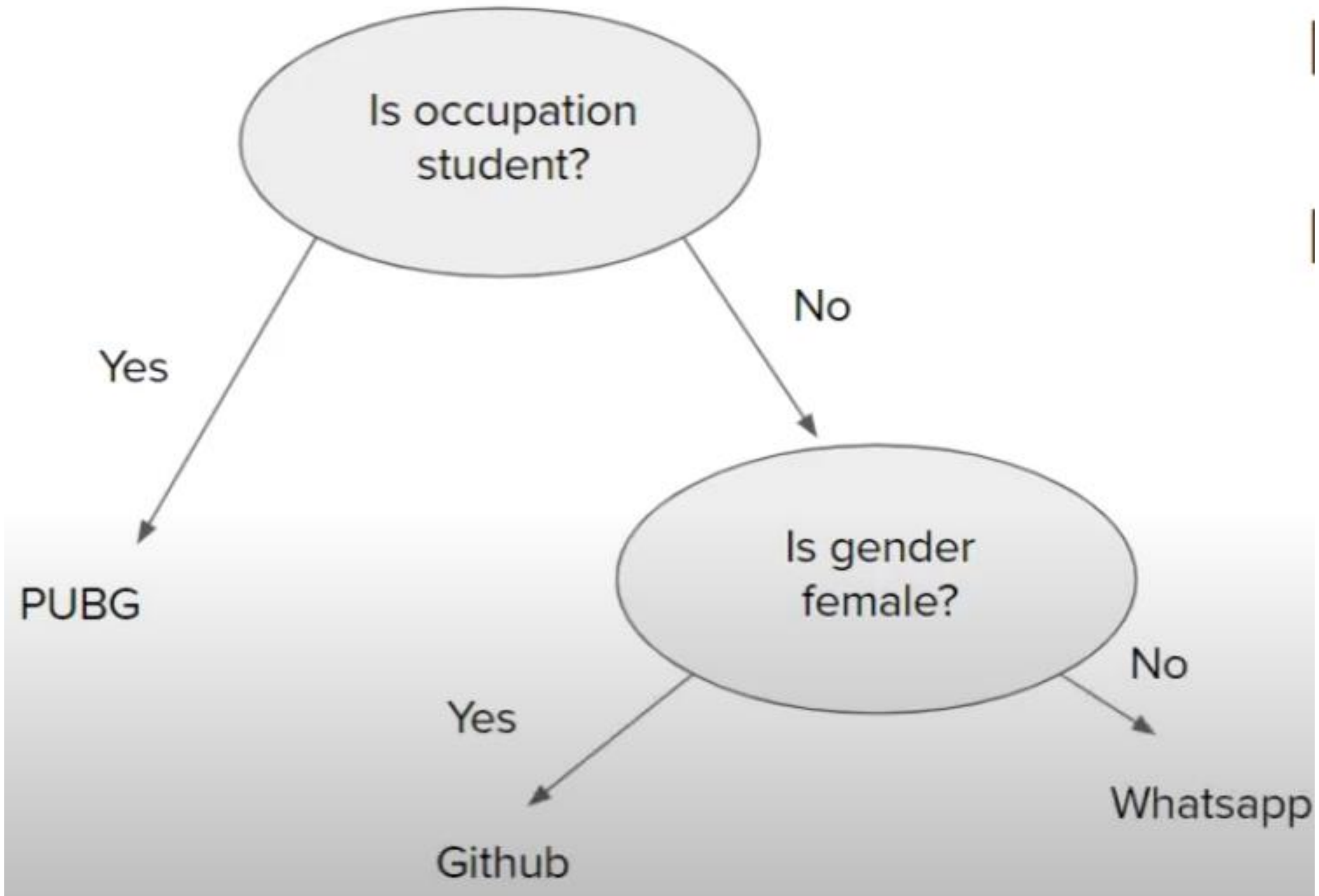
Decision Tree Intuition

Example 1

Gender	Occupation	Suggestion
F	Student	PUBG
F	Programmer	Github
M	Programmer	Whatsapp

