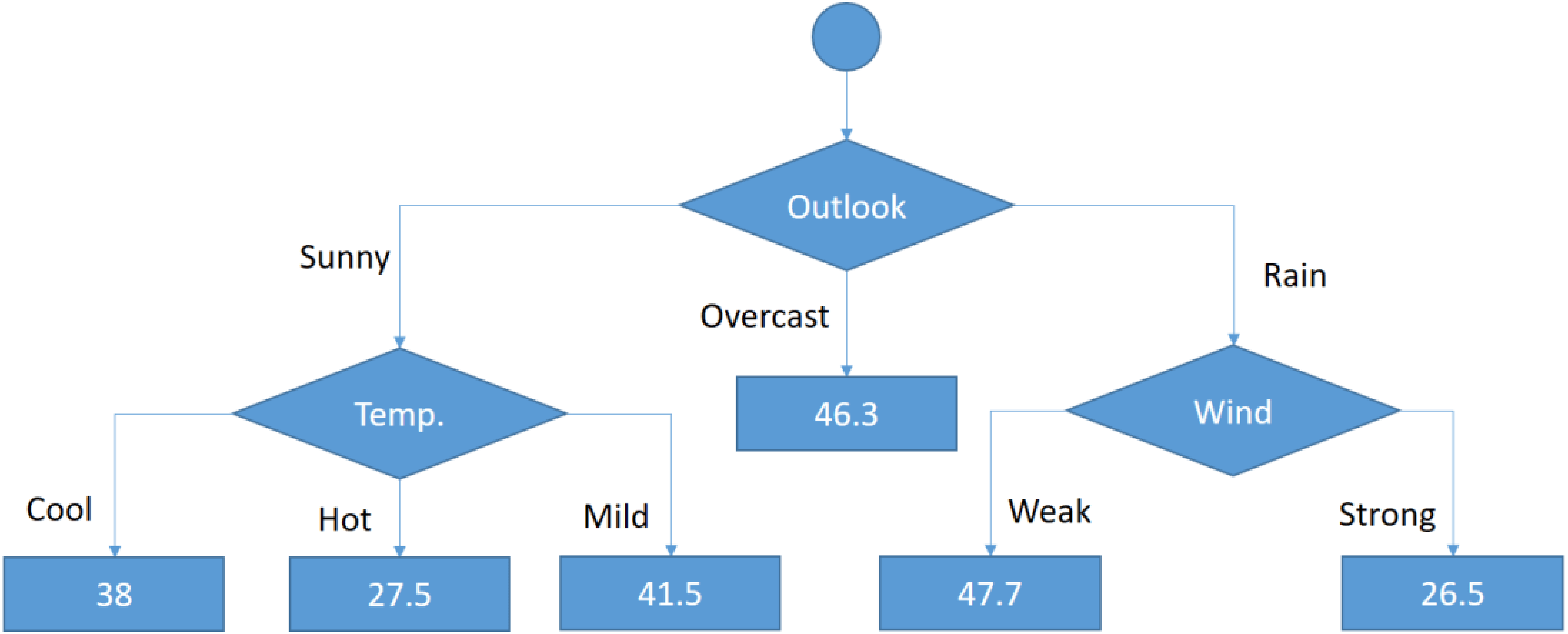
Day	Outlook	Temp.	Humidity	Wind	Golf Players
3	Overcast	Hot	High	Weak	46
7	Overcast	Cool	Normal	Strong	43
12	Overcast	Mild	High	Strong	52
13	Overcast	Hot	Normal	Weak	44

Regression Trees

Continue the process till the defined criteria. Eg- <=4 tuples in each leaf

Take average of continuous values in labels

Regression Trees (Final)



Tree Pruning

- Pruning is a technique that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances.
- Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting.
- There are two methods of pruning decision tree classifier:
- **Pre-pruning** that stop growing the tree earlier, before it perfectly classifies the training set.
- **Post-pruning** that allows the tree to perfectly classify the training set, and then post prune the tree.

Pre-Pruning

- **Limiting the maximum depth of the tree**: Restricting the depth of the tree prevents it from becoming too deep and overly complex.
- **Setting a minimum number of samples required to split a node**: Nodes with fewer samples than the specified threshold are not split further, preventing the creation of low-confidence splits.
- **Setting a minimum number of samples required to be present in a leaf node:** Nodes with fewer samples than the specified threshold are not split, which helps to avoid creating leaf nodes with very few instances.

Post-pruning

- **Cost-complexity pruning (or Minimal cost-complexity pruning):** This method involves calculating a complexity parameter for each node in the tree and then pruning the nodes that result in the smallest increase in overall error or cost when removed.
- The complexity parameter often balances the fit of the tree to the training data with the complexity of the tree.
- **Reduced error pruning:** This method involves removing nodes from the tree if it results in a decrease in error rate on a separate validation dataset.