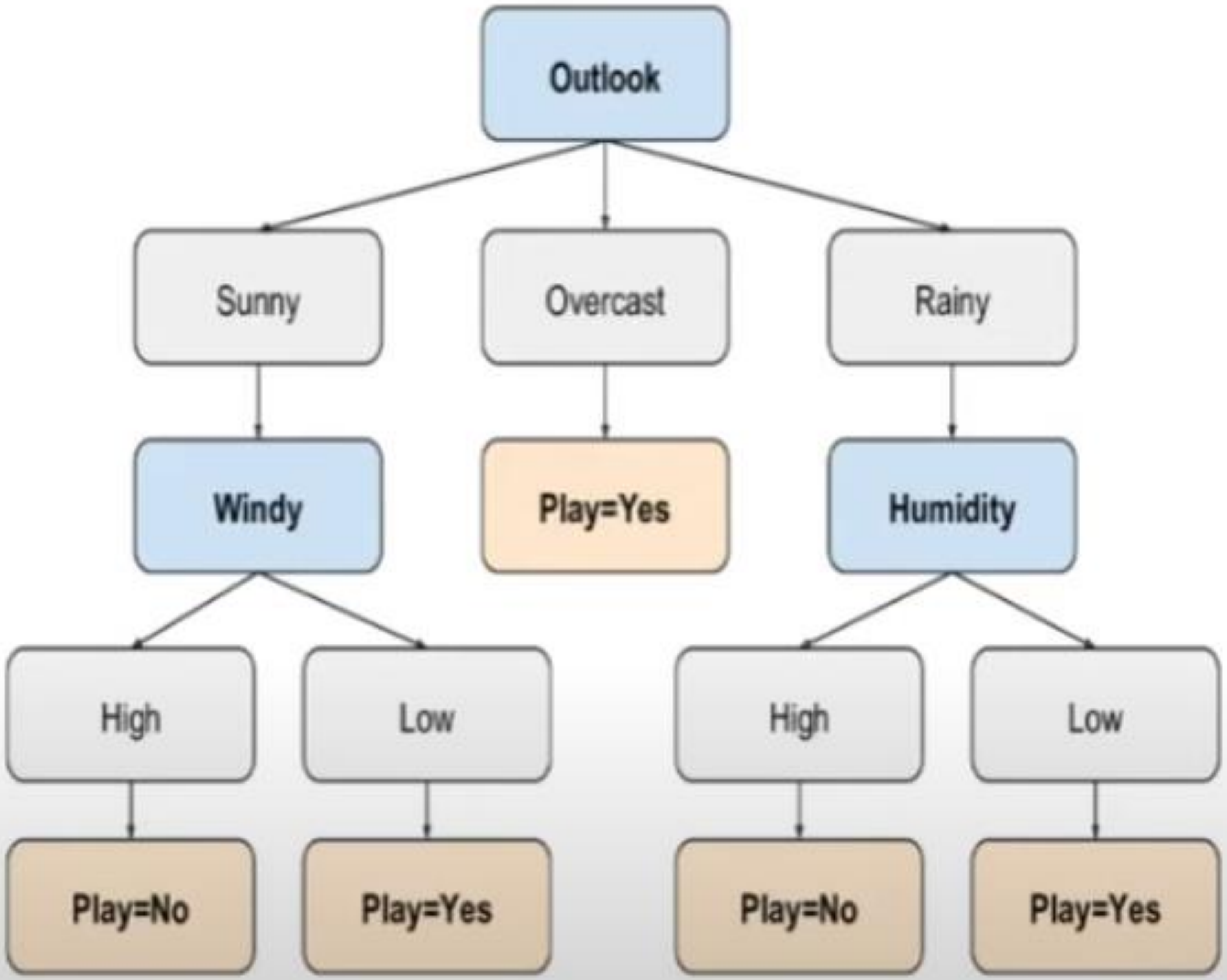
```
If occupation==student
    print(PUBG)
Else
    If gender==female
        print(Github)
    Else
        print(Whatsapp)
```


Is occupation student?
Yes
No
PUBG
Is gender female?
Yes
No
Github
Whatsapp

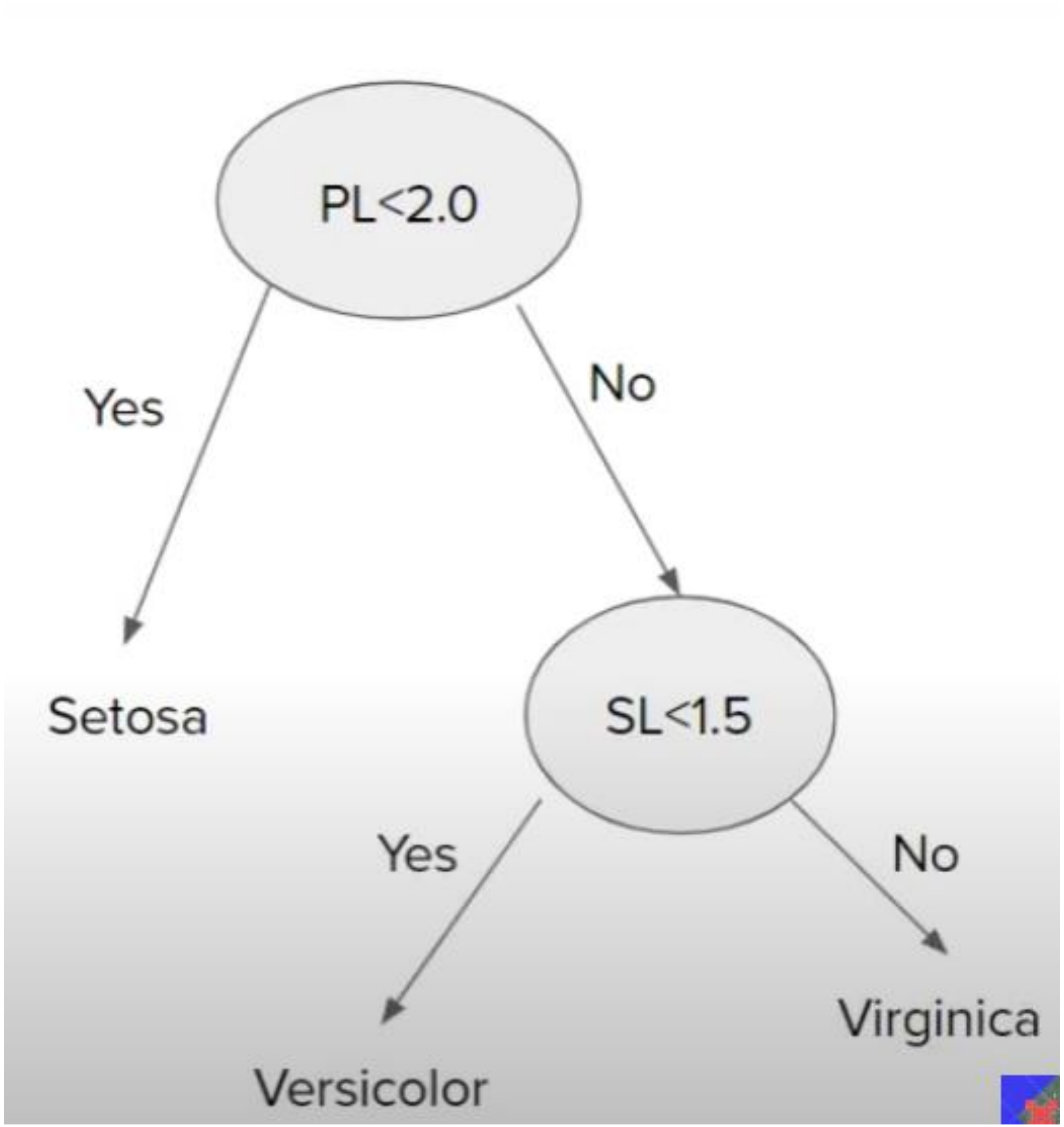
Example 2

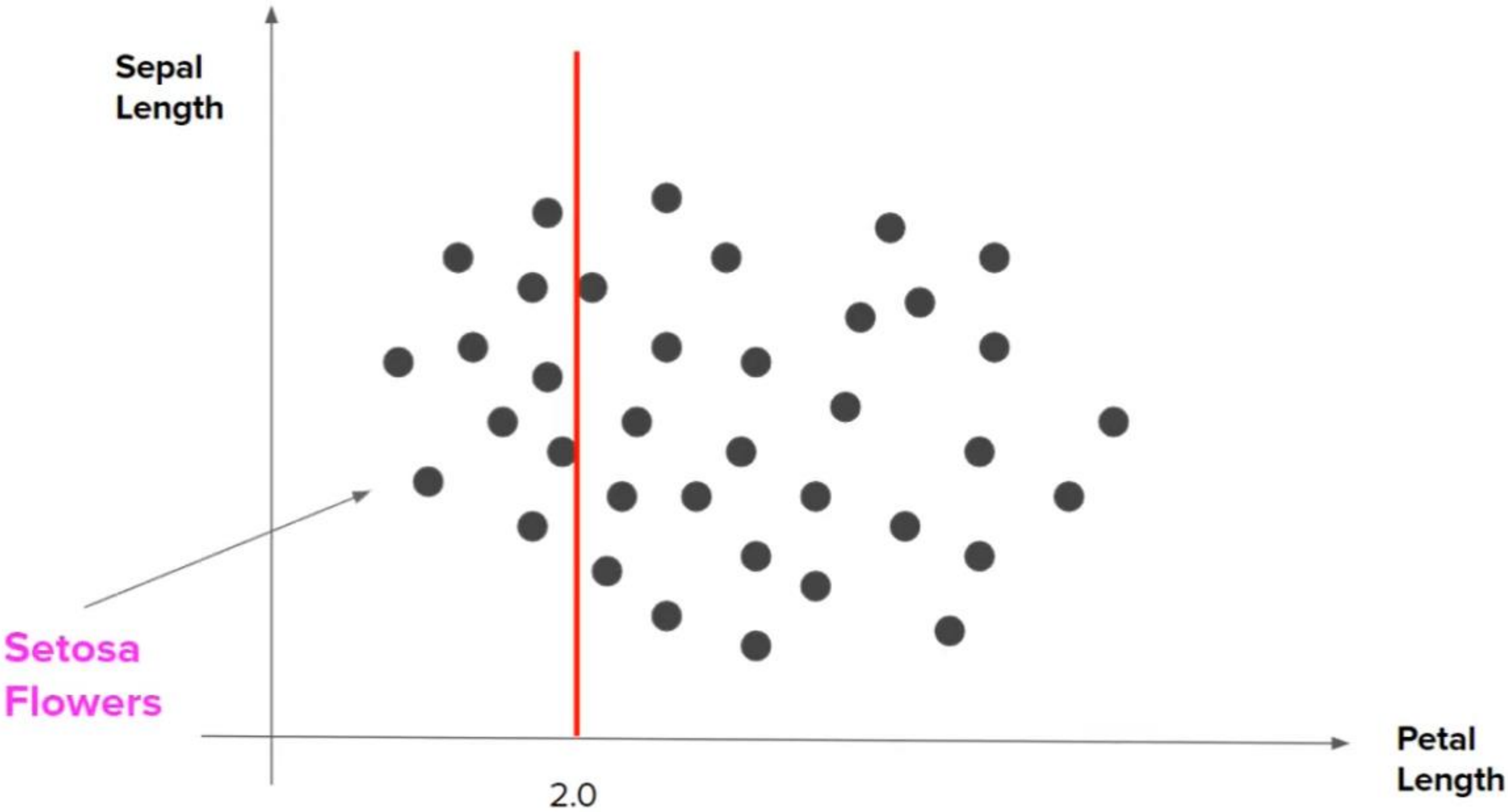
Day	Outlook	Temp	Humid	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	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

Input query point:
[Rainy, Mild, High, Strong]

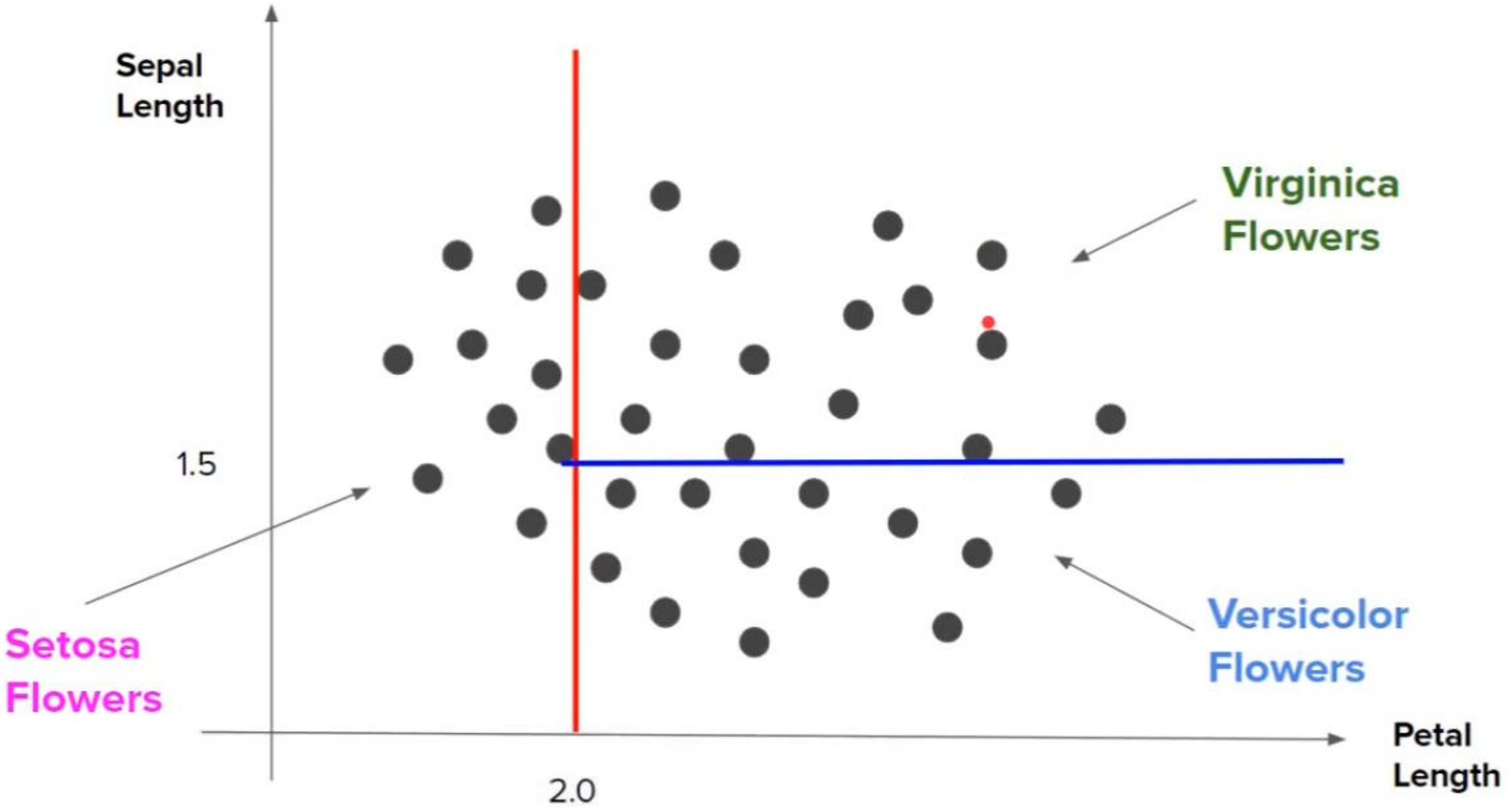


Petal Length	Sepal Length	Type
1.34	0.34	Setosa
3.45	1.45	Versicolor
1.69	0.98	Setosa
2.56	1.79	Virginica
3.00	1.13	Versicolor
1.3	0.88	Setosa


PL<2.0
Yes
No
Setosa
SL<1.5
Yes
No
Versicolor
Virginica


Sepal Length
Setosa Flowers
Petal Length
2.0

Geometric Intuition



Some unanswered questions

How to decide which column should be considered as root node?

How to select subsequent decision nodes?

How to decide splitting criteria in case of numerical columns?

Advantages

Intuitive and easy to understand

Minimal data preparation is required

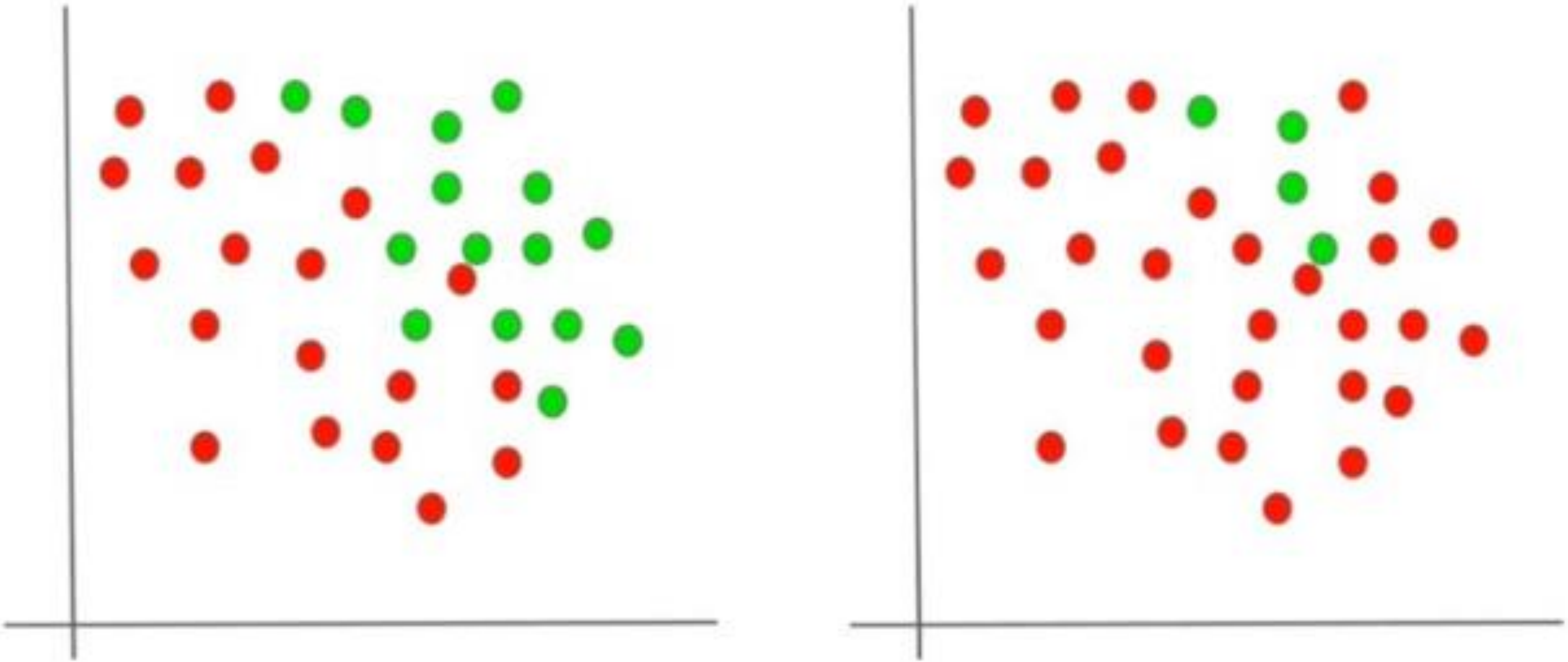
The cost of using the tree for inference is **logarithmic** in the number of data points used to train the tree

Disadvantages

Overfitting

Prone to errors for imbalanced datasets

Entropy



How to calculate Entropy?

The mathematical formula for entropy is:

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

Where 'Pi' is simply the frequentist probability of an element/class 'i' in our data.

For e.g if our data has only 2 class labels Yes and No.

$$\mathbf{E(D) = -p_{yes}\log_2(p_{yes}) - p_{no}\log_2(p_{no})}$$

Salary	Age	Purchase
20000	21	Yes
10000	45	No
60000	27	Yes
15000	31	No
12000	18	No

Salary	Age	Purchase
34000	31	No
15000	25	No
69000	57	Yes
25000	21	No
32000	28	No

$$H(d) = -P_y \log_2(P_y) - P_n \log_2(P_n)$$

$$H(d) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5)$$

$$H(d) = 0.97$$

$$H(d) = -P_y \log_2(P_y) - P_n \log_2(P_n)$$

$$H(d) = -1/5 \log_2(1/5) - 4/5 \log_2(4/5$$

$$H(d) = 0.72$$

Salary	Age	Purchase
20000	21	No
10000	45	No
60000	27	No
15000	31	No
12000	18	No

$$H(d) = -P_y \log_2(P_y) - P_n \log_2(P_n)$$

$$H(d) = -0/5 \log_2(0/5) - 5/5 \log_2(5/5)$$

$$H(d) = 0$$

Calculating entropy for a 3 class problem

Salary	Age	Purchase
20000	21	Yes
10000	45	No
60000	27	Yes
15000	31	No
30000	30	Maybe
12000	18	No
40000	40	Maybe
20000	20	Maybe

$$H(d) = -P_y \log_2(P_y) - P_n \log_2(P_n) - P_m \log_2(P_m)$$

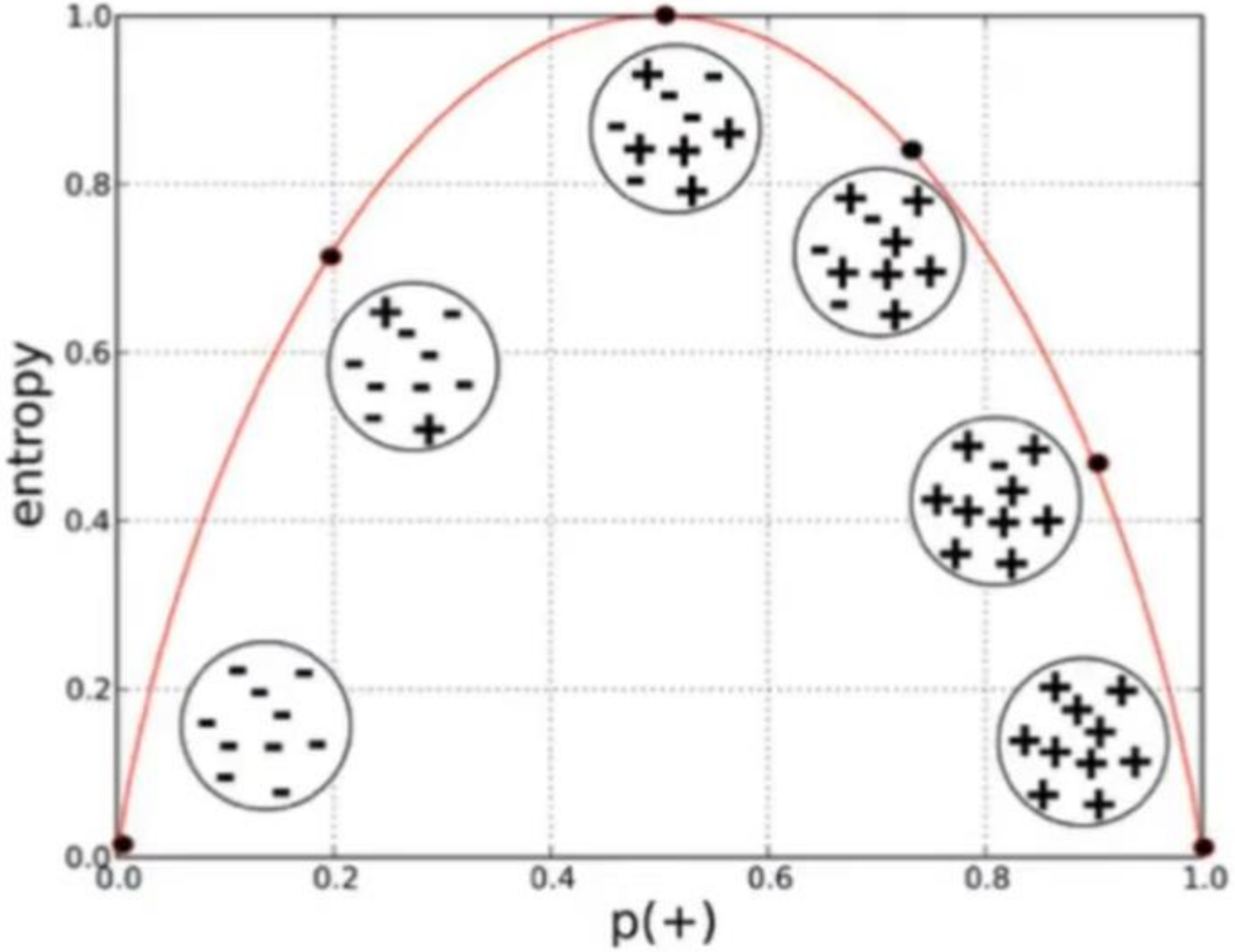
$$H(d) = -2/8\log_2(2/8) - 3/8\log_2(3/8) - 3/8\log_2(3/8)$$

$$H(d) = 1.56$$

Observation

- More the uncertainty more is entropy
- For a 2 class problem the min entropy is 0 and the max is 1
- For more than 2 classes the min entropy is 0 but the max can be greater than 1
- Both $\log_2$ or $\log_e$ can be used to calculate entropy

Entropy Vs Probability



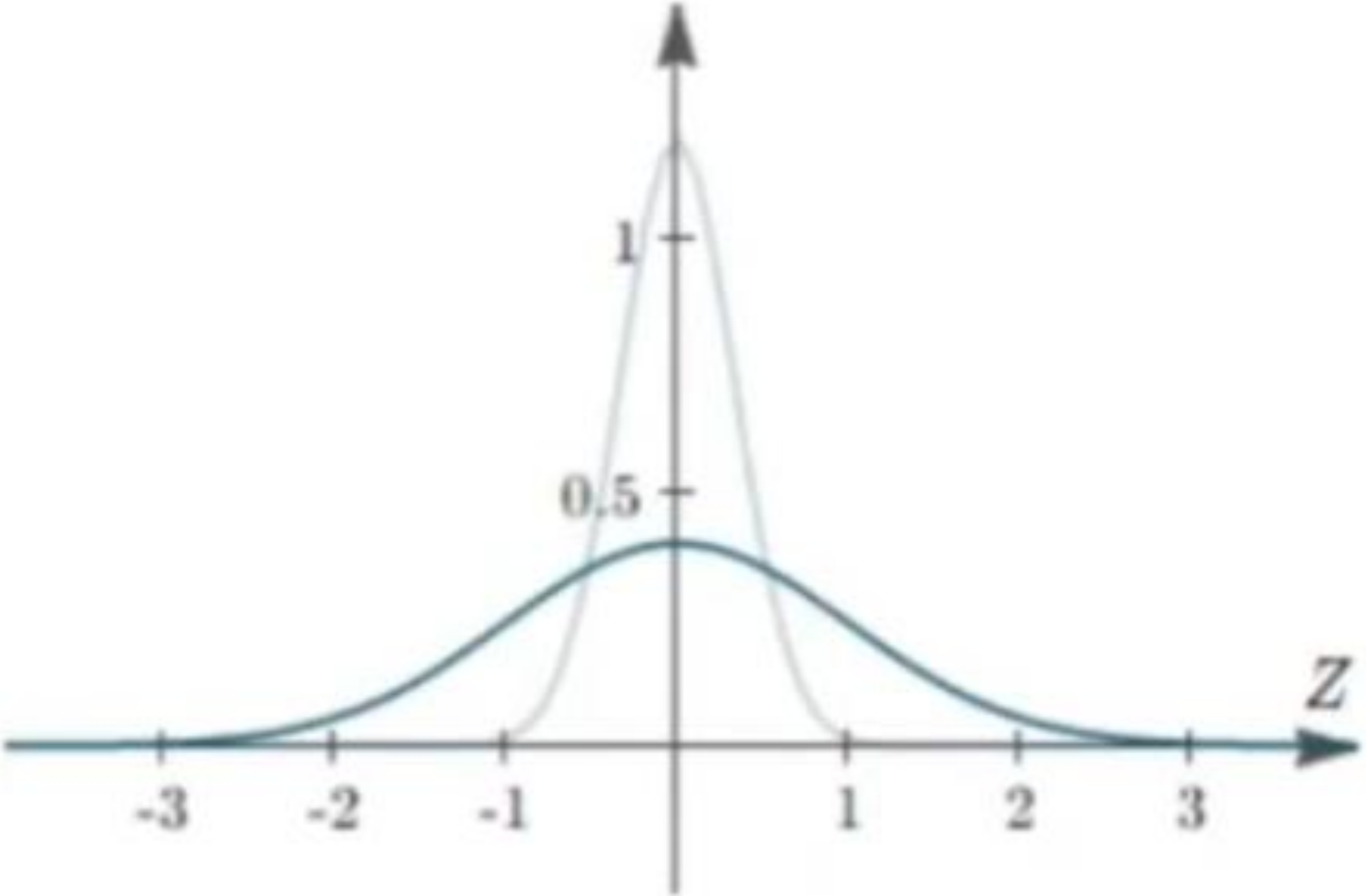
Area	Built in	Price
1200	1999	3.5
1800	2011	5.6
1400	2000	7.3
...	...	...

Dataset 1

Area	Built in	Price
2200	1989	4.6
800	2018	6.5
1100	2005	12.8
...	...	...

Dataset 2

Quiz: Which of the above datasets have higher entropy?



1
0.5
Z
-3
-2
-1
1
2
3

Information Gain

Information Gain, is a metric used to train Decision Trees. Specifically, this metric measures the quality of a split. The information gain is based on the decrease in entropy after a data-set is split on an attribute.Constructing a decision tree is all about finding attribute that returns the highest information gain

Information Gain = E(Parent) - {Weighted Average} * E{Children}

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

Step 1:

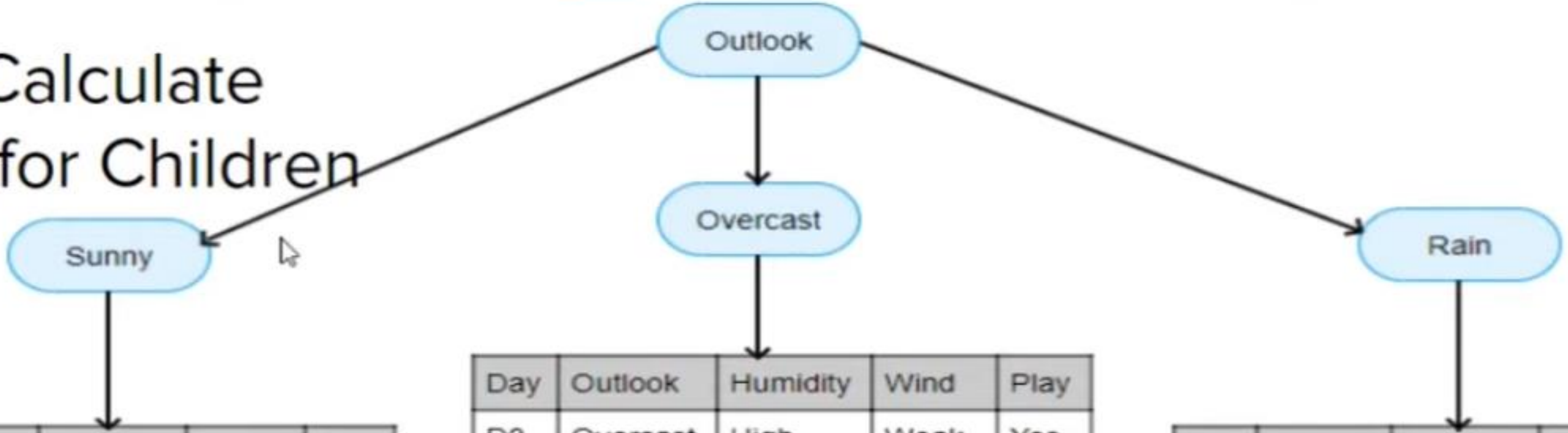
Entropy of Parent

$E(P) = -p_y \log_2(p_y) - p_n \log_2(p_n)$

$= 9/14 \log_2(9/14) - 5/14/\log_2(5/14)$

$E(P) = \mathbf{0.94}$

Step 2: Calculate Entropy for Children

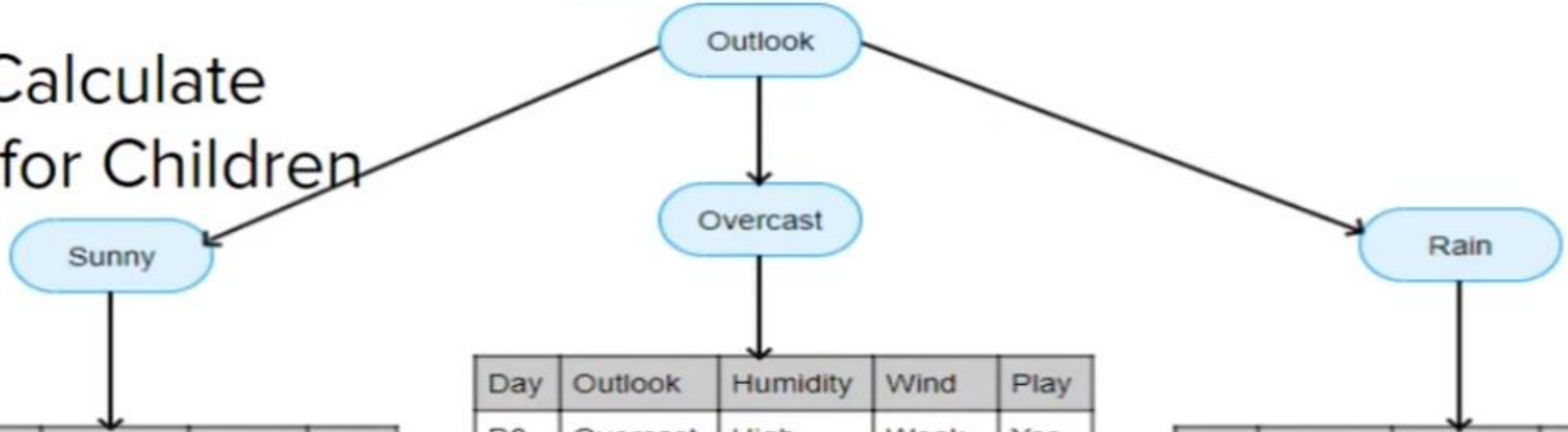


Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes

Day	Outlook	Humidity	Wind	Play
D3	Overcast	High	Weak	Yes
D7	Overcast	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes

Day	Outlook	Humidity	Wind	Play
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D10	Rain	Normal	Weak	Yes
D14	Rain	High	Strong	No

Step 2: Calculate Entropy for Children



Day	Outlook	Humidity	Wind	Play
D1	Sunny	High	Weak	No
D2	Sunny	High	Strong	No
D8	Sunny	High	Weak	No
D9	Sunny	Normal	Weak	Yes
D11	Sunny	Normal	Strong	Yes

Day	Outlook	Humidity	Wind	Play
D3	Overcast	High	Weak	Yes
D7	Overcast	Normal	Strong	Yes
D12	Overcast	High	Strong	Yes
D13	Overcast	Normal	Weak	Yes

Day	Outlook	Humidity	Wind	Play
D4	Rain	High	Weak	Yes
D5	Rain	Normal	Weak	Yes
D6	Rain	Normal	Strong	No
D10	Rain	Normal	Weak	Yes
D14	Rain	High	Strong	No

E(S) =
-2/5log(2/5) - 3/5log(3/5)

E(S)= 0.97

E(O) =
-5/5log(5/5) - 0/5log(0/5)

E(O)= 0

E(R) =
-3/5log(3/5) - 2/5log(2/5)

E(S)= 0.97

Step 3 : Calculate weighted Entropy of Children

Weighted Entropy = 5/14 * 0.97 + 4/14 * 0 + 5/14 * 0.97

W.E(Children) = **0.69**

P(Overcast) is a leaf node as it's entropy is 0

Step 4 : Calculate Information Gain

Information Gain = E(Parent) - {Weighted Average} * E(Children)

IG = **0.97 - 0.69 = 0.28**

So the information gain(or the decrease in entropy/impurity) when you split this data on the basis of **Outlook** condition/column is **0.28**

Step 5 : Calculate Information Gain for all the columns

Whichever column has the highest Information Gain(maximum decrease in entropy) the algorithm will select that column to split the data.

Step 6 : Find Information Gain recursively

Decision tree then applies a recursive greedy search algorithm in top bottom fashion to find Information Gain at every level of the tree.

Once a leaf node is reached (Entropy = 0), no more splitting is done